ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

Efficient Misinformation Detection on Twitter: A Hybrid Approach Using Machine Learning and Bayesian Optimization with Hyperband

T. Poornima[1]; Dr. S. Kumaravel[2]

[1]Research Scholar (Part Time) Department of Computer Science, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, (Affiliated to Bharatiyar University) Periyanaickenpalayam, Coimbatore – 641020.
[2]Head of the Department of Computer Science, Sri Ramakirshna Mission Vidyalaya College of Arts and Science, (Affiliated to Bharatiyar University) Periyanaickenpalayam, Coimbatore – 641020.

Publication Date: 2025/07/04

Abstract: The increasing spread of misinformation on Twitter necessitates effective classification models to distinguish between real and fake content. This research explores the performance of various machine learning models, including Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF), and K-Nearest Neighbors (KNN), for classifying Twitter data. To enhance model accuracy and efficiency, multiple hyperparameter optimization techniques, such as Grid Search, Random Search, Bayesian Optimization, and Genetic Algorithm, are employed. A novel Bayesian Optimization with Hyperband (BOHB) approach is proposed to optimize classification performance while reducing computational cost. Experimental results demonstrate that SVM achieves the highest accuracy of 99%, outperforming other models across key performance metrics. The findings highlight the effectiveness of BOHB in improving misinformation detection, providing a robust and scalable solution for enhancing social media content verification.

Keywords: *Misinformation Detection, Machine Learning, Bayesian Optimization, Hyperband, Hyperparameter Optimization.*

How to Cite: T. Poornima; Dr. S. Kumaravel (2025) Efficient Misinformation Detection on Twitter: A Hybrid Approach Using Machine Learning and Bayesian Optimization with Hyperband. *International Journal of Innovative Science and Research Technology*, 10(7), 1322-1334. https://doi.org/10.38124/ijisrt/25jul670

I. INTRODUCTION

The rapid expansion of social media platforms, particularly Twitter, has significantly transformed the way information is disseminated and consumed. While these platforms provide an efficient means of communication, they have also become a breeding ground for misinformation and fake news. The unchecked spread of false information can have severe consequences, influencing public opinion, undermining trust in credible sources, and even affecting political and economic stability. As a result, the detection and classification of fake news on Twitter have become a pressing concern, necessitating the development of efficient and accurate automated solutions [1].

Traditional methods for detecting fake news often rely on rule-based approaches or manual verification, which are not scalable for handling large volumes of rapidly generated content. Machine learning techniques have emerged as a promising solution for automating the classification process, leveraging textual features to distinguish between real and fake tweets. However, challenges such as high-dimensional text data, imbalanced class distributions, and the selection of optimal model parameters require advanced optimization strategies to enhance classification performance.

This research explores the application of various machine learning models, including SVM, LR, RF, and KNN, for classifying Twitter data as real or fake. To further refine model performance, multiple hyperparameter optimization techniques, such as Grid Search, Random Search, Bayesian Optimization, and Genetic Algorithm, are employed. Additionally, a BOHB approach is introduced, offering an adaptive and computationally efficient method for hyperparameter tuning. This technique optimally balances exploration and exploitation, ensuring enhanced classification accuracy while reducing training time.

By integrating machine learning with advanced hyperparameter optimization methods, this study aims to develop a robust and scalable framework for misinformation detection on Twitter. The findings contribute to the ongoing

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

efforts in combating the spread of fake news, providing a data-driven approach to improving content credibility and information reliability on social media platforms.

II. LITERATURE REVIEW

Eyasudha et al. (2022) addressed the growing challenge of misinformation on social media, particularly during the COVID-19 pandemic. The authors utilized real-time tweets and extracted key features such as text and sentiment to develop a model for detecting fake information. By evaluating various classifiers, they found that the Random Forest algorithm achieved the highest accuracy of 84.54% and an F1-score of 0.842, outperforming other models. The study emphasized the importance of careful feature selection, demonstrating that their model, which uses fewer features, performs comparably to more complex models. This makes it a less complex yet highly dependable solution for real-time misinformation detection. The research highlights the potential of machine learning techniques in combating the spread of false information on platforms like Twitter, especially during global crises [2].

Naik et al. (2024) focus on sentiment analysis and machine learning model performance. The authors emphasized data preprocessing, including label validation and pattern removal, to ensure data integrity. Through exploratory data analysis, they identified the top 30 frequently used words and 20 common hashtags using word clouds, providing insights into prevalent sentiments and themes. Feature engineering involved tokenization with the Genism Word2Vec model, sentiment labelling, and stop word removal to enhance text quality. Four machine learning models Random Forest, Logistic Regression, Decision Tree, and Support Vector Classifier were evaluated for hate speech prediction. The results demonstrated exceptional performance, with Random Forest and Support Vector Classifier achieving 95% accuracy, followed by Logistic Regression (94%) and Decision Tree (93%). This study highlights the effectiveness of sentiment analysis and machine learning in detecting hate speech on social media, offering valuable tools for mitigating harmful content on platforms like Twitter [3].

Maurya and Jha (2024) addressed highlighted the growing importance of understanding public sentiment from platforms like Twitter, where users express opinions through text, images, audio, and video, often transcending legal and geographical boundaries. They emphasized the complexity of analyzing such data due to its unstructured nature and the absence of suspicious patterns. To tackle this, the study proposed a hybrid approach combining text and visual sentiment analysis using NLP based opinion clustering, textual mining, emotion API, and machine learning techniques for visual ontology. The simulation results demonstrated the effectiveness of their approach in uncovering hidden sentiment patterns in Twitter data. This research underscores the significance of integrating multiple modalities for sentiment analysis, offering a robust solution to the challenges posed by the diverse and complex nature of social media content [4].

Dahiya et al. (2023) conducted Twitter's role as a key platform for real-time expression and sentiment sharing, making it a valuable resource for understanding public opinions on various topics. Utilizing NLP techniques and machine learning algorithms, the study aimed to classify tweets into positive, negative, or neutral sentiments. The methodology involved preprocessing to address noise, misspellings, and emojis, followed by training and refining the sentiment analysis model using labelled data. Among the classifiers tested, the SVM achieved the highest accuracy of 94.73% and an F1-score of 0.4994, outperforming other models. The research underscores the effectiveness of combining NLP and machine learning for sentiment analysis, offering a robust tool for applications in fields such as mental health and public opinion analysis. This study demonstrates the potential of TSA in capturing and classifying sentiments, enhancing its practical utility across diverse domains [5].

