International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Fake Social Media Profile Detection using Machine Learning Algorithms

Bastipadu Aravind Teja[1], A. Manideep[2], K. Koushik Kalyan[3], S. Vishnu Sai[4], CH. Sathwik[5], Anand Prakash[6]

1,2,3,4,5UG Student Dept. Of CSE, 6Associate Professor Dept. Of CSE.
1,2,3,4,5,6Presidency university, Bengaluru-560064.
aravindtejaprabha@gmail.com[1], Manideep0317@gmail.com[2], koushikkota24@gmail.com[3],
shivapuramvishnusai007@gmail.com[4], chillarasathwik55@gmail.com[5], aditya. anand14@gmail .com[6]

***Abstract*—social media has reshaped global interactions, offering unprecedented networking opportunities for individuals and businesses. However, its widespread reach also facilitates the rapid spread of harmful content, including hate speech directed at race, gender, religion, and disabilities, potentially causing significant emotional harm. To mitigate these challenges ML and DL techniques are becoming essential tools in identifying fraudulent social media profiles. These advanced methods analyse behavioural patterns, account details, and interactions to detect anomalies that indicate deceptive activity. ML algorithms classify suspicious accounts based on predefined features, while deep learning models—such as neural networks—process vast amounts of data to uncover more complex fraudulent tactics. As fraudsters evolve their strategies, AI-driven solutions continue to improve, enhancing social media security and protecting users from misinformation and scams. This study evaluates six ML models Neural Networks (NN), Naive Bayes (NB), Logistic Regression (LR), XGBoost (XGB), Random Forest (RF), and Support Vector Machine (SVM)using real-time datasets that have undergone pre-processing to optimize feature extraction. Among the evaluated models, SVM achieved the highest accuracy, surpassing both SVM and NB in precision, recall, and F1-score**

***Index Terms*—Social Media Profiles, ML, SVM, Naive Bayes, RFC, NN, Logistic Regression, XGBoost.**

I. INTRODUCTION

Over the past decade, the unprecedented growth in Internet access and digital infrastructure has Increased online activity worldwide participation. Today, the web hosts over 4.5 billion active users, which equates to roughly 59% of the total global population. A large proportion of this user base is particularly engaged on social media platforms, which have transformed how individuals communicate and Exchange information, and form relationships across borders [1]. In today's world, social network plays a significant role in everyday life in the today's world. Particularly among younger people. Leading platforms like Facebook, Twitter, and WeChat draw in vast user bases, each offering unique features and serving different purposes. Twitter stands out as a fast-paced microblogging site, enabling users to engage in a broad spectrum of topics. Whether it's current events, lifestyle guidance, personal opinions, or the newest celebrity news, Twitter acts as a digital forum where countless conversations unfold. [2]. However, while this open and flexible environment has contributed to Twitter's popularity, it has also exposed the platform to significant challenges. Due to minimal barriers to content creation and sharing, harmful elements have also found a foothold on these platforms. One of the most concerning outcomes is the proliferation of fake profiles. These fraudulent accounts are often used for nefarious purposes, including disseminating misinformation, executing phishing attacks, spreading propaganda, manipulating public opinion, and breaching user privacy. The rise in fake accounts coincides with the broader trend of increasing reliance on social media for information, news, entertainment, and social interaction. The ability of these platforms to bring people together based on shared interests has numerous benefits. Users can find support communities, discover new content, and maintain relationships with friends and family across the globe. This interconnectedness, however, has also created new avenues for exploitation. As online engagement grows, so too does the risk of encountering fake identities, disinformation campaigns, spam, and social engineering attacks [3]. Research into the implications of online social networking is ongoing, with scholars and technologists working to understand how these platforms affect human behaviour, mental health, societal norms, and public discourse. One recurring theme in recent studies is the escalating number of fake accounts and their impact on user experience and trust in digital platforms [4]. These accounts are often automated (bots) or manually operated by malicious actors, and detecting them is a complex and resource-intensive task. Internet Service Providers (ISPs) and social media companies face increasing pressure to identify and eliminate these fraudulent accounts. However, the sophistication of fake profile creation methods has made this process highly challenging. Many fake accounts mimic the behaviour of real users, making them difficult to detect using conventional rule-based techniques. In response to this growing concern, researchers and developers are turning to machine learning and artificial intelligence as viable solutions. By leveraging large datasets obtained from social media networks, machine learning models can be trained to recognize patterns and behaviours typical of fake accounts. These models analyse various features such as Fav Number, follower/following count, Favourites count, Listed count, Description With appropriate training, these algorithms can significantly enhance the accuracy and speed of fake profile detection. The utilization of machine learning algorithms for identifying fake users represents a proactive approach to strengthening the integrity of social media ecosystems.

