EAI Endorsed Transactions

on Scalable Information Systems Research Article EAI.EU

Detecting Fake Social Media Profiles Using the Majority Voting Approach

Dharmaraj R. Patil*[,1],Tareek M. Pattewar[2], Vipul D. Punjabi[3] and Shailendra M. Pardeshi[4]

[1]Department of Computer Engineering, R.C. Patel Institute of Technology, Shirpur, India
[2]Department of Computer Engineering, Vishwakarma University, Pune, India
[3]Department of Computer Engineering, R.C. Patel Institute of Technology, Shirpur, India
[4]Department of Computer Science & Engineering (Data Science), R.C. Patel Institute of Technology, Shirpur, India

Abstract

INTRODUCTION: The rise of social media platforms has brought about a concerning surge in the creation of fraudulent user profiles, with intentions ranging from spreading false information and perpetrating fraud to engaging in cyberbullying. The detection of these deceptive profiles has emerged as a critical imperative to safeguard the trustworthiness and security of online communities.
OBJECTIVES: This paper focused on the detection and identification of fake social media profiles.
METHODS: This paper introduces an innovative approach for discerning and categorizing counterfeit social media profiles by leveraging the majority voting approach. The proposed methodology integrates a range of machine learning algorithms, including Decision Trees, XGBoost, Random Forest, Extra Trees, Logistic Regression, AdaBoost and K-Nearest Neighbors each tailored to capture distinct facets of user behavior and profile attributes. This amalgamation of diverse methods results in an ensemble of classifiers, which are subsequently subjected to a majority voting mechanism to render a conclusive judgment regarding the legitimacy of a given social media profile.
RESULTS: We conducted thorough experiments using a dataset containing both legitimate and fake social media profiles to determine the efficiency of our methodology. Our findings substantiate that the Majority Voting Technique surpasses individual classifiers, attaining an accuracy rate of 99.12%, a precision rate of 99.12%, a recall rate of 99.12%, and an F1-score of 99.12%.
CONCLUSION: The results show that the majority vote method is reliable for detecting and recognising fake social media profiles.

Received on 29 October 2023; accepted on 13 February 2024; published on 14 February 2024

Keywords: Fake social media profiles, social media security, majority voting approach, machine learning, user behavior analysis.

doi:10.4108/eetsis.4264

1. Introduction

In the digital age, social media platforms have become an integral part of our daily lives, serving as arenas for communication, information sharing, and community building. However, the unrestricted accessibility of these platforms has given rise to a pressing concern, the proliferation of fake social media profiles. These deceptive profiles, often created with malicious intent, pose significant threats to the integrity, security, and trustworthiness of online communities. Fake social media profiles are used for a myriad of nefarious purposes, including spreading misinformation, perpetrating scams, engaging in cyberbullying, and conducting online fraud. As these profiles become increasingly sophisticated and convincing, the challenge of identifying and mitigating them has grown substantially. Traditional methods of manual inspection and rule-based systems are no longer sufficient to combat the sheer volume and sophistication of these fraudulent accounts. According to Statista's Q4 2022 report on Facebook's fake account removal, Facebook initiated

*Corresponding author. Email: dharmaraj.rcpit@gmail.com

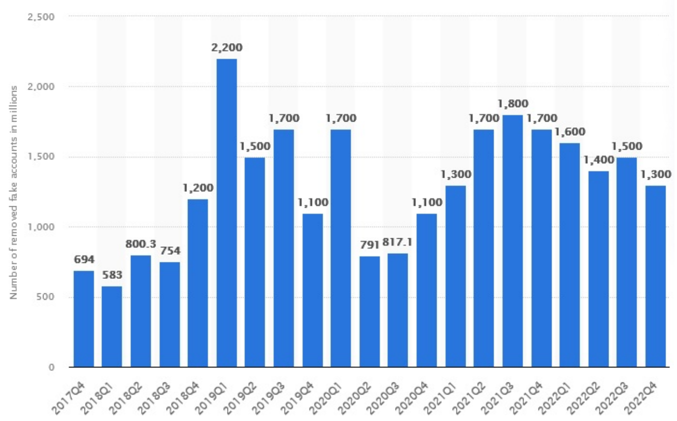

Figure 1. Global statistics on Facebook's actions against fake accounts from the fourth month of 2017 to the fourth month of 2022 [1].

measures against 1.3 billion fraudulent accounts in the final quarter of 2022, marking a decrease from 1.5 billion in the previous quarter. Notably, in the first quarter of 2019, the social media platform eliminated an unprecedented 2.2 billion spurious profiles. Meta defines spurious accounts as those created with malicious intent or with the intention to represent a company, organization, or non-human entity. Figure 1 depicts global statistics on Facebook's actions against fake accounts from the fourth month of 2017 to the fourth month of 2022 [1].

Twitter discloses approximate percentages of fraudulent, spam, and automated accounts in its securities filings. According to Twitter Inc.'s 2013 Annual Report, "fake or spam accounts accounted for less than 5%." It's worth noting that this percentage differs from the 11% of active users who utilize 'applications that automatically communicate with our servers for frequent updates without requiring user interaction.' Whether these numbers overlapped or were related solely to automated accounts remained unclear. The 5% statistic was reiterated in the following year's annual report, along with the same cautions, while the 11% figure was reduced to 8.5%. These 5% and 8.5% statistics were consistently reiterated in Twitter Inc.'s annual reports for the fiscal years ending in December 2015, 2016, and 2017. However, from 2018 onwards, Twitter discontinued the provision of the second statistic but continued to report the 5% figure [2]. On the other hand, various social media platforms like Snap and LinkedIn acknowledge the presence of duplicate, numerous, or inauthentic users on their platforms but refrain from providing specific numerical figures [3, 4].

This study delves into the pressing concern of identifying fake social media profiles using an innovative application of the Majority Voting Technique. Our methodology leverages the capabilities of machine learning and ensemble classifiers to analyze user behavior, profile attributes, and network connections, enabling us to make informed assessments regarding the authenticity of social media profiles. By amalgamating multiple classifiers, each with its own distinct strengths and capabilities, and subjecting their decisions to a majority voting mechanism, we create a resilient and scalable solution for distinguishing real user profiles from their counterfeit counterparts.

The primary goal of this study is to present and evaluate the efficiency of the Majority Voting Technique in detecting phony social media profiles. Our objective is to demonstrate that our approach surpasses conventional methods and individual classifiers, achieving an accuracy of 99.12%, precision of 99.12%, recall of 99.12%, and an F1-score of 99.12%. Our experimental findings signify a significant stride toward bolstering the credibility of social media platforms and ensuring the safety of their users. The major contributions of our paper are as below,

1. Presenting an innovative approach for detecting fraudulent social media profiles that harnesses the Majority Voting Technique. This technique amalgamates diverse machine learning algorithms, encompassing natural language processing, image analysis, and network-based features, within an ensemble framework to enhance the precision of fake profile identification.