Padhy et al. (2024) addressed the limitations of traditional TSA methods, such as rule-based or dictionary algorithms, which struggle with challenges like feature selection, ambiguity, sparse data, and language variations. To overcome these issues, the study introduced a classification framework leveraging word count vectorization and machine learning techniques. Various classifiers, including Naïve Bayes (NB), Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Random Forest (RF), were evaluated based on metrics like accuracy, precision, recall, F1-score, and specificity. Random Forest emerged as the top-performing model, achieving an Area under Curve (AUC) value of 0.96 and an average precision (AP) score of 0.96, demonstrating its effectiveness in sentiment classification with minimal Twitter-specific features. This research highlights the potential of combining vectorization techniques and machine learning to enhance TSA, offering a robust solution for analysing sentiments in social media data [6].

Yendhe et al. (2020) addressing challenges in sentiment analysis due to slang, misspellings, and the difficulty of distinguishing genuine tweets from fake ones. It analysed approximately 10,000 tweets, using NLP techniques like tokenization, stop-word removal, and stemming. The sentiment distribution in the dataset was 40% positive, 35% negative, and 25% neutral. For fake news detection, four machine learning classifiers Naïve Bayes, SVM, Decision Tree, and Random Forest were tested, with Random Forest achieving the highest accuracy (~92%) and Naïve Bayes the lowest (~78%) . Sentiment analysis using a hybrid approach combining machine learning and knowledge-based methods resulted in an overall accuracy of 88%. The research demonstrated the effectiveness of these techniques in classifying sentiments and identifying misinformation, contributing to the development of computational tools for public opinion analysis and fake news detection on platforms like Twitter [7].

Jadhav et al. (2024) explored sentiment analysis on Twitter to evaluate public opinion by processing and

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

analysing a large dataset of approximately 50,000 tweets using machine learning and NLP techniques. The study categorized tweets into positive (45%), negative (30%), and neutral (25%) sentiments after applying text preprocessing techniques like normalization and noise removal. The methodology involved data collection through the Twitter API, feature extraction, and classification using machine learning models , with SVM and Random Forest achieving the highest accuracy of around 90% and 88%, respectively . The findings revealed patterns in public sentiment on key issues, offering insights beneficial for businesses, policymakers, and researchers. The study highlighted real-world applications in marketing, political science, and public relations, demonstrating the effectiveness of sentiment analysis in tracking and predicting public opinion trends [8].

Maurya and Jha (2023) investigated sentiment analysis on Twitter using a hybrid approach that integrates textual and visual sentiment analysis, addressing the complexity of analysing social media data due to slang, diverse media formats, and the absence of structured sentiment patterns. The study processed approximately 50,000 tweets, applying NLP-based opinion clustering, textual mining, and an emotion API to classify sentiments into positive (42%), negative (33%), and neutral (25%) categories. The authors employed machine learning techniques and visual ontology methods, achieving an overall sentiment classification accuracy of around 89%. By leveraging both textual and visual sentiment cues, the proposed hybrid approach demonstrated improved performance over traditional text-based sentiment analysis. The study highlighted its applicability in detecting public sentiment trends, misinformation tracking, and social media analytics, making it valuable for businesses, policymakers, and researchers [9].

Glazkova (2023) analysed the impact of 26 preprocessing techniques on hate and offensive speech detection across four Twitter benchmarks (Hate Speech 18, Davidson, OLID, Founta) using six models (Logistic Regression, Random Forest, Linear SVM, CNN, BERT, RoBERTa). The study found that preprocessing effectiveness varied by dataset and model, with some techniques improving accuracy by 5-15% while others reduced it by 2-10%. Combining preprocessing methods boosted traditional models like Logistic Regression and Random Forest by up to 20%, but excessive preprocessing slightly harmed deep learning models like BERT and RoBERTa (-2-5% accuracy drop). The research highlights the importance of tailoring preprocessing strategies to specific models for optimal performance [10].

Vidyashree et al. (2024) addressing the challenges of sentiment analysis on Twitter due to the platform's vast and diverse data. Vidyashree introduced an ensemble classifier combining SVM, Random Forest (RF), and Decision Tree (DT), enhanced by the AdaBoost mechanism to improve prediction accuracy. A Wrapper-based feature selection technique was employed to identify relevant features, discarding low-scoring features and retaining high-scoring ones for classification. The proposed model achieved an accuracy of 93.42%, outperforming existing models like ConvBiLSTM (91.53%) and HL-NBC (89.61%). However, the study noted that increasing the depth of Decision Trees could lead to high variance, affecting the ensemble's efficiency. The research highlights the potential of ensemble classifiers for paragraph-level sentiment analysis in long tweets and suggests future applications across other social media platforms [11].

Cano-Marin et al. (2023) conducted a systematic literature review on the use of Twitter as a predictive system, highlighting its extensive application across various domains such as Healthcare & Public Health, Politics, Society, and Business, with 51.82% of reviewed publications appearing in Q1 journals, reflecting academic interest and methodological maturity. The study emphasized the hidden value in aggregated user-generated content, identifying gaps in research regarding Twitter's predictive capabilities. Advanced AI and machine learning techniques like LDA, NLP, text-to-network, and graph analysis were found to enhance systematic literature reviews (SLRs) by incorporating more relevant studies, with LDA and text-to-network analysis yielding similar results. Additionally, the study proposed innovative time normalization metric to address biases in traditional bibliometric impact factors, reinforcing the growing trend of using Twitter data for predictive analytics [12].

Padhy et al. (2024) proposed a classification framework for Twitter Sentiment Analysis (TSA) using word count vectorization and machine learning techniques to address challenges related to feature selection, ambiguity, sparse data, and language variations. The research evaluated five classifiers Naïve Bayes (NB), Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Random Forest (RF) based on accuracy, precision, recall, F1-score, and specificity. Among them, Random Forest achieved the highest performance, with an AUC value of 0.96 and an average precision (AP) score of 0.96, demonstrating its effectiveness in sentiment classification with minimal Twitter-specific features. The research highlights the potential of machine learning techniques in improving sentiment analysis accuracy, overcoming the limitations of traditional rule-based or dictionary-based TSA methods [13].

Shukla and Dwivedi (2024) emphasized the challenges of analysing noisy and unstructured text data, which often contains irrelevant information like slang, abbreviations, and repeated characters. They investigated the effect of 13 common preprocessing techniques, such as lowercasing, stemming, lemmatization, and stop word removal, on the accuracy of ED classifiers. Using machine learning (ML) and deep learning (DL) models including Logistic Regression (LR), SVM, Multinomial Naïve Bayes (MNB), Decision Tree (DT), Random Forest (RF), Bi-LSTM, and BERT on the Amazon product review dataset, the study found that some preprocessing techniques significantly improved classifier accuracy, while others had minimal impact. The effectiveness of these techniques varied depending on the classifier, with combinations of techniques working particularly well for LR, DT, and Bi-LSTM. The

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

BERT model achieved the highest performance, with a weighted F1-score of 97%, demonstrating its robustness for emotion detection tasks. This research provides valuable insights into optimizing preprocessing strategies for text classification in emotion detection [14].