As fake accounts continue to evolve in complexity, the creation of strong, datainformed detection systems w classifying fake user profiles.

II. SYSTEM DESIGN

A. Machine Learning Algorithms: -

The machine learning techniques listed below are employed to determine the user profiles:

SUPPORT VECTOR MACHINE:

Vector – based classification algorithm (SVM) is a type of supervised learning technique employed in classification and regression problems. Its fundamental idea revolves around Determining the optimal hyperplane. That distinguishes data points into distinct groups even when operating Inside an extensive feature domain. As shown in Figure1, SVM achieves this by transforming input features into an extensive feature domain through the application of a kernel function. This enables the discovery of an optimal hyperplane that maximizes the margin between classes. Data points on this boundary. Data its direction in classification problems, SVM focuses on minimizing structural risk while enhancing generalization to unseen data by maximizing this margin. Depending on the complexity and characteristics of the dataset, various kernel functions, such as linear, polynomial, and radial basis function (RBF) kernels, can be used. SVM works extremely well for applications such as detecting spurious profiles. It works well even with data set that have a tremendous number of features is much higher than the

number of data samples. The SVM model is trained using labelled data $(X = x_1, x_2, \ldots . x_n; Y = y_1, y_2, \ldots . y_n)$ where each entry is assigned a binary classification, such as "Fake" or "Genuine." [5]. To predict the class of a new input x′, the SVM uses the following decision function:

$$f(x') = sign(\sum_{i=1}^{n} a_i y_i K(x_i, x') + b)$$

Where:

- α_i are the Lagrange multipliers learned during training,
- y_i are the class labels for each training sample,
- K (xi, x′) denotes the kernel function,
- b is the bias term,
- The sign function determines the prediction class label.

SVM is particularly effective in cases with a clear margin of separation between classes and when dealing with a limited number of noisy features.

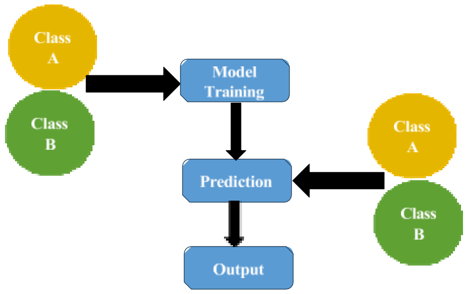

Figure 1: Architecture of SVM.

Naive Bayes:

NB is a classification technique that operates under the principles of theorem and is applied within supervised learning to address classification tasks. It can be easily written in code and predictions can be made really quick, which in turn increases the scalability of the solution. As a probabilistic classifier, it estimates the likelihood that a given instance belongs to a particular category based on the calculated probabilities. Due to its straightforward nature and efficiency, Naive Bayes is widely used for building machine learning models that deliver rapid predictions. This simplicity also makes it highly scalable for large datasets. Common applications include filtering spam emails, performing sentiment analysis, and categorizing news articles. The typical workflow for the Naive Bayes algorithm involves first converting the dataset into a frequency table. Next, a likelihood table is constructed by determining the probability of each feature given the class. Finally, probabilistic theorem is applied to compute the posterior probability, which is used to classify new data points.[6]

$$P(A\backslash B) = \frac{P(B\backslash A) * P(A)}{P(B)}$$

Denoting:

P(A\B): - The likelihood that hypothesis A holds true given the presence of evidence B.