2. Validating the Majority Voting Technique's constant dominance over individual classifiers, as indicated by its extraordinary efficiency of 99.12%, precision of 99.12%, recall of 99.12%, and F1-score of 99.12%. This increased precision offers a more reliable and effective strategy for detecting fake social media profiles in the online world.

3. Offering a scalable solution capable of efficiently scrutinizing a large quantity of social media profiles, rendering it apt for implementation on popular social media platforms with extensive user bases.

4. Providing a practical and actionable remedy that social media platforms and online communities can adopt to enhance their security measures and safeguard their user communities.

The rest of this paper is structured as follows: In Section 2,we have provided the motivation of this study. Section 3 gives a concise overview of relevant prior research. Section 4 outlined the methodology involving feature extraction, pre-processing and supervised batch machine learning algorithms. The findings and discussions from our experiments are detailed in

Section 5. Our final conclusions are presented in Section 6.

2. Motivation

The spread of social media platforms in recent years has completely changed networking, communication, and the sharing of information. However, in addition to social media's advantages, the number of fake accounts on the platform has increased dramatically. These fake individuals, which are frequently run by bots or malicious individuals, have the potential to distribute spam, dangerous content, and false information, raising major social media and cybersecurity issues. The identification of fraudulent social media profiles presents a notable obstacle since malicious actors deploy intricate strategies to imitate authentic user conduct. These fake profiles are difficult to recognise using conventional detection techniques that rely on manual inspection or basic rule-based algorithms, especially as they get more complex.

Furthermore, the development of effective fake profile detection systems holds profound implications for various stakeholders. Social media platforms can mitigate the spread of misinformation and malicious content, thereby fostering a safer and more trustworthy online environment for users. Additionally, businesses and advertisers can protect their brands and investments by ensuring that their interactions on social media platforms are with genuine users rather than bots or fraudulent accounts. Overall, the motivation behind this paper is to address the escalating threat posed by fake social media profiles and contribute to the development of robust detection mechanisms that safeguard the integrity and security of online social networks.

3. Related works

Numerous researchers have explored the realm of detecting fraudulent social media profiles, employing a range of techniques and methodologies to counter the rise of misleading profiles. This overview offers insight into related work within this domain and acknowledges the contributions of notable authors in this field.

Ramalingam, D., et al. have provided a comprehensive assessment of major strategies for detecting false profiles in Online Social Networks. They delves into both historical methods and contemporary state-of-the-art approaches for identifying Sybil or counterfeit accounts within social networks. These various approaches, alongside their corresponding synthetic network types and dataset statistics, are meticulously compared and presented in tabular form. The focus of the review extends to recently proposed schemes, where their strengths and limitations are assessed, and comparisons are made based on qualitative performance [5].

Ezarfelix, J. et al. Have offered an in-depth review of multiple approaches to machine learning used for detecting bogus profiles on social media networks. Their work also presents the results of a thorough analysis that draws on existing literature to determine the most efficient approach. Through an exhaustive series of assessments and evaluations, their results consistently emphasize the effectiveness of neural networks as the superior method for detecting counterfeit accounts [6].

Kaubiyal, J. et al. have utilized a feature-based strategy to enhance the detection of fraudulent profiles on social media platforms. By employing a set of twenty-four unique features, they attain precise identification. To substantiate their classification outcomes, they applied three classification algorithms. Their experiments unveiled remarkable performance, with the Random Forest algorithm achieving an impressive accuracy rate of 97.9%. This approach, therefore, proves to be a proficient and potent method for identifying counterfeit profiles [7].

Mohammadrezaei, M. et al. have introduced an innovative model that capitalizes on the concept of user friend network similarity to unveil counterfeit accounts within social networks. Using the adjacency matrix of the relevant social network graph, they compute many matching measures, including common friends, cosine, Jaccard, L1-measure, and weight matching. To assess the efficiency of their model, they conducted a comprehensive evaluation using a Twitter dataset. Their findings show that the Medium Gaussian SVM method has exceptional predictive skills, with an AUC score of 1 and a significantly small rate of false alarms of 0.02 in fraudulent account identification [8].

Joshi, U. D et al. have conducted an in-depth exploration of models tailored for discerning fake profiles on Twitter. Their major focus is on identifying between genuine and fake accounts using observable factors like as follower count, friend count, status updates, and more. These models are crafted through a diverse array of machine learning techniques. The study made use of datasets such as MIB for real accounts, TFP and E13 for counterfeit accounts, and INT, TWT, and FSF for legitimate accounts. The experiments included Neural Networks, Random Forest, XGBoost, and LSTM, among other learning algorithms. By thoughtfully selecting key features to assess the legitimacy of social media profiles, they established a binary output of 0 for authentic profiles and 1 for fake ones. Impressively, their models achieved an accuracy rate of 99.46% with XGBoost and 98% with Neural Networks. The identification of fraudulent profiles serves the crucial purpose of proactively mitigating potential cybersecurity threats through measures such as blocking or deletion [9].

Ajesh, F., Aswathy et al. have proposed solutions to improve the accuracy of fake profile identification using Artificial Intelligence and Natural Language Processing approaches. They classified profiles into genuine and counterfeit segments using the Random Forest Classifier, Support Vector Machine, and Optimized Nave Bayes method. This automated identification tool exhibits scalability, rendering it particularly suitable for online social networks with a substantial number of unverifiable accounts. These three algorithms excel at distinguishing between authentic and deceptive profiles, showcasing robust performance, even with their modest feature requirements, and accurately classifying approximately 98% of the profiles within their training dataset [10].

Khaled, S. et al. have unveiled an innovative algorithm named SVM-NN, which is tailored to efficiently detect fake Twitter accounts and bots. During its development, they incorporated feature selection and dimension reduction methods. Their strategy relies on machine learning classification algorithms, encompassing Support Vector Machine (SVM), Neural Network, and their groundbreaking SVM-NN algorithm. Notably, the SVM-NN algorithm, even with a reduced set of features, consistently attains an impressive classification accuracy of around 98% for the profiles in their training dataset [11].

Xiao, C. et al. have introduced a scalable approach for identifying networks of fake accounts created by the same entity. A supervised ML process designed to categorize large clusters of identities as either fraudulent or authentic is at the heart of their technique. The model relies on crucial features derived from user-generated text fields, including details such as name, email address, company, or university. These features encompass patterns within the cluster and comparisons of text frequencies throughout the whole user population. This framework was used to examine LinkedIn account data classified by signup IP address and date. During thorough testing, their model attained a stunning AUC of 0.98 on a held-out test set and 0.95 on out-of-sample testing data. This model has been seamlessly integrated into production and has effectively identified over 250,000 fake accounts since its deployment [12].