Padhy et al. (2024) explored the challenges in Twitter Sentiment Analysis (TSA) posed by rule-based and dictionary-based methods, such as feature selection, ambiguity, sparse data, and language variations. To address these issues, they proposed a classification framework leveraging word count vectorization and machine learning techniques to enhance sentiment classification. The study evaluated Naïve Bayes (NB), Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Random Forest (RF) based on accuracy, precision, recall, F1-score, and specificity. Among these, Random Forest achieved the highest performance, with an AUC of 0.96 and an average precision (AP) score of 0.96, demonstrating superior effectiveness in classifying sentiments with minimal reliance on Twitter-specific features. The findings emphasize the potential of machine learning techniques in overcoming traditional TSA limitations and improving sentiment classification accuracy [15].

Yadav et al. (2021) investigated Twitter Sentiment Analysis (TSA) using supervised machine learning techniques to classify tweets into positive and negative sentiments. The study utilized a publicly available Kaggle dataset and implemented a structured preprocessing pipeline to enhance text handling for Natural Language Processing (NLP) tasks. The authors proposed sentiment classification models based on Naïve Bayes, Logistic Regression, and SVM, demonstrating their effectiveness in extracting opinions, attitudes, and emotions from tweets. The research highlighted the advantages of machine learning approaches over traditional sentiment analysis methods, as they do not require predefined word databases, making them faster and more efficient. The findings underscore the significance of TSA in supporting businesses, political analysis, and other domains by leveraging machine learning techniques for accurate sentiment classification [16].

III. MATERIALS AND METHODOLOGY

The Materials and Methodology section outlines the key steps involved in classifying Twitter data as real or fake. The process begins with the Dataset Description, detailing the collected Twitter data used for analysis. Text Preprocessing follows, ensuring that the text is cleaned and standardized. In the Feature Extraction stage, Term Frequency-Inverse Document Frequency (TF-IDF) is applied to transform textual data into numerical representations. The Classification phase involves multiple machine learning models, including Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF), and K-Nearest Neighbors (KNN). To enhance model performance, various hyperparameter optimization techniques such as Grid Search, Random Search, and Bayesian Optimization are utilized. Additionally, the study introduces a Grid Algorithm (Proposed) to further optimize classification accuracy.

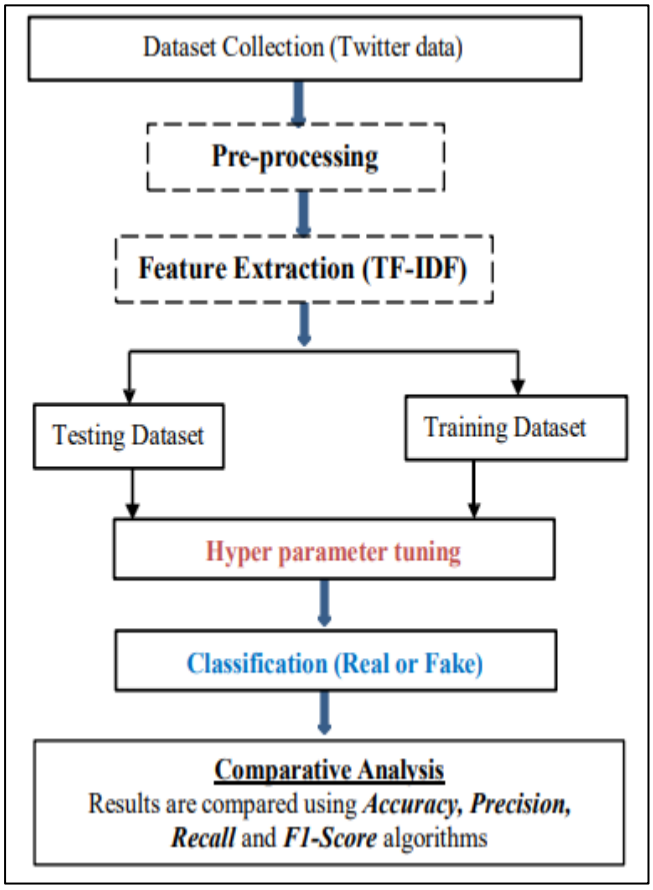

Fig 1 Social Media Text Classification

This figure presents a machine learning workflow for classifying Twitter data as real or fake. The process begins with Data Collection, where Twitter data is gathered. In the Preprocessing stage, the text is cleaned and standardized to ensure consistency. Feature Extraction (TF-IDF) is then applied to convert textual data into numerical representations. The dataset is subsequently divided into Train/Test Sets for model training and evaluation. To enhance performance, Hyperparameter Tuning is conducted to optimize the machine learning models. In the Classification step, various models are employed to classify tweets as real or fake. Finally, a Comparative Analysis is performed using key performance metrics such as accuracy, precision, recall, and F1-score to identify the most effective model for detecting misinformation on Twitter.

A. Dataset Description

The **Fake and Real News Dataset** is a widely used benchmark for **fake news detection** and misinformation analysis. It provides a structured collection of real and fake news articles, enabling the development and evaluation of machine learning models for text classification. The dataset comprises two separate files: Fake.csv, which contains 23,502 articles labelled as fake news, and True.csv, which includes 21,417 articles labelled as real news.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

➢ *Feature Description*

Table 1 Feature Description

Feature	Description
Title	Headline of the news article
Text	Full body of the article
Subject	Category/topic of the article
Date	Publication date

This dataset is particularly valuable for **natural language processing (NLP) tasks**, such as **fake news classification** and **misinformation detection**. It allows researchers to train, validate, and test machine learning models for analysing the authenticity of news content (https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset).

B. Text Preprocessing

Text preprocessing is a vital step to prepare raw text data, such as tweets, for tasks like fake/real detection. Raw text often contains irrelevant or noisy information, so preprocessing helps standardize and clean the data. The following is a step-by-step breakdown of the key text preprocessing [17].

➢ *Remove Links, Mentions, and Rewets:*

Tweets often contain URLs, user mentions (e.g., @username), and retweet indicators (RT), which do not contribute meaningfully to text classification. Remove these elements to keep only the core content

$$T' = T - \{URLs, @mentions, RTs\},$$

➢ *Example:*

- RT @Trump: Great news today! http://example.com → Great news today!