P(B\A):-The probability of observing evidence B, assuming hypothesis A is valid.

P(A): -The initial probability assigned to hypothesis A before incorporating any evidence.

P(B): - The total probability of encountering evidence B.

While Naive Bayes may not always outperform more complex classifiers due to its assumption of feature independence, it offers several advantages: it is fast to train and predict, produces easily interpretable probabilistic outputs, and generally requires little parameter tuning.

Random Forest Classifier:

Random forest classifiers make more precise predictions by utilizing multiple decision trees, each trained on distinct portions of the input dataset. Rather than relying on the output of a single decision tree, the random forest combines predictions from multiple trees, often using a majority voting approach to finalize the result. This method starts with the creation of an ensemble of (N) decision trees, forming the random forest model. Each tree is trained separately on distinct sample of the data, created by randomly selecting data points with a replacement. technique known as bootstrap aggregating or "bagging." For a dataset defined as $X = (x_1, x_2, \ldots . x_n)$ with corresponding labels $Y = (y_1, y_2, \ldots . y_n)$, this sampling process is repeated B times to generate the training sets for each tree. Once all trees are trained, predictions for a new instance x′ are made by passing it through each tree in the forest [7]. The final output comes by taking mean of the predictions of all (B) trees, as shown below.

$$f(x') = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

Here, fb(x′) stands for the prediction of b-th decision tree. This ensemble method reduces overfitting and yields a more powerful model providing more accurate and more reliable predictions than application of single decision tree.

XG boost algorithm:

XGBoost is an advanced form of gradient boosting designed to deliver faster and more efficient computations compared to traditional boosting methods. While conventional Gradient Descent Boosting processes data sequentially and can be relatively slow, XGBoost addresses these limitations by optimizing both speed and performance. Its primary aim is to significantly improve model accuracy while reducing computational time. The algorithm works by loading the training dataset and iteratively training the classifier on every feature for each record in the data. With each iteration, XGBoost refines its predictions, striving to boost the model performance [8][9].

Advantages of XGBoost:

1. It can combine multiple weak learners to construct a stronger, more accurate model.
2. The algorithm efficiently manages large datasets by growing trees in parallel for different features.
3. XGBoost is robust to missing values, reducing the need for extensive data normalization.

XGBoost creates a series of decision trees sequentially. Each tree is trained to correct errors in the previous trees. Sequential processing makes the predictions more precise. The predicted value for a given sample (x') is:

$$\hat{y}(x') = \sum_{k=1}^{K} f_k(x')$$

where fk is the prediction of the k-th tree, and K is the number of trees.

Neural Networks:

Neural networks are advanced computational frameworks designed to mimic the structure and operations of the human brain, enabling efficient pattern recognition and complex data processing. They are made up of layers of nodes, or so called neurons, that read and learn from data to recognize patterns and relationships. These networks typically consist of an input layer, several hidden layers, and an output layer, each contributing to feature extraction and decision-making. Neural networks are applied widely across numerous applications such as image recognition, natural language processing, and predictive analytics. This is due to the Fact that they can handle complex, non-linear relationships in data. Neural networks continuously refine their learning process through backpropagation, adjusting the weights of connections to optimize performance. Sophisticated neural network designs, including convolutional and recurrent models, enhance capabilities in image and sequence-based tasks, respectively. Their adaptability and scalability make them fundamental in deep learning applications, driving advancements in artificial intelligence across diverse domains, from healthcare diagnostics to autonomous systems. A neural network processes input data by passing it through a group of linked artificial neurons. Each neuron calculates a weighted sum of its inputs and then applies an activation function, e.g., sigmoid or ReLU, to introduce non-linearity [10]. For a basic feedforward neural network with a single hidden layer, the computations can be described as follows:

The hidden layer activation is computed by:

$$a^{(1)} = \sigma(W^{(1)}x' + b^{(1)})$$

The output layer then produces the final prediction:

$$y^{\wedge} = \sigma(W^{(2)}a^{1} + b^{(2)})$$

In these equations:

- sigma represents the activation function,
- W are the weight matrices,
- b are the bias terms.