Kodati, S. et al. have put forward an innovative strategy for spotting fraudulent profiles on Twitter by utilizing a hybrid Support Vector Machine (SVM) algorithm. Their approach relies on machine learning and the hybrid SVM algorithm to categorize Twitter profiles as either fake or authentic. They have integrated dimension reduction techniques, feature selection, and bot analysis into their methodology. Remarkably, their hybrid SVM algorithm effectively uses a reduced set of features, yet it consistently attains an impressive 98% accuracy in the classification of accounts [13].

Sahoo, S. R. et al. Have proposed a ML-based strategy to detecting possibly suspect identities on Facebook who are manipulating or contaminating multi-media large data. Multimedia massive data refers to datasets that are differentiated by their variety, human-centric content, and large volumes of media-related information, such as written material, audio recordings, and video recordings, which are generally created inside various online social networks. Their investigations show that combining based on material and profile-based characteristics resulted in outstanding results, outperforming competing techniques [14].

Akyon, F. C. et al. have introduced two datasets specifically designed for identifying fake and automated accounts. They used algorithms based on ML such as Naive Bayes, Logistic Regression, Support Vector Machines, and Neural Networks to detect these accounts. Moreover, to address the challenge of detecting automated accounts, they have proposed a cost-sensitive genetic algorithm to handle the inherent bias in the dataset. Additionally, to mitigate the imbalance issue within the fake account dataset, they've implemented the Smote-nc algorithm. Their efforts have yielded impressive results, with classification accuracies of 86% for the automated account detection dataset and 96% for the fake account detection dataset [15].

Bordbar, J. et al. have proposed a unique strategy for detecting false user accounts on Twitter datasets by measuring similarity measurements across users and applying the Generative Adversarial Network algorithm. The outcomes of this method demonstrate an impressive accuracy rate, reaching 98.1% for the classification and detection of fake user accounts [16].

Samreen, S et al. have employed Artificial Neural Networks (ANN) to distinguish between authentic and counterfeit social network account details. Their approach involves training the ANN algorithm using existing datasets comprising both fake and genuine user account information. Subsequently, when presented with new test data for prediction, the trained model is applied to determine whether the provided account details belong to genuine or fake users [17].

Ali, A. K. et al. have introduced a mechanism for the identification of fake accounts on the social media platform Twitter. Their approach involves two key data preprocessing methods for extracting the most influential features: the coefficient and the test. To conduct the classification, they've harnessed supervised ML algorithms within an ensemble system, employing the stack method. In the first level of the stack, they've utilized Random Forest, Support Vector Machine, and Naive Bayes algorithms, The LG algorithm acts as the meta classifier. Their stack ensemble system has shown to be incredibly successful, exceeding the individual algorithms it integrates. It has achieved an impressive data accuracy rate of 99% [18].

Muñoz, S. D. et al. have introduced an approach for identifying counterfeit profiles on social media platforms. They accomplished this by applying various machine learning detection techniques to a newly created dataset. This dataset was specifically designed, encompassing 17 metadata features from both authentic and fake profiles, and it was put to the test using Instagram profiles. Through the utilization of various machine learning algorithms, they achieved a commendable detection rate of nearly 96%, all while maintaining favourable false positive rates [19].

Rostami et al. have undertaken the task of detecting fake accounts through the utilization of a multi-objective hybrid feature selection approach aimed at optimizing classification performance. To begin, they picked a potential feature set based on its strong relationship to the target class and low duplication across features, as obtained by the Minimum duplication - Maximum Relevance method. The final attribute set for the identification operation was a robust set of attributes with the fewest characteristics necessary to achieve the highest level of performance. This strategy was tested with data from Twitter's social networking network. The outcomes were then compared to those obtained by known efficient existing techniques. According to the results, the suggested classifier strategy beats the current approaches in terms of performance [20].

Kadam, N. et al. have To demonstrate the suggested notion of ML-based false news identification, an experiment involving the detection of fake accounts using a Twitter dataset was carried out. Their model encompasses a preprocessing phase aimed at refining the contents and attributes, thus enhancing dataset quality and reducing data dimensions. They then used five common ML techniques to detect bogus profiles. The evaluation of the system was carried out under two scenarios, with training and testing samples divided into ratios of 70-30% and 80-20%. This was achieved using 4-fold cross-validation. The results of their study indicate that the 80-20% sample ratio reduces resource consumption, while the 70-30% ratio leads to improved classification accuracy [21].

Al-Qurishi, Muhammad et al. have introduced an innovative and robust centralized key management protocol designed to establish secure communication services within Online Social Networks (OSNs). At the heart of this approach is the concept of a 'roadblock,' which all prospective group members must navigate. This 'roadblock' involves a task that only a human user can successfully complete, effectively preventing automated or controlled accounts from joining. As a result, the group consists solely of confirmed genuine users. This mechanism exhibits exceptional efficiency in detecting bot accounts, thereby enhancing the network's defenses against malicious activities conducted by counterfeit accounts [22].

Conti, Mauro et al. have suggested a potential technique to handle the potential danger of the False Account Attack, in which a rival attempts to mimic a victim on an Online social media platform when the victim does not already have a profile. Notably, their study is the first attempt to examine social-media structures from a dynamic standpoint in the context of privacy threats [23].

Gurajala, Supraja et al. undertook an investigation of sixty-two million openly accessible Twitter accounts belonging to users in order to develop an approach for the detection of automatically created false profiles. They used a pattern-matching algorithm based on display names, as well as an examination of tweet modify times, to discover a remarkably consistent sample of bogus user accounts. When compared to a ground truth dataset, further study into creating profiles schedules and URLs connected with these fraudulent accounts revealed different behavioural patterns of these fraudulent individuals. The combination of this approach with current social network analysis techniques has the potential to improve the identification of fraudulent profiles in online social media platforms in the quickest and most efficient way [24, 25].

Meligy, Ali M. et al. have introduced a detection technique named the Fake Profile Recognizer (FPR), designed to verify profile identities and identify fake profiles within Online Social Networks (OSNs). The detection method in their proposed technique relies on the utilization of Regular Expression (RE) and Deterministic Finite Automaton (DFA) approaches. Their proposed detection technique underwent evaluation using three different types of OSN datasets from Facebook, Google+, and Twitter. The results demonstrated a high level of precision, recall, accuracy, and notably low False Positive Rates (FPR) in the detection of fake profiles across all three datasets [26].

Jin, Lei et al. have conducted an analysis and in-depth characterization of the behaviours of Identical Copied Accounts (ICAs). Subsequently, they introduced a detection framework with a primary emphasis on the discovery and validation of suspicious identities. In the pursuit of identifying such identities, they proposed two distinct approaches, one based on attribute similarity and the other on the resemblance of social networks. The first technique handles a simpler case involving common friends inside friend networks, whilst the second is intended to handle instances with similar friend IDs. The study also includes experimental data that demonstrate the adaptability and efficacy of the offered methodologies. Finally, they investigate viable options for confirming these suspect IDs [27].