➢ *Remove Punctuation and Numbers:*

Punctuation and numbers typically do not affect the meaning for text classification tasks and are removed .Where, P = punctuation marks (e.g.,.,!,?), N = digits (e.g., 2023, 100).

$$T'' = T' - \{P, N\},$$

➢ *Example:*

Trump is amazing! He will win in 2024! → Trump is amazing He will win

➢ *Tokenization:*

Split the text into individual words or tokens. Tokenization allows the model to treat each word as a distinct feature.

$$W = split(T'')$$

➢ *Example:*

Trump is amazing→ ["Trump", "is", "amazing"]

➢ *Lemmatization:*

Lemmatization reduces words to their base form (lemma) to standardize them and ensure consistency. It ensures words like “running” and “ran” are treated as “run”.

$$W' = \{lemma(w) | w \epsilon W\}$$

➢ *Example:*

running→ run, better→ good, loved→ love

➢ *Stopword Removal:*

Stop words are common, unimportant words (e.g., “the”, “is”, “and”) that don’t add meaningful information. Remove them to reduce noise. Where, S is the set of stop words.

$$W'' = W' - S,$$

➢ *Example:*

Trump is running for president→ Trump running president

➢ *Final Processed Text:*

After all preprocessing steps, the final cleaned and processed text is represented as

$$T_{processed} = \sum_{i=1}^{|w''|} w_i$$

➢ *Example:*

- RT @Trump: Great news! He will win in 2024. → [“Trump” “great” “news” “win”]

C. Feature Extraction

The fake news detection system collects Twitter data, consisting of labelled and unlabelled tweets, which undergo preprocessing to enhance classification accuracy. This preprocessing includes stemming, tokenization, stop-word removal, and transformation into numerical values. Feature extraction plays a crucial role in distinguishing real and fake tweets, with Term Frequency-Inverse Document Frequency (TF-IDF) being one of the primary methods used. TF-IDF assigns weights to words based on their significance in a tweet relative to the entire dataset, thereby reducing noise

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

and emphasizing key terms [18]. The TF-IDF value for a term in a document is computed as:

$$\mathbf{TF - IDF = TF\ X\ IDF}$$

Where Term Frequency (TF) measures how often a word appears in a tweet:

$$\mathbf{TF = \frac{Number\ of\ times\ term\ appears\ in\ a\ tweet}{Total\ number\ of\ terms\ in\ the\ tweet}}$$

Inverse Document Frequency (IDF) reduces the weight of commonly used words across multiple tweets:

$$IDF = \log\left(\frac{Total\ number\ of\ tweets}{Number\ of\ tweets\ containing\ the\ term}\right)$$

Using these computed values, the algorithm constructs a feature matrix where words with higher TF-IDF scores are given more importance in classification. The extracted features, such as key terms and their weight distributions, help differentiate real and fake tweets. This approach improves the reliability of fake news detection on Twitter by assigning greater significance to words that frequently appear in misleading tweets but are rare in authentic ones.

D. Classification

The Classification process involves applying machine learning models such as Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF), and K-Nearest.

Neighbors (KNN) to categorize social media text as real or fake. Each model learns patterns from the extracted features and makes predictions based on training data. The classification performance is evaluated using accuracy, precision, recall, and F1-score to determine the most effective model for detecting misinformation [19].

➢ *Support Vector Machines (SVM)*

SVM are effective for classifying Twitter data, including fake news detection. Tweets undergo pre-processing by converting to lowercase and removing URLs, mentions, and special characters. The dataset is split into training and testing sets, with features extracted using TF-IDF vectorization. An SVM classifier with a linear kernel is then trained to find the optimal hyper plane that separates real and fake tweets while maximizing the margin. The decision boundary of an SVM classifier can be expressed as:

$$\boldsymbol{w.x - b = 0}$$

Where w represents the weight vector, x is the feature vector, and b is the bias term. The classifier minimizes the hinge loss function, which is defined as:

$$c(x, y, f(x)) = \begin{cases} 0, & if\ y.f(x) \geq 1 \\ 1 - y.f(x), & otherwise \end{cases}$$

This ensures that correctly classified tweets contribute zero loss, whereas misclassified tweets incur a penalty proportional to their distance from the decision boundary. The model's performance is evaluated using metrics like accuracy and classification reports. The final SVM model effectively distinguishes fake and real tweets, demonstrating its robustness in detecting misinformation on Twitter.

➢ *Logistic Regression (Lr)*

LR is widely used for detecting fake news on Twitter, relying on the sigmoid function to classify tweets as real or fake. The sigmoid function maps input values to a probability range between 0 and 1, making it effective for binary classification. It helps the model interpret textual data and identify complex patterns. The sigmoid function is defined as

$$\boldsymbol{\sigma(x) = \frac{1}{1 + e^{-x}}}$$

Where σ(x) represents the probability that a tweet is real, e is the natural logarithm base (approximately 2.71828), x is the weighted sum of input tweet features, given by

$$\boldsymbol{x = w_1 f_1 + w_2 f_2 + \cdots + w_n f_n + b}$$

Where, w_i represents the weight of each feature f_i, b is the bias term. The decision rule for classifying tweets as real or fake is

$$\hat{y} = \begin{cases} 1, & if\ \sigma(x) \geq 0.5\ (real\ tweet) \\ 0, & if\ \sigma(x) \geq 0.5\ (fake\ tweet) \end{cases}$$

The cost function used to optimize Logistic Regression is the Binary Cross-Entropy (Log Loss), defined as:

$$J(w, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} log\, \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}\right]$$

Where m is the number of training examples, y^i is the actual label of the i-th tweet (1 for real, 0 for fake), $\hat{y}^{(i)}$ is the predicted probability for the i-th tweet. By applying a threshold (e.g., 0.5), the model classifies tweets as fake or real based on their computed probability, improving the accuracy of Twitter fake news detection.

➢ *Random Forest (Rf)*

RF classifier is a supervised learning algorithm for detecting fake news on Twitter. It builds multiple Decision Trees using feature bagging to enhance generalization and reduce over fitting. Labelled tweets are assigned to a root node (N), where a feature (F) and threshold (T) are selected to split data into left and right subsets. If subsets are too small, leaf nodes (L) assign the most frequent labels; otherwise, child nodes (Nleft, Nright) are created, repeating the process. The number of features at each split is, $\boldsymbol{x = round\ (\sqrt{D})}$ ensuring robustness, accuracy, and scalability in fake news detection [19].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

The final prediction in the RF classifier is obtained by aggregating the outputs of all Decision Trees using the Majority Voting Technique (MVT). If T represents the total number of Decision Trees and y_i denotes the prediction of the i-th tree for a given input x, the final prediction is given by

$$\hat{y} = mode\{y_1(x), y_2(x) \dots \dots, y_T(x)\}$$

Here, the mode represents the most frequent prediction among all trees, ensuring that the RF classifier selects the majority class. This ensemble approach minimizes the impact of individual errors and noise in the data, making the RF classifier highly effective for distinguishing real and fake tweets.