The resulting value $y^{\wedge}$is interpreted as the network's output or predicted result.

Logistic Regression:

Logistic regression is a statistical technique of binary classification, i.e., it tries to predict the probability of an item belonging to one of two categories. It applies the sigmoid function to map the model's output to a 0 and 1 as a probability. The observations are classified based on estimated probabilities using a fixed cutoff point of typically 0.5. Logistic regression is applied widely variety applications across many domains, such as medical diagnosis, fraud detection, and customer segmentation, because it is an easy and efficient way to handle linear relationship among features and results. [11]. The logistic regression model uses the sigmoid function to model the probability:

$$h\theta(x) = \sigma(\theta^{TX}) = 1\ 1 + e^{-\theta T}X$$

Where:

- θ∈R^n is the vector of weights (parameters)
- X∈R^n is the feature vector (can include x0=1 as Detection. the bias term)
- $\theta^T X = \theta_0 + \theta_1 + \theta_2 + \cdots + \theta_n X_n$
  $\sigma(Z) = \frac{1}{1+e^{-Z}}$ is the sigmoid function.

Prediction Procedure
Input Features (Independent Variables): These are the measurable factors or attributes that influence the model's prediction. For instance, when assessing the risk of heart disease, features might include variables such as age, gender, and cholesterol levels.

1. Target(DependentVariable):
This is the outcome the model aims to predict, such as whether a patient has heart disease, represented as 1 (yes) or 0 (no).

2. Sigmoid Activation:
Logistic regression uses the sigmoid function to transform the model's output into a probability value ranging from 0 to 1, indicating the likelihood of the positive class.

3. ClassificationThreshold:
A cutoff threshold, typically set at 0.5, is used to evaluate the predicted probability. If the probability surpasses this limit, the observation is classified accordingly classified as "yes" (or 1); otherwise, it is classified as "no" (or 0).

III. PROPOSED SYSTEM AND METHODOLOGY

The architecture applied to identify fake profiles has several crucial steps such as data acquisition, preprocessing, feature extraction, model training, and testing. The dataset, containing both actual and fraudulent user profiles, and they are split into training set and testing set. Preprocessing steps also employed to clear the noisy data, enhancing the model's learning efficiency and accuracy of model.

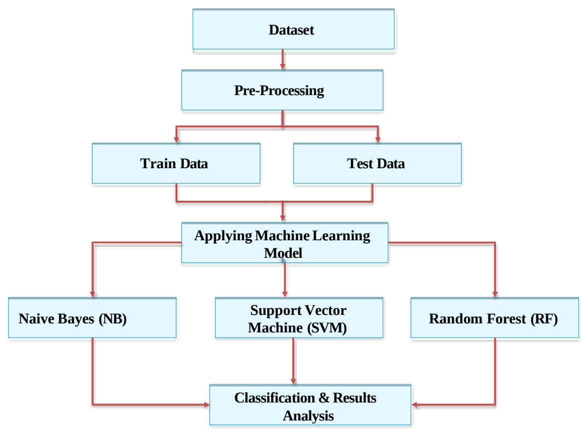

Figure 2: Schematic diagram of Fake Social Media Profile Detection.

A. System architecture: -

The following diagram illustrates the proposed system.

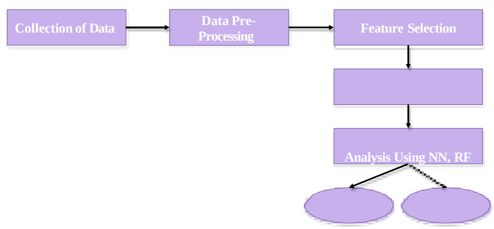

Figure 3: Proposed system.