Nagariya, Himanshi Gupta et al. undertook a comparative analysis of eight different combinations of classification algorithms and assessed their accuracy using an Online Social Network dataset. The combinations included Random Forest, Support Vector Machine, Logistic Regression, KNN, and Decision Trees. After conducting a thorough evaluation of the outcomes of each hybrid approach, they determined that the most favourable accuracy was achieved when combining Support Vector Machine (SVM), Logistic Regression, and Neural Network. Consequently, they proposed a model for the detection of fake accounts, leveraging this hybrid approach that demonstrated the highest accuracy among all the considered combinations [28].

Uppada, Santosh Kumar et al. have introduced the SENAD methodology, which focuses on determining the legitimacy of news pieces published on Twitter, was introduced. This conclusion depends on the credibility and influence of the users who interact with these articles. The SENAD model adds a novel notion known as the genuineness score and takes into consideration a variety of user-centric social interaction metrics such as the Following-followers ratio, account age, and bias. Their suggested approach significantly improved the identification of fraudulent news and accounts, resulting in a classification accuracy of 93.7%. They also identified the importance of visuals coupled with textual data in the spread of fake news and developed the Credibility Neural Network architecture. This framework makes use of the geographic characteristics of Convolutional Neural Networks to detect physical changes in pictures and determine if the picture conveys a negative attitude, as fraudulent images frequently display one or both of these qualities. Their hybrid technique, which combines Error Level Analysis and sentiment analysis, is critical in detecting false photos, with an accuracy rate of about 76% [29].

To fill up the gap in the relationship between reviews and their neighbours, Jiahua Du et al. have presented the first end-to-end neural architecture. Twelve (three selection × four aggregation) schemes are possible with their model, which situates a review within the context hints that it has learnt from its neighbours. Utilising a number of cutting-edge benchmarks, they have assessed model across six categories of actual internet evaluations. Their model beats the baselines by 1% to 5%, as demonstrated by experimental data that validate the impact of sequential neighbours on reviews. By doing a thorough analysis, they were also able to uncover the ways in which reviews' neighbours affect how useful they are perceived [30].

Jiahua Du et al. have introduced a novel deep interactive architecture designed to capture the intricate interplay between text and ratings, known as text-rating interaction (TRI), for the purpose of modeling review helpfulness. By incorporating TRI, the model not only expands the representation capacity of star ratings but also amplifies the impact of rating information on review texts. The effectiveness of TRI was assessed across six real-world domains using the Amazon 5-Core dataset. Through extensive experimentation, the study demonstrates that TRI outperforms existing methods in predicting review helpfulness. Additionally, the researchers conducted ablation studies and qualitative analyses to gain deeper insights into the behavior of the model and the parameters learned during training [31].

Yuzhong Zhou et al. have introduced convolutional neural networks (CNNs) as part of a typical deep learning architecture tailored for knowledge service systems. Subsequently, CNNs were applied to facilitate knowledge classification through deep learning techniques. The study culminated in the presentation of simulation results derived from the knowledge service system, aimed at validating the efficacy of the proposed approach outlined in the paper [32].

Shiva Shankar Reddy et al. have conducted an analysis of various algorithms alongside Extreme Learning Machine (ELM) and Gradient Boosting techniques. To ensure robust performance, the study employed k-fold cross-validation with values of k set to 3, 5, and 10. The implemented algorithms included Naive Bayes classifier, Support Vector Machine, K-Nearest Neighbour, ID3, CART, and J48, while Gradient Boosting and ELM were proposed as additional approaches. All algorithms were implemented using R programming. Evaluation metrics such as accuracy, kappa statistic, sensitivity/recall, specificity, precision, f-measure, and AUC were employed to compare their performance. Notably, Gradient Boosting Machine (GBM) exhibited superior performance compared to existing algorithms. Subsequently, GBM was compared with ELM, wherein GBM demonstrated superior efficacy. Based on these findings, it is recommended to leverage the gradient-boosting algorithm for effective prediction of gestational diabetes [33].

4. Methodology

Figure 2 shows the framework of our social media fake profile detection system. It consists of phases like feature extraction, pre-processing, machine learning training and testing. These phases are explained as below.

4.1. Dataset

We used a MIB dataset that included 3,474 authentic accounts and 3,351 fraudulent profiles. For legitimate accounts, we specifically chose the following information: TFP and E13 were used for real accounts, whereas INT, TWT, and FSF were used for false accounts. The

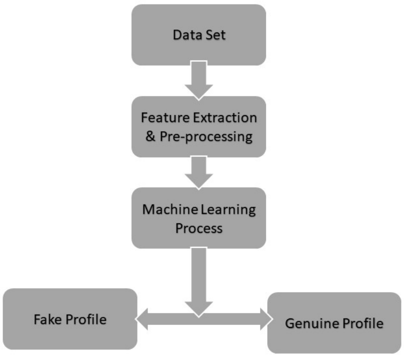

Figure 2. Social Media Fake Profile Detection Framework

information was saved in CSV format to make it easier for machines to read. The x-axis labels reflect the characteristics selected throughout the phase of pre-processing for use in the false profile detection procedure. The y-axis, on the other hand, represents the number of entries linked with each characteristic in the dataset [34].

4.2. Feature Extraction Phase

Only the numeric data was chosen for inclusion, while the categorical attributes were excluded. We have used eight features like, statuses_count, followers_count, friends_count, favourites_count, listed_count, default_profile, geo_enabled and profile_use_background_image [35]. Subsequently, the datasets containing fake and genuine user profiles were combined into a unified dataset, each profile being assigned an additional label ’is Fake,’ represented as a Boolean variable. This combined dataset is stored in the variable ’Y,’ which represents the response variable for a given profile ’X.’ Finally, any empty or ’NaN’ entries were replaced with zeros.

4.3. Pre-processing Phase

1. **Checking for missing values:** We aim to identify any missing data within the dataset. This can be accomplished by utilizing the ‘dataframe.isnull ()’ function in the Pandas library. This function will yield ’True’ for any absent elements and ’False’ for those that are not missing. Furthermore, the expression ‘dataframe.isnull ().sum ()’ enables us to determine the total count of missing values within the dataset.

2. **Filling Missing Values:** We have numerous alternatives for dealing with values that are absent when working with quantitative variables, among which are substituting them with the sample mean, mode, median, or other numerical values. Conversely, when working with categorical variables, we can address missing values by introducing a new category and replacing the missing values with a designated string. Alternatively, one may choose to replace missing values with zeros in certain cases.