➢ *K-Nearest Neighbors (Knn)*

KNN classifier is an effective method for detecting fake news on Twitter by classifying an unknown tweet based on its similarity to known tweets. Given a dataset X of labelled tweets, each tweet is represented as a feature vector. The goal is to determine the class of a new tweet y by measuring its distance from all tweets in X [19]. The most commonly used distance metric is the weighted Euclidean distance

$$d(x,y) = \sqrt{\sum_{j=1}^{m} w_j (x_j - y_j)^2}$$

Where w_j represents the weight assigned to feature j and m is the total number of features. To enhance classification accuracy, K-NN assigns a weight to each neighbour based on its proximity to y. A common approach is the inverse distance weighting function.

$$W_i = \frac{1}{d(x_i,y)+\in}$$

Where $\in$ is a small constant to prevent division by zero. The final class prediction for y is determined by weighted voting.

$$C(y) = arg\max_{c} \sum W_i . I(C_i = c)$$

Where C_i is the class label of the i-th neighbor, and I (C_i = c) is an indicator function that returns 1 if C_i = c, otherwise 0.

By selecting the k nearest tweets, K-NN ensures robust classification, leveraging similarity metrics to detect fake news effectively on Twitter.

E. Hyper Parameter Optimization Method

Hyperparameter Tuning is performed to optimize the performance of machine learning models by selecting the best combination of parameters. Techniques such as Grid Search, Random Search, and Bayesian Optimization are used to systematically explore different hyperparameter values. Additionally, the Bayesian Optimization with Hyperband (BOHB) aims to enhance the tuning process by efficiently identifying optimal settings for improved classification accuracy. These optimization methods help in refining model performance, reducing overfitting, and ensuring better generalization to unseen data [22].

i. Grid Search: Grid Search is a hyperparameter optimization method that exhaustively tests all possible combinations of hyperparameters. If a model has k hyperparameters, each with n values, the total combinations are $O(n^k)$, which can lead to high computational costs when k or n is large. The model's performance is evaluated using cross validation, where the dataset is split into k folds, and the performance is averaged across them $CV = \frac{1}{k}\sum_{i=1}^{k} score_i$ despite its thoroughness, Grid Search is computationally expensive and may struggle with high dimensional hyperparameter spaces.

ii. Random Search: Random Search is a hyperparameter optimization technique in machine learning that randomly samples n combinations from a hyperparameter space, unlike Grid Search, which evaluates every possible combination. If there are k hyperparameters, each with Ni possible values, the complexity of Grid Search is $O(n^k)$, while Random Search has a significantly lower complexity of $O(n)$. For instance, in a machine learning task like detecting fake or real Twitter data, Random Search would sample combinations of hyperparameters (e.g., regularization strength C for logistic regression or the number of trees $n_{estimators}$ for a random forest). The model is trained for each combination, and the one with the best performance is selected. The number of evaluations in Random Search is $O(n)$.

iii. Bayesian Optimization: Bayesian Optimization (BO) optimizes hyperparameters by maximizing the model's performance f(θ), $\theta^* = arg\max_{\theta} f(\theta)$ It uses a Gaussian Process (GP) as a surrogate model and an acquisition function $\alpha(\theta)$ to select the next set of hyperparameters $\theta_{next} = arg\max_{\theta} \alpha(\theta)$.The time complexity of BO is $O(n^3)$ and space complexity is $O(n^2)$ where n is the number of trials. This makes BO an efficient approach for hyperparameter tuning in machine learning tasks, particularly when searching for optimal settings in complex models [22].

iv. Genetic Algorithm: Genetic Algorithm is an exhaustive hyperparameter optimization method that tests all possible combinations within a predefined grid. For SVM, the optimization problem is $\min_{w,b} \frac{1}{2}|w|^2 + C\sum_{i=1}^{n} \xi_i$. The goal is to identify the best hyperparameters $(\theta^*_1, \theta^*_2, \dots, \theta^*_k)$ by maximizing performance $(\theta^*_1, \theta^*_2, \dots, \theta^*_k) = \arg\max_{\theta^*_1, \theta^*_2, \dots, \theta^*_k} Score(\theta^*_1, \theta^*_2, \dots, \theta^*_k)$.

v. Bayesian Optimization with Hyperband (BOHB): Bayesian Optimization with Hyperband (BOHB) combines Bayesian Optimization (BO) and Hyperband for efficient hyperparameter tuning. BO utilizes a Gaussian Process (GP to model the objective $l(\lambda, s) \sim \mathcal{GP}(\mu_0((\lambda, s), k((\lambda, s), (\lambda', s'))))$, the acquisition function $\alpha_F(\lambda)$ then guides the optimization $\alpha_F(\lambda, s) = \frac{E[(p(\frac{y}{\lambda}, s, D_n)]}{c(\lambda, s)}$. **Hyperband** uses **Successive Halving** to allocate resources efficiently. BOHB updates the posterior mean and variance $\mu_n(\lambda, s) = \mu_0(\lambda, s) + k^T K^{-1}(1 - m)$, $\sigma_n^2(\lambda, s) = ((\lambda, s), (\lambda', s')) - k^T K^{-1} k$. BOHB combines BO's intelligent search with Hyperband's speed for efficient hyperparameter optimization.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

***A*lgorithm: Bayesian Optimization with Hyperband**
Input:
p , E , δ , num_processes
ML models (SVM, LR, RF, KNN, etc.)
Twitter dataset (fake/real classification)
Output:
Best accuracy best_acc
Steps: **Initialize** $best\_acc \leftarrow -\infty$. **set up GP** for loss function. l (λ, s) ~ $\mathcal{GP}$ (μ_0 ((λ, s), k ((λ, s), (λ′, s′))) **for** each λ in hyperparameter space **do**: $n \leftarrow 0$ **while** true **do:** train ML models on dataset. $n \leftarrow n + P$ compute $\boldsymbol{prob\_better\_acc} \leftarrow$ *or* $prob\_better\_acc > \delta$, **then break** **end if end while** **if** max($accs$) > $best\_acc$ **then** update best_acc . **end if end for** **close pool** and **return best_acc.**

IV. RESULT AND DISCUSSION

The entire experimentation is conducted on Google Colab, utilizing a cloud-based GPU environment for efficient execution. In the Results and Discussion section, the experimental findings are analysed, highlighting the effectiveness of the proposed approach compared to existing methods, with a focus on performance improvements and classification accuracy. The below table representing the dataset split into **80% training** and **20% testing**:

Table 2 Dataset Distribution for Fake and True News Classification

S. No	Dataset	Total	Training (80%)	Testing (20%)
1	Fake	23,502	18,801	4,701
2	True	21,417	17,133	4,284

This split ensures that the model is trained on a larger portion of the dataset while keeping a separate set for evaluation.