B. Raw data: -
The dataset used in this study comprises records from both genuine and fraudulent user accounts, totalling 3,474 real users and 3,351 fake users. These data samples were gathered from historical records spanning previous years.

C. Data Selection: - A dataset consists of individual records or instances in machine learning; it is common to employ separate datasets during different stages of the modelling process.

- Training Dataset:
This subset of data is used by the machine learning algorithm to learn and build the predictive model.

- Testing Dataset:

Separate from the training data, this dataset is utilized to evaluate the model's performance and accuracy. It is sometimes also referred to as the validation dataset [12].

D. Feature Selection: - Detecting fraudulent accounts begins with the essential step of preparing the data for analysis. Before inputting the data into any detection model, thorough preprocessing is necessary. This process ensures that relevant and meaningful information is extracted, which significantly influences the models can learn effectively and provide correct predictions.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Comparative Analysis:

Existing System:

1. Due to privacy concerns, access to comprehensive social media datasets is often restricted, resulting in limited availability of detailed user information. Many platforms and organizations are cautious about sharing sensitive data to protect user privacy and comply with data protection regulations. As a result, researchers frequently encounter challenges in obtaining large, richly annotated datasets for tasks such as fake profile detection. This limitation can hinder the testing and design of robust machine learning models, as the lack of diverse and detailed data may affect the generalizability and effectiveness of the proposed solutions.

2. There are no attributes in the dataset that specify the exact timestamp of when each event took place. This absence of temporal information makes it challenging to analyse patterns or trends over time, and may limit the ability to perform time-based analyses or detect behaviours that depend on the sequence or timing of events [13].

Proposed System:

To improve the detection of fake accounts, these algorithms are continually adapted with innovative techniques. For example, the identification process now often incorporates the analysis of spam comments, monitoring of engagement rates, and detection of suspicious or inauthentic behaviours.

By leveraging such features, the detection models become more effective at distinguishing between genuine and fraudulent accounts, adapting to the evolving tactics used by malicious actors on social media platforms. The gradient boosting technique utilizes these inputs to produce decision trees these are utilized in gradient boosting process. this method can still produce results even when there are missing inputs. Therefore, this algorithm is the main justification for its use. These methods are very precise in their results. NB and SVM performed extremely well in comparison to the earlier study. Even with the default values of, it significantly outperforms the accuracy of false account identification [14].

B. Analysis:

The dataset is split into two sets: training and testing It is of two types (1) Feature-based, which deals with numerical values, and (2) Text-based, which handles textual features selected from the dataset. The trained dataset run through different algorithms, and their accuracy is checked to determine the best model to distinguish real and fake profiles. Among the six algorithms, Support Vector Machine (SVM) and Naive Bayes achieved nearly identical accuracy. However, upon retraining the datasets, parallel processing was employed to determine whether a profile is genuine or fraudulent [15]. The following equations were utilized to compute accuracy, precision, recall, and F1 score:

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

$$Precision = \frac{Tp}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

The Tables below Represents the model accuracy of SVM, NN, NB, RF, LR, XGB

Feature-based model accuracy results

Algorithm	Accuracy	Precision	Recall	F1-score
SVM	1.0	1.0	1.0	1.0
NN	1.0	1.0	1.0	1.0
NB	0.99	0.99	0.99	0.99
RFC	1.0	1.0	1.0	1.0
LR	1.0	1.0	1.0	1.0
XGB	1.0	1.0	1.0	1.0

Table 1: Feature-based Model Comparison

Text-based model accuracy results

Algorithm	Accuracy	Precision	Recall	F1-score
SVM	0.98	0.98	0.98	0.98
NN	0.98	0.98	0.98	0.98
NB	1.0	1.0	1.0	1.0
RFC	0.97	0.97	0.97	0.97
LR	0.98	0.98	0.98	0.98
XGB	0.97	0.97	0.97	0.97