4.4. Machine Learning classifiers for Social Media Fake profiles detection

To identify social media fake profiles, we integrated multiple supervised methods, each sharing the common objective of distinguishing fake profiles, albeit with varying levels of accuracy. Each model relies solely on observable characteristics for fake profile detection. The identical dataset is supplied to all of the aforementioned trained models, and the models used are mentioned below.

(a) Decision Tree (DT). In ML, a DT is a tree-like structure that is used for both classification and regression problems. It entails recurrent binary splits depending on the input attribute values. A decision tree can be mathematically represented as [36]:

1. A DT T is a binary tree structure where each node n represents a feature F_n and a threshold value T_n as a result of which the data is divided into two parts, D_{left} and D_{right} based on a decision rule.

2. At each internal node n, the decision rule is represented as $F_n \leq T_n$ and the data is split into two subsets:
   D_{left} where $F_n \leq T_n$
   D_{right} where $F_n > T_n$

3. Leaf nodes contain a prediction value for regression tasks or class probabilities for classification tasks.

Impurity Measures

- **Gini Impurity:** The Gini impurity $Gini(D)$ for a dataset D with class C is calculated as:

$$Gini(D) = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{1}$$

Where:

  – C is the number of available classes.

  – p_i is the proportion of samples in class i at the node.

- **Entropy:** Entropy $Entropy(D)$ measures impurity based on information content and is calculated as:

  $$Entropy(D) = -\sum_{i=1}^{C} p_i \log_2(p_i) \tag{2}$$

  Where:

  - C is the number of available classes.
  - p_i is the proportion of samples in class i at the node.

- **Splitting Criterion:** For classification, the choice of the feature and threshold at each node is based on minimizing Gini impurity or entropy, leading to the highest information gain. In a decision tree algorithm, these measures are used to determine the optimal splits in the data, recursively creating the tree structure until certain stopping conditions are satisfied.

(b) Random Forest (RF). RF is an ensemble learning method that combines multiple decision trees to improve predictive accuracy and reduce overfitting. Here is a mathematical representation of RF: A RF is composed of multiple decision trees, typically denoted as, $T_1, T_2, \ldots, T_n$ where n is the number of trees in the forest.

Each decision tree T_i is constructed using a random subset of the data and a random subset of the features. This is done to introduce diversity in the individual trees, which collectively results in a more robust and accurate model [37].

- **Bootstrapping:** Bootstrapping is utilized to make multiple subsets of the original dataset for training each tree. Each subset is generated by randomly selecting samples with replacement. The size of each subset can be the same as the original dataset or smaller. Mathematically, bootstrapping can be represented as:

  Given a dataset D with N samples, D_i is a bootstrapped dataset with N samples, where N_i amples are randomly selected from D with replacement.

- **Feature Subset Selection:** For each DT T_i, a random subset of attributes is taken into account when splitting nodes. This subset is typically of size m, where$m$ is less than the total number of features M. Mathematically, feature subset selection can be represented as: Given a dataset with M features, m features are at each node split, a decision at random is made for consideration.

- **Random Forest Predictions:** The predictions from the Random Forest are typically aggregated as of the following:

  - **Classification (Voting):** For classification tasks, each tree in the forest casts a vote for the class, and the class with the most votes is the final prediction. Mathematically, it can be represented as:

    $$Class_{\text{Final}} = \text{mode}(Class_{T_1}, Class_{T_2}, \ldots, Class_{T_n}) \tag{3}$$

    Where: Where:

    - $Class_{T_i}$ indicates the class predicted by the i-th decision tree.

(c) Logistic Regression (LR). LR is a statistical model used for binary classification. Here's the mathematical representation of logistic regression [38]:

- **Hypothesis Function:** The logistic regression model uses the logistic function (also called the sigmoid function) to model the probability that a given input belongs to a specific class. The hypothesis function $h_\theta(x)$ for logistic regression is defined as:

  $$h_\theta(x) = \frac{1}{1 + e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2+\ldots+\theta_n x_n)}} \tag{4}$$

  Where:

  - $h_\theta(x)$ is the predicted probability that input x belongs to the positive class (class 1).
  - $\theta_0, \theta_1, \theta_2, \ldots, \theta_n$ are the model parameters.
  - $x_1, x_2, \ldots, x_n$ are the input features.
  - e is the base of the natural logarithm.

- **Linear Model:** The linear model within the sigmoid function is represented as:

  $$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \tag{5}$$

  Where, z is the linear combination of the model parameters and input features.

- **Cost Function:** To train the logistic regression model, a cost method is employed to calculate the error between the predicted values and the actual class labels. The cost function is typically the logistic loss (also called log loss or cross-entropy loss) for binary classification:

  $$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))\right] \tag{6}$$

  Where,

  - $J(\theta)$ is the cost function.
  - m is the quantity of training instances.

- $x^{(i)}$ is the feature vector of the i-th training example.
- $y^{(i)}$ is the actual class label of the i-th training example (0 for the negative class, 1 for the positive class).
- $h_\theta(x^{(i)})$ is the predicted probability that $x^{(i)}$ belongs to the positive class.

(d) XGBoost. Extreme Gradient Boosting is a powerful machine learning algorithm that excels in both classification and regression tasks [39].

- **Objective Function:** XGBoost employs a specific objective task that must be performed optimized during the training process. In the case of binary classification, a common objective function is the logistic loss:

$$L(\theta) = \sum_{i=1}^{n} \left[y_i \cdot \log(1 + e^{-\hat{y}_i}) + (1 - y_i) \cdot \log(1 + e^{\hat{y}_i}) \right] + \Omega(f) \tag{7}$$

  Where,

  - $L(\theta)$ is the objective function to be minimized.
  - n is the number of data points in the training set.
  - y_i is the true label of the i-th data point (0 or 1 for binary classification).
  - $\hat{y}_i$ is the prediction for the i-th data point.
  - $\Omega(f)$ represents an attribute of regularization that regulates the model's level of complexity to prevent overfitting.

- **Prediction for Each Tree:** In XGBoost, the prediction for each tree (boosting iteration) is defined as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot f_t(x_i) \tag{8}$$

  Where,

  - $\hat{y}_i^{(t)}$ is the prediction for data point after the t-th tree.
  - $\hat{y}_i^{(t-1)}$ is the prediction for data point i after the $(t-1)$ -th tree.
  - η is the training rate, that determines the size of the step used to update the predictions.
  - $f_t(x_i)$ is the contribution of the t-th tree to the prediction for data point i.

- **Tree Structure:** XGBoost builds trees to fit the loss caused by the function's negative gradient. The structure of each tree ($f_t(x)$) is learned using techniques like gradient boosting with tree-based learners. The mathematical representation of the tree structure is intricate and involves various components, including the split points and leaf values. The entire XGBoost model combines the predictions from all trees in the ensemble to provide the final prediction for a given data point. In practice, XGBoost uses gradient boosting techniques and employs regularization terms to optimize the objective function. The mathematical details of the algorithm's implementation can be complex, but the above equations represent the core principles underlying XGBoost.