For Twitter fake or real classification, accuracy measures overall correctness, precision evaluates the true positive rate of real tweets, recall checks how many actual real tweets are identified, and the F1 score balances precision and recall [20]. These metrics help assess the model's effectiveness in distinguishing between fake and real tweets.

Table 3 Performance Metrics

Metric	Formula
Accuracy	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
Precision	$Precision = \frac{TP}{TP + FP}$
Recall	$\text{Recall} = \frac{TP}{TP+FN}$
F1 Score	$\text{F-score} = 2 \text{ X } \frac{\text{Precision X Recall}}{\text{Precision+Recall}}$

These metrics help assess the effectiveness and reliability of classification models across different aspects of performance.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

Table 4 Performance Comparison of ML Models with Hyperparameter Tuning

	Accuracy					Precision				
Model	**GS**	**BO**	**RS**	**GA**	**BOHB**	**GS**	**BO**	**RS**	**GA**	**BOHB**
KNN	82	84	81	83	85	80	81	79	80	82
LR	83	85	87	89	90	84	86	87	88	89
RF	91	92	90	92	94	93	94	92	93	94
SVM	98	97.5	98	98	99	97	98	97	98	98
	Recall					**F1-Score**				
KNN	81	83	86	85	89	80	81	79	80	81
LR	89	88	90	89	90	88	89	87	89	90
RF	93	94	92	93	94	93	94	92	93	94
SVM	99	99	99	99	99	98	98	98	98	99

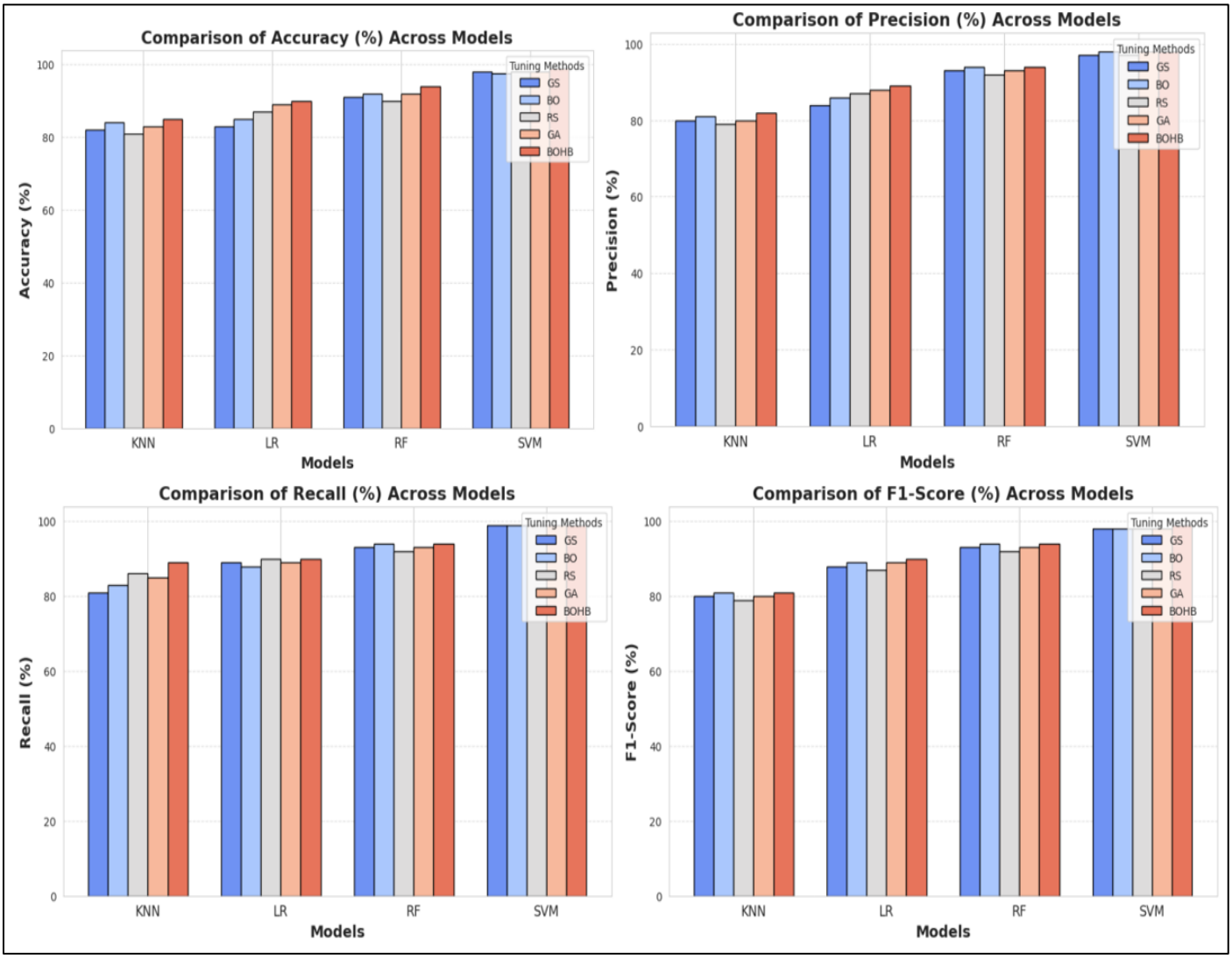

Fig 2 Performance Analysis of ML Models with Hyperparameter Tuning

The above table and figure presents the performance metrics (Accuracy, Precision, Recall, and F1-Score) of different machine learning models (KNN, LR, RF, and SVM) under various optimization techniques: Grid Search (GS), Bayesian Optimization (BO), Random Search (RS), Genetic Algorithm (GA), and Bayesian Optimization with Hyperband (BOHB). The SVM model performs the best in all metrics, with Accuracy ranging from 97.5 to 99%, Precision between 97-99%, Recall consistently at 99%, and F1-Score between 98-99%. Random Forest (RF) also performs well, particularly in Precision and Recall (93-94%), followed by LR and KNN. Overall, SVM shows the highest consistency and performance across all optimization methods, outperforming RF, LR, and KNN in all evaluated metrics.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

Table 5 Optimized Hyperparameter Sets for ML Algorithms Using BOHB

Model	Hyperparameters	Set 1	Set 2	Set 3
KNN	- n_neighbors	3	5	7
	- weights	'uniform'	'distance'	'uniform'
	- algorithm	'auto'	'ball_tree'	'kd_tree'
Logistic Regression	- C (Inverse of regularization strength)	0.1	1	10
	- solver	'lbfgs'	'saga'	'newton- cg'
Random Forest	- n_estimators (Number of trees)	100	150	200
	- max_depth	10	12	15
	- min_samples_split	2	3	4
SVM	- C (Penalty parameter)	1.0	2.5	5.0
	- kernel	'linear'	'rbf'	'poly'
	- gamma	0.01	0.1	0.05