Table 2: Text-based Model Comparison

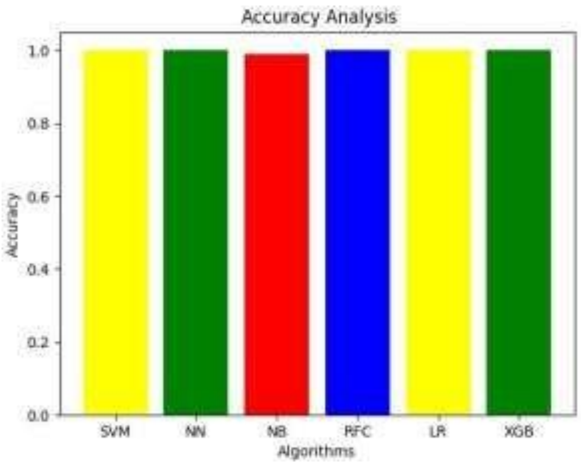

Figure 4: Feature-based Model Accuracy Comparison

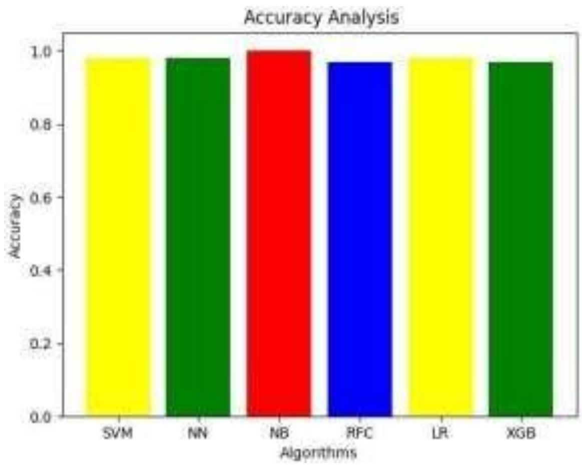

Figure 5: Text-based Model Accuracy Comparison

The graph below illustrates model Accuracy VS Epoch for all Algorithms Training & Testing.