(e) Extra Trees (ET). Extra Trees, also known as Extremely Randomized Trees or Extra-Trees, is an approach to collaborative learning that is similar to Random Forest but has several significant distinctions. Extra Trees are mathematically represented as follows [40]:

- **Ensemble Structure:** An Extra Trees ensemble consists of a set of decision trees, denoted as $T_1, T_2, \ldots, T_n$ where n is the number of trees in the ensemble. Each decision tree T_i s constructed based on a random subset of the data and a random subset of features, similar to Random Forest. This is done to introduce diversity in the individual trees and collectively result in a more robust and accurate model.

- **Bootstrapping:** Bootstrapping is used to create multiple subsets of the original dataset for training each tree. Each subset is generated by randomly selecting samples with replacement. The size of each subset can be the same as the original dataset or smaller. Mathematically, bootstrapping can be represented as: Given a dataset D with N samples, D_i is a bootstrapped dataset with N samples, where N_i samples are randomly selected from D with replacement.

- **Feature Subset Selection:** For each DT T_i, a random subset of features is considered when splitting nodes, similar to RF. This subset is typically of size m, where m is less than the total number of features M. Mathematically, feature subset selection can be represented as: Given a dataset with M features, m features are at each node split, an arbitrary number is chosen for consideration.

- **Node Splitting:** In Extra Trees, nodes are split in a different way compared to Random Forest. While Random Forest typically chooses the best split, Extra Trees chooses splits randomly. For each node, Extra Trees randomly selects a feature and a threshold for the split, rather than choosing the optimal feature and threshold. This “extra" randomness contributes to the “extremely randomized" nature of Extra Trees.

- **Aggregation:** The final prediction in Extra Trees is typically aggregated in one of the following ways, similar to Random Forest:
  - **Classification (Voting):** For classification tasks, each tree in the ensemble casts a vote for the class, and the class with the most votes is the final prediction.
  - **Regression (Averaging):** For regression tasks, the final prediction is often the average of the predictions from all trees in the ensemble.

  The “extra" randomness in feature selection and node splitting differentiates Extra Trees from other ensemble methods like Random Forest. This randomness helps reduce overfitting and can lead to more robust models.

(f) K-Nearest Neighbors (KNN)). KNN is a simple and effective machine learning algorithm used for both classification and regression tasks. Here’s a mathematical representation of the KNN algorithm: KNN is a non-parametric, instance-based algorithm that makes predictions based on the majority class or the average of the k-nearest data points in the feature space. Given a dataset D with n data points, where each data point x_i is associated with a label y_i KNN works as follows [41]:

- **Step 1: Define the distance metric:** Choose a distance metric, typically the Euclidean distance is used to calculate the distance between data points. The Euclidean distance between two data points x_i and x_j in a feature space with m features is given by:

  $$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2} \tag{9}$$

  Where:

  - x_{ik} and x_{jk} are the values of feature k for data points x_i and x_j.

- **Step 2: Select the number of neighbors (k):** Select the number of nearest neighbors (k) to consider when making predictions. Typically, k is a positive integer.

- **Step 3: Predict a new data point:** To forecast a new data point’s class label x, find the k data points in D that are closest to x based on the chosen distance metric. Count the number of data points in each class among these k neighbors, and assign the class label that appears most frequently among the k neighbors to x.

- **Mathematical Representation for Classification:** In the case of KNN classification, the prediction for a new data point x can be represented mathematically as:

  $$y(x) = \arg\max_{y_i} \sum_{i=1}^{k} \mathbb{1}(y_i = y) \tag{10}$$

  Where:

  - $y(x)$ is the predicted class label for data point x.
  - y_i represents the class labels of the nearest neighbors of x.
  - $1(y_i = y)$ is a function that returns a notification 1 if $y_i = y$ and 0 otherwise.

  KNN is a straightforward algorithm to understand and implement, and it’s particularly useful when dealing with small to medium-sized datasets. However, it can be computationally expensive for large datasets, and the choice of the distance metric and k value can significantly impact its performance.

(g) AdaBoost. AdaBoost is a form of collaborative learning that combines several weak learners to produce a strong learner. Below is a mathematical representation of the AdaBoost algorithm. AdaBoost works by iteratively training a series of weak classifiers and assigning them weights based on their performance. The final prediction is a weighted sum of the weak classifier predictions. Given a training dataset (X, y) where X represents the feature vectors and y represents the labels (typically -1 and 1 for binary classification), the AdaBoost algorithm can be mathematically represented as follows [42]:

Algorithm:

- **Step 1: Initialize Weights:** Initialize equal weights for all training samples, $w_i = \frac{1}{N}$, where N is the number of training samples.

- **Step 2: For $t = 1$ to T**

  - **a.** Train a weak classifier $h_t(x)$ using the training data weighted by w_i.

- **b.** Compute the error ϵ_t of the classifier on the weighted training data:

$$\epsilon_t = \sum_{i=1}^{N} w_i \cdot 1(h_t(x_i) \neq y_i) \tag{11}$$

where 1 is a function that returns a notification 1 when $h_t(x_i) \neq y_i$ (the classifier made a mistake) and 0 otherwise.

- **c.** Calculate the classifier weight α_t as:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{12}$$

- **d.** Update the weights for the training samples:

$$w_i \leftarrow w_i \cdot \exp\left(-\alpha_t \cdot y_i \cdot h_t(x_i)\right) \tag{13}$$

- **e.** Normalize the weights to ensure they sum to 1:

$$w_i \leftarrow \frac{w_i}{\sum_{j=1}^{N} w_j} \tag{14}$$

- **Step 3: Final Prediction:** The final prediction for a new data point x is computed as the weighted sum of the weak classifier predictions:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x) \right) \tag{15}$$

Where:

- $H(x)$ is the final prediction for data point x.
- $h_t(x)$ represents the prediction of the t-th weak classifier.
- α_t is the weight assigned to the t-th weak classifier.

AdaBoost combines the predictions of multiple weak classifiers, giving more weight to those that perform well. The final prediction is based on the weighted votes of the individual classifiers. The main idea behind AdaBoost is to focus on examples that are challenging and reweight the data during training to emphasize the misclassified samples, ultimately improving the overall performance of the ensemble.

(h) Majority Voting Approach. Majority voting is a straightforward ensemble approach that is frequently applied to classification challenges. It entails merging the predictions of numerous independent classifiers in order to create a final prediction based on a majority vote. The mathematical representation of majority voting is straightforward: Given a set of N classifiers $\{C_1, C_2, \ldots, C_N\}$, where each classifier C_i produces a binary prediction $y_i \in \{0, 1\}$ or $\{-1, 1\}$ for a given input sample, the majority voting ensemble works as follows [43]:

- **Step 1:** Each classifier C_i provides a binary prediction, where 1 or “positive" typically indicates the positive class, and 0 or −1 indicates the negative class.
- **Step 2:** For each input sample, the majority vote is calculated by counting the number of learners that accurately forecast positive class (1 or “positive") and the number of learners that predict the negative class (0 or −1).
- **Step 3:** The final prediction for the ensemble is based on the majority class determined in the vote. If a majority of the learners correctly forecast the positive class, then, the ensemble predicts the positive class; otherwise, it predicts the negative class.