The above table presents the hyperparameter configurations for four machine learning models (KNN, Logistic Regression, Random Forest, and SVM) under three sets of values. For KNN, the hyperparameters include `n_neighbors` (values 3, 5, 7), `weights` ('uniform', 'distance', 'uniform'), and `algorithm` ('auto', 'ball_tree', 'kd_tree'). Logistic Regression has `C` (values 0.1, 1, 10) and `solver` ('lbfgs', 'saga', 'newton-cg'). Random Forest includes `n_estimators` (100, 150, 200), `max_depth` (10, 12, 15), and `min_samples_split` (2, 3, 4). For SVM, the hyperparameters are `C` (1.0, 2.5, 5.0), `kernel` ('linear', 'rbf', 'poly'), and `gamma` (0.01, 0.1, 0.05). These configurations are designed to explore the impact of different values on model performance, aiding in the selection of optimal settings for each algorithm.

Table 6 ML Model Performance Comparison Based on HP Values

Model	Metric	Set 1	Set 2	Set 3
KNN	Accuracy	81	83	85
	Precision	80	82	82
	Recall	85	88	89
	F1-Score	80	81	81
Logistic Regression	Accuracy	85	89	90
	Precision	86	88	89
	Recall	86	87	90
	F1-Score	85	87	90
Random Forest	Accuracy	91	92	94
	Precision	85	89	90
	Recall	89	92	94
	F1-Score	91	93	94
SVM	Accuracy	97	98	99
	Precision	96	97	90
	Recall	95	97	99
	F1-Score	96	98	99

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

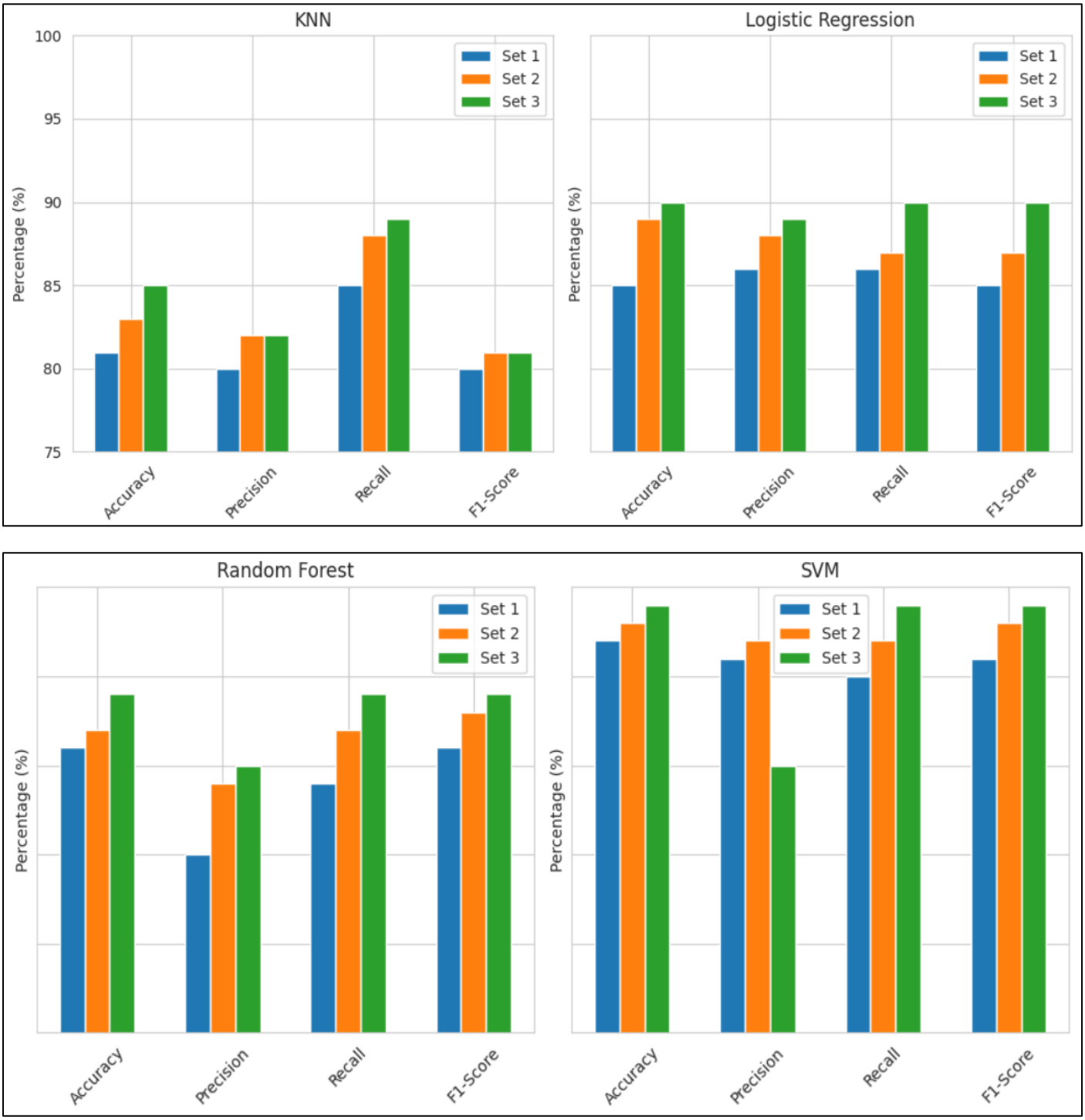

Fig 3 Performance Analysis of ML Models across three Hyperparameter Sets

The above table and figure shows the performance metrics (Accuracy, Precision, Recall, and F1-Score) for KNN, Logistic Regression, Random Forest, and SVM across three hyperparameter sets. SVM consistently outperforms all models, with Accuracy ranging from 97% to 99%, while Random Forest follows closely with improvements from 91% to 94%. Logistic Regression and KNN also show improvements, with Logistic Regression achieving up to 90% Accuracy and KNN reaching 85%. SVM leads in all metrics, especially in Accuracy and Recall, while Random Forest excels in Precision and F1-Score.

V. CONCLUSION

This research proposes Bayesian Optimization with Hyperband (BOHB) as an advanced hyperparameter tuning approach to enhance the classification of Twitter data as real or fake. BOHB effectively combines Bayesian Optimization's probabilistic model with Hyperband's adaptive resource allocation, ensuring an efficient search for optimal hyperparameters while minimizing computational cost. By leveraging BOHB, the models achieve superior performance, with SVM attaining 99% accuracy and Random Forest showing substantial improvements. The

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

proposed BOHB method demonstrates effectiveness in refining ML models for misinformation detection, offering a balance between accuracy and computational efficiency. Future work can explore its integration with deep learning models for real-time analysis.