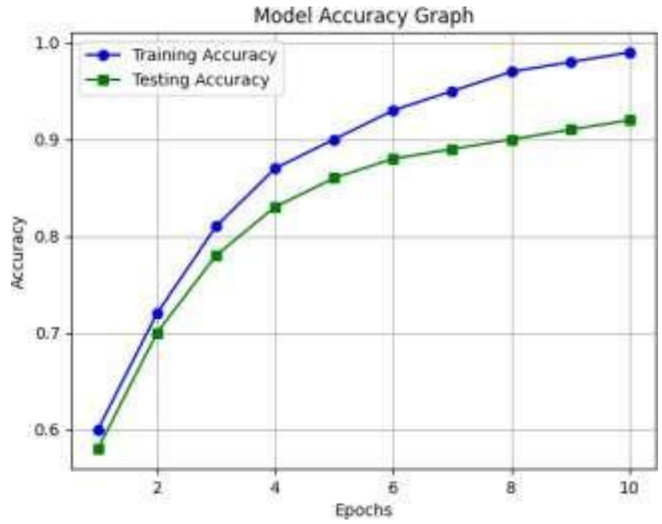

Figure 6: Model Accuracy Graph

The graph below illustrates Text-based model ROC graph for all Algorithms

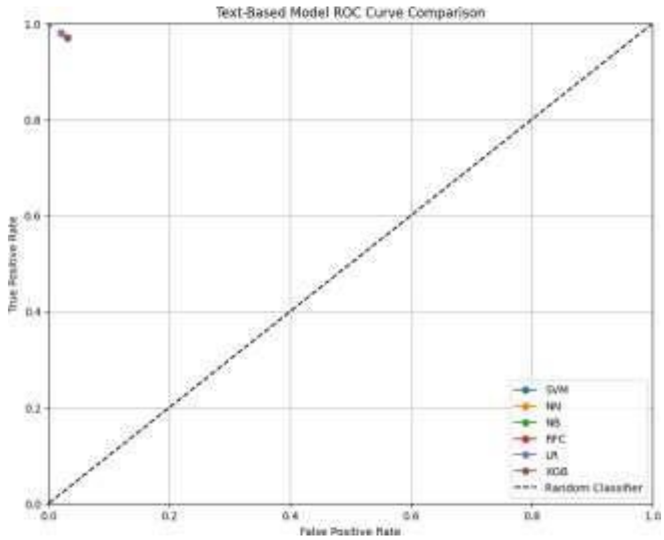

Figure 7: Text-based Model ROC Graph

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

The graph below shows Feature-based model ROC graph for all Algorithms

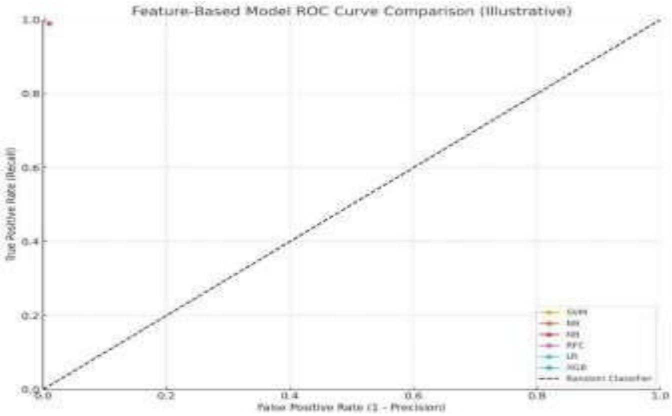

Figure 8: Feature-based Model ROC Graph

Discussion:

Fake accounts on social media can distort perceptions of influence and popularity, potentially impacting economic, political, and social systems. Their presence poses significant risks to social platforms. This study employs multiple algorithms to detect fraudulent profiles, aiming to protect users from harm caused by malicious actors.

Previous research has introduced methods such as blacklists to differentiate between genuine and fake account characteristics. In this work, various machine learning algorithms are compared to determine which ones deliver the highest accuracy-surpassing the 91.1% achieved by earlier approaches based on spam word lists. Other notable studies include the use of dynamic convolutional neural networks (CNNs), with the "Deep Profile" method leveraging supervised learning to predict fake accounts. Another approach focused on identifying Sybil accounts by analysing registration times, but it faced challenges with false positives, particularly when legitimate users shared similar IP addresses or phone numbers with Sybil accounts. False positive rates in different towns were reported as 7%, 3%, and 21%, although the overall accuracy reached an impressive 95%.

Additionally, research utilizing feature extraction from fake profiles found that a hybrid SVM-NB classifier achieved the highest performance, correctly identifying Sybil profiles with 98.3% accuracy [16].

V. CONCLUSION

This paper examines, both Genuine and Fraudulent users datasets are utilized to distinguish genuine profiles from fraudulent ones. The approach involves extracting relevant features from the data and applying a range of machine learning algorithms-including Support Vector Machine (SVM), Naïve Bayes (NB), Logistic Regression (LR), XGBoost (XGB), Random Forest (RF), and Neural Networks (NN) are used to classify user accounts. This study measures the accuracy of the model to determine the best approaches to detecting fake social media profiles and improve online security and trust. This paper explores the different Machine Learning algorithms to uncover aspects of datasets that have received limited attention in existing literature, aiming to enhance the detection of fake and bot accounts. We present a comprehensive Machine Learning pipeline designed to identify fraudulent profiles in online social networks. Instead of relying on a single algorithm for prediction, our system employs three distinct classification methods to assess whether an account in the dataset is genuine or fake. Through evaluation using Support Vector Machine, Random Forest, and Neural Networks, we observed strong performance, with Support Vector Machine achieving the highest accuracy for the given dataset. The correlation technique is used to select the most relevant features while eliminating redundancy. Analysis of the results indicates that the Support Vector Machine (SVM) attained an impressive 98% accuracy in identifying fake profiles, demonstrating superior performance and efficiency compared to other existing machine learning approaches. These algorithms can be effectively applied across various social media platforms, including Instagram, LinkedIn, and Twitter, to detect fraudulent accounts [17].