Mathematically, the majority voting ensemble can be represented as follows: For a given input sample x, let $y_1, y_2, \ldots, y_N$ represent the binary predictions from the N individual classifiers, where $y_i \in \{0, 1\}$ or $\{-1, 1\}$.

In Majority Voting, the ensemble’s final decision is based on the majority opinion of the individual classifiers. If more than half of the classifiers vote for the positive class (1), the ensemble predicts the positive class; otherwise, it predicts the negative class (0).

5. Experimental Results and Discussion

This section shows the results of our research on the identification of fraudulent social media profiles and provides a thorough discussion of the findings. Our study was meant to evaluate the usefulness of various characteristics and machine learning models in recognizing fake accounts on social media sites.

5.1. Dataset

We used a MIB dataset that included 3,474 authentic accounts and 3,351 fraudulent profiles. For legitimate accounts, we specifically chose the following information: TFP and E13 were used for real accounts, whereas INT, TWT, and FSF were used for false accounts. The information was saved in CSV format to make it easier for machines to read. The x-axis labels reflect the characteristics selected throughout the phase of preprocessing for use in the false profile detection procedure. The y-axis, on the other hand, represents the number of entries linked with each characteristic in the dataset [34].

5.2. Model Evaluation

We explored the performance of several machine learning models, including Decision Trees, XGBoost, Random Forest, Extra Trees, Logistic Regression, AdaBoost, K-Nearest Neighbors and Majority Voting classifier for the task of fake profile detection. We have used following measures to evaluate our proposed approach for fake social media profiles detection [44].

- **Accuracy:** Accuracy evaluates the overall efficiency of a categorization model. It is the proportion of accurately predicted occurrences to the total number of examples in the dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

- **Precision:** Precision assesses the accuracy of the positive predictions made by the model. It is the ratio of true positive predictions to all positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

- **Recall:** The model's ability to properly identify positive cases is measured by recall. It is the proportion of genuine positive predictions to all occurrences of true positive prediction.

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

- **F1-Measure (F1-Score):** The F1-measure is a assessment of the model's performance since it is an average of accuracy and recall.

$$F1\text{-}Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{19}$$

The confusion matrix yielded the values TP, TN, FP, and FN. These performance indicators, which offer a quantifiable assessment of accuracy, precision, recall, and overall performance, are frequently used to assess the success of classification algorithms. The performance assessment of machine learning classifiers on the MIB fake social media profile dataset is shown in Table 1 and Figure 3.

Table 1 presents the performance evaluation of various machine learning classifiers applied to the MIB fake social media profile dataset, assessing accuracy, precision, recall, and F1-measure. The Decision Tree Classifier demonstrated exceptional results with an accuracy, precision, recall, and F1-measure of 98.49%. Meanwhile, the Random Forest classifier achieved equally outstanding scores across all metrics, with an accuracy, precision, recall, and F1-measure of 99.12%. Similarly, the Extra Trees classifier delivered strong performance, boasting an accuracy, precision, recall, and F1-measure of 99.07%. The XGBoost classifier exhibited remarkable consistency, registering an accuracy, precision, recall, and F1-measure of 98.83%.

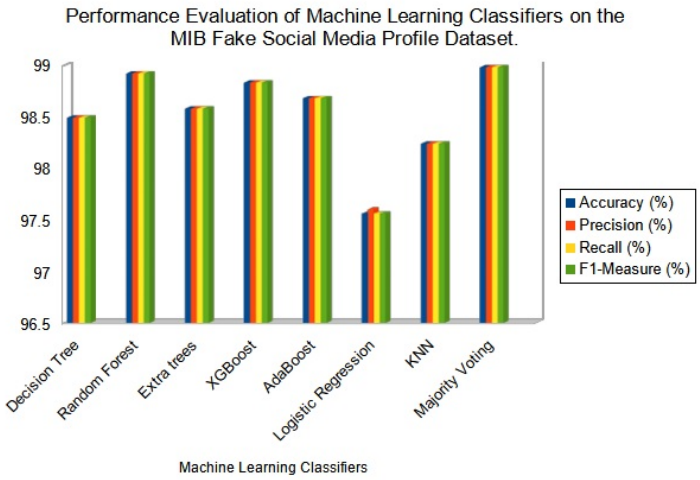

Figure 3. Performance Assessment of Machine Learning Classifiers on the MIB Fake Social Media Profile Dataset.

Furthermore, the AdaBoost classifier excelled in all categories, garnering an accuracy, precision, recall, and F1-measure of 99.22%. On the other hand, the Logistic Regression learning classifier posted respectable results, recording an accuracy of 97.31%, precision of 97.37%, recall of 97.31%, and an F1-measure of 97.31%. The K-Nearest Neighbors (KNN) classifier demonstrated a noteworthy performance, securing an accuracy, precision, recall, and F1-measure of 98.63%. Notably, the Majority Voting classifier demonstrated superior results, with an accuracy, precision, recall, and F1-measure of 99.12%.

Our approach uniquely leverages the Majority Voting technique to amalgamate predictions from multiple classifiers, establishing itself as a robust and dependable strategy for addressing this challenging issue. The ensemble method significantly outperforms individual classifiers, yielding a notably impressive detection performance.

Following Figure 4 to 11 shows the performance analysis of individual classifier in terms of confusion matrix and ROC curve.

Table 1. Performance Assessment of Machine Learning Classifiers on the MIB Fake Social Media Profile Dataset

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Decision Tree	98.49	98.49	98.49	98.49
Random Forest	99.12	99.12	99.12	99.12
Extra Trees	99.07	99.07	99.07	99.07
XGBoost	98.83	98.83	98.83	98.83
AdaBoost	99.22	99.22	99.22	99.22
Logistic Regression	97.31	97.37	97.31	97.31
K-Nearest Neighbors	98.63	98.63	98.63	98.63
Proposed Approach using Majority Voting	99.12	99.12	99.12	99.12

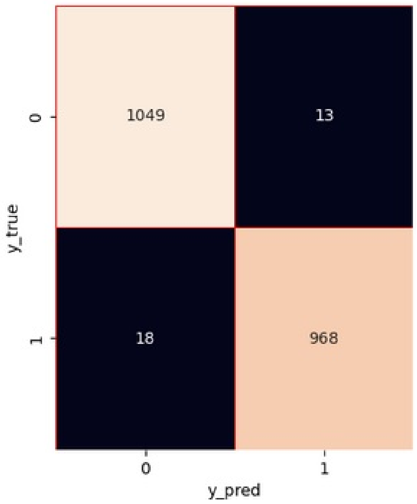

Figure 4. Confusion Matrix for Decision Tree Classifier.