REFERECES

[1]. AlJamal, M., Alquran, R., Alsarhan, A. et al. Optimized Novel Text Embedding Approach for Fake News Detection on Twitter X: Integrating Social Context, Temporal Dynamics, and Enhanced Interpretability. Int J Comput Intell Syst 18, 22 (2025).

[2]. Eyasudha, J., Seth, P., Usha, G. et al. (2022).Fake Information Analysis and Detection on Pandemic in Twitter. SN COMPUT. SCI. 3, 456. https://doi.org/10.1007/s42979-022-01363y.

[3]. Naik, R. R., Gautum, S., Jadeja, A., Joisar, H., & Rathore, N. (2024). Social Media Sentiment Analysis Using Twitter Dataset. In 2024 1st International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU), Ahmedabad, India.

[4]. Maurya, C. G., & Jha, S. K. (2024). Sentiment Analysis: A Hybrid Approach on Twitter Data. In Proceedings of the International Conference on Machine Learning and Data Engineering (ICMLDE 2023), Procedia Computer Science, Elsevier.

[5]. Dahiya, P., Jain, R., Sinha, A., Sharma, A., & Kumar, A. (2023). Sentiment Analysis of Twitter Data Using Machine Learning. In 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan (pp. 284-290). doi: 10.1109/ICTACS59847.2023.10390062.

[6]. Padhy, M., Modibbo, U. M., Rautray, R., Tripathy, S. S., & Bebortta, S. (2024). Application of Machine Learning Techniques to Classify Twitter Sentiments Using Vectorization Techniques. Algorithms, 17(11), 486.

[7]. Yendhe, Y. R. S., Kasturi, K. A. K., Jatar, J. R. S., & Patil, A. (2020). Fake News Detection and Sentiment Analysis in Twitter. International Journal of Advance Scientific Research and Engineering Trends (IJASRET), 5(9), 72.

[8]. N. Jadhav, P. More, A. Dixit, and A. Sharma, "Evaluating Public Opinion Through Twitter Sentiment Analysis," 2024 2nd International Conference on Networking, Embedded and Wireless Systems (ICNEWS), Bangalore, India, 2024, pp. 1-5.

[9]. C. G. Maurya and S. K. Jha, "Sentiment Analysis: A Hybrid Approach on Twitter Data," Proceedings of the International Conference on Machine Learning and Data Engineering (ICMLDE 2023), Procedia Computer Science, vol. XX, pp. XX-XX, 2023. Elsevier. DOI: 10.1016/j.procs.2024.04.094

[10]. Glazkova, A. (2023). A Comparison of Text Preprocessing Techniques for Hate and Offensive Speech Detection in Twitter. *Social Network Analysis and Mining*, 13, 155. https://doi.org/10.1007/s13278-023-01156-y.

[11]. Vidyashree, K. P., Rajendra, A. B., Gururaj, H. L., Ravi, V., & Krichen, M. (2024). A Tweet Sentiment Classification Approach Using an Ensemble Classifier. International Journal of Cognitive Computing in Engineering, 5, 170- 177.

[12]. E. Cano-Marin, M. Mora-Cantallops, and S. Sánchez-Alonso, “Twitter as a predictive system: A systematic literature review,” J. Bus. Res., vol. 157, p. 113561, 2023. doi: 10.1016/j.jbusres.2022.113561.

[13]. Padhy, M.; Modibbo, U.M.; Rautray, R.; Tripathy, S.S.; Bebortta, S. Application of Machine Learning Techniques to Classify Twitter Sentiments Using Vectorization Techniques. Algorithms 2024, 17, 486.

[14]. Shukla, D., & Dwivedi, S. K. (2024). The Study of the Effect of Preprocessing Techniques for Emotion Detection on Amazon Product Review Dataset. Social Network Analysis and Mining, 14, 191. https://doi.org/10.1007/s13278-024- 01352-4.

[15]. Padhy, M.; Modibbo, U.M.; Rautray, R.; Tripathy, S.S.; Bebortta, S. Application of Machine Learning Techniques to Classify Twitter Sentiments Using Vectorization Techniques. Algorithms 2024, 17, 486. https://doi.org/10.3390/a17110486.

[16]. Yadav, N., Kudale, O., Rao, A., Gupta, S., & Shitole, A. (2021). Twitter Sentiment Analysis Using Supervised Machine Learning. In J. Hemanth, R. Bestak, & J. Chen (Eds.), Intelligent Data Communication Technologies and Internet of Things (Vol. 57, pp. 589–598). Springer, Singapore. https://doi.org/10.1007/978-981-15-9509-7_51.

[17]. Ahmad, T.; Faisal, M.S.; Rizwan, A.; Alkanhel, R.; Khan, P.W.; Muthanna, A. Efficient Fake News Detection Mechanism Using Enhanced Deep Learning Model. Appl. Sci. 2022, 12, 1743. https://doi.org/10.3390/app12031743

[18]. Folino, F., Folino, G., Guarascio, M. et al. Towards Data- and Compute-Efficient Fake-News Detection: An Approach Combining Active Learning and Pre-Trained Language Models. SN COMPUT. SCI. 5, 470 (2024).

[19]. A. Altheneyan and A. Alhadlaq, "Big Data ML-Based Fake News Detection Using Distributed Learning," in IEEE Access, vol. 11, pp. 29447-29463, 2023, doi: 10.1109/ACCESS.2023.3260763.

[20]. S. Kumar and B. Arora, "A Review of Fake News Detection Using Machine Learning Techniques," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1-8, doi: 10.1109/ICESC51422.2021.9532796.

[21]. R. R. Rajalaxmi, L. V. N. Prasad, B. Janakiramaiah, C. S. Pavankumar, N. Neelima, and V. E. Sathishkumar, “Optimizing hyperparameters and performance analysis of LSTM model in detecting fake news on social media,” Trans. Asian Low-Resour. Lang. Inf. Process., accepted Jan. 17, 2022.

[22]. S. Kumar and B. Arora, "A Review of Fake News Detection Using Machine Learning Techniques," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1-8, doi:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25jul670

10.1109/ICESC51422.2021.9532796.

[23]. R. R. Rajalaxmi, L. V. N. Prasad, B. Janakiramaiah, C. S. Pavankumar, N. Neelima, and V. E. Sathishkumar, "Optimizing hyperparameters and performance analysis of LSTM model in detecting fake news on social media," Trans. Asian Low-Resour. Lang. Inf. Process., accepted Jan. 17, 2022.