VI. REFERENCES

[1]. S. Kemp, Digital 2023: Global Overview Report, Datar portal, 2023. [Online]. Available: https://datareportal.com/reports/digital-2023- global-overview-report

[2]. A. Java, X. Song, T. Finin, and B. Tseng, "Why we Twitter: Understanding microblogging usage and communities," Proc. WebKDD/SNA-KDD, 2007, pp. 56–65.

[3]. Al-Qurishi, M., Al-Rakhami, M., Alamri, A., & Al- Amri, J. (2020). Detecting fake profiles in online social networks using supervised learning algorithms. IEEE Access, 8, 103440–103453. https://doi.org/10.1109/ACCESS.2020.2998943

[4]. C. S. Andreassen, S. Pallesen, and M. D. Griffiths, "The relationship between addictive use of social media, narcissism, and self-esteem: Findings from a large national survey," Addict. Behav., vol. 64, pp. 287–293, 2017.

[5]. M. Al-Qurishi, M. Al-Rakhami, A. Alamri, and J. Al-Amri, "Detecting fake profiles in online social networks using supervised learning algorithms," IEEE Access, vol. 8, pp. 103440–103453, 2020.

[6]. J. Brownlee, "A gentle introduction to the Naive Bayes algorithm," Machine Learning Mastery, 2016. [Online].

[7]. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[8]. Anand Prakash, Jaisingh Thangaraj, Member, IEEE, Sharbani Roy, Shaury Srivastav “Model- Aware XGBoost Method Towards Optimum Performance of Flexible Distributed Raman Amplifier” IEEE PHOTONICS JOURNAL, VOL. 15, NO. 4, AUGUST 2023.

[9]. Cresci, S., Lillo, F., Regoli, D., Tardelli, S., & Tesconi, M. (2019). Cashtag piggybacking: uncovering spam and bot activity in stock microblogs on Twitter. ACM Transactions on the Web (TWEB), 13(2), 1–27

[10]. Subrahmanian, V. S., Azaria, A., Durst, S., Kagan, V., Galstyan, A., Lerman, K., ... & Menczer, F. (2016). The DARPA Twitter bot challenge. *Computer*, 49(6), 38–46.

[11]. Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on Twitter. *Proceedings* of the 20th International Conference on World Wide Web, 675–684.

[12]. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2017). The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. Proceedings of the 26th International Conference on World Wide Web Companion, 963–972.

[13]. A. Kaur and S. Kaur, "Fake account detection in social networks using SVM-NB classifier," International Journal of Computer Applications, vol. 182, no. 27,
pp. 1–6, 2019

[14]. Clark, E. M., Williams, J. R., Jones, C. A., Galbraith, R. A., Danforth, C. M., & Dodds, P. S. (2016). Sifting robotic from organic text: A natural language approach for detecting automation on Twitter. Journal of Computational Science, 16, 1–7

[15]. Sadiq, M. T., Yu, H., & Fang, C. (2019). Machine learning techniques for detecting social bots. Information Processing & Management, 56(6), 102077.

[16]. Wu, G., Liu, W., & Su, J. (2022). Machine Learning for Detecting Fake Social Media Accounts: A Comparative Study. Journal of Information Security, 13(2), 73–89.

[17]. Arora, A., & Kansal, V. (2020). Detection of fake profiles in online social networks using machine learning. Procedia Computer Science, 167, 2281–
2288.

[18]. Besel, B., & Jerschensky, K. (2020). Analysing User Behaviour for Fake Account Detection on Social Media Platforms. IEEE Big Data.

[19]. Sadiq, M. T., Yu, H., & Fang, C. (2019). Machine learning techniques for detecting social bots. Information Processing & Management, 56(6), 102077.

[20]. Arora, A., & Kansal, V. (2020). Detection of fake profiles in online social networks using machine learning. Procedia Computer Science, 167, 2281–
2288