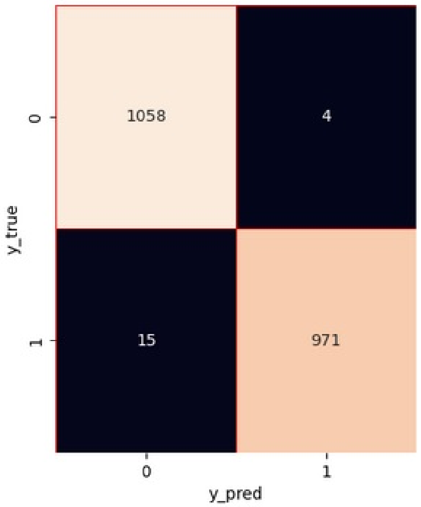

Figure 6. Confusion Matrix for Extra Tree Classifier.

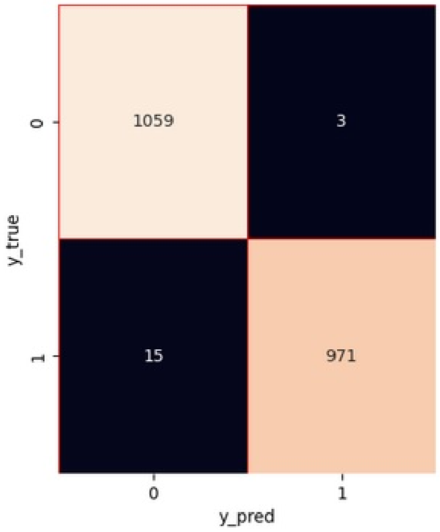

Figure 5. Confusion Matrix for Random Forest Classifier.

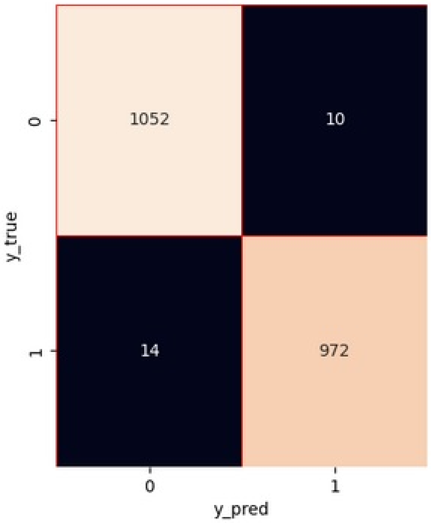

Figure 7. Confusion Matrix for XGBoost Classifier.

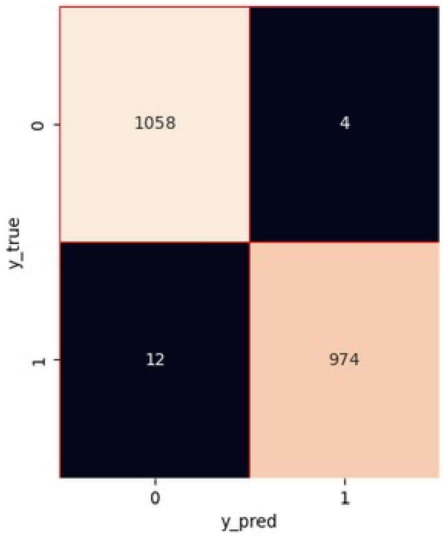

Figure 8. Confusion Matrix for AdaBoost Classifier.

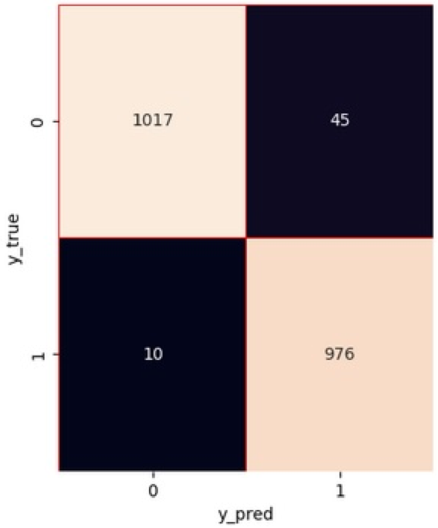

Figure 9. Confusion Matrix for Logistic Regression Classifier.

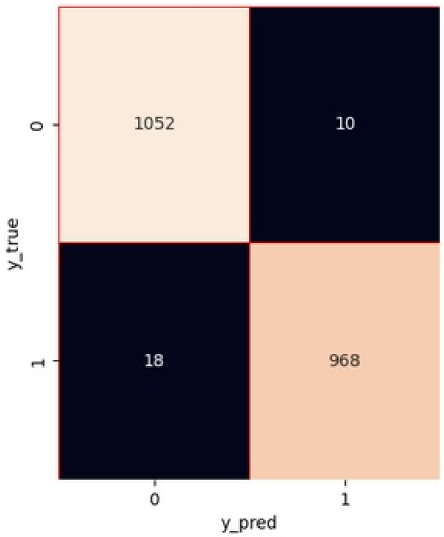

Figure 10. Confusion Matrix for K-Nearest Neighbors Classifier.

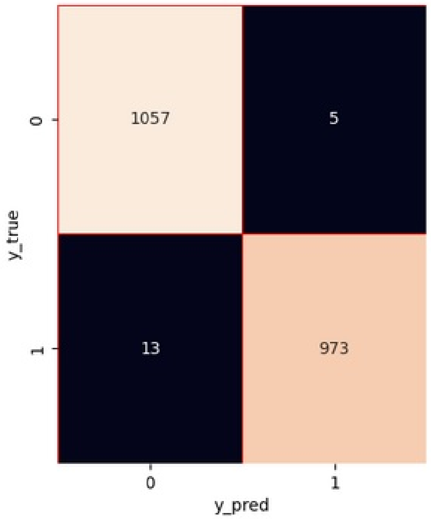

Figure 11. Confusion Matrix for Majority Voting Classifier.

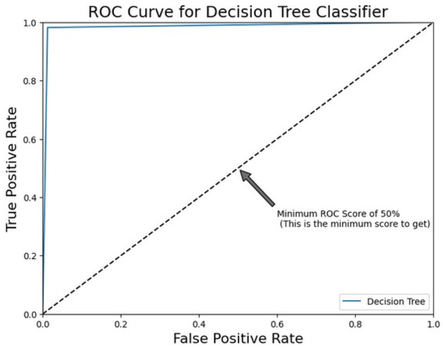

Figure 12. ROC Curve of Decision Tree Classifier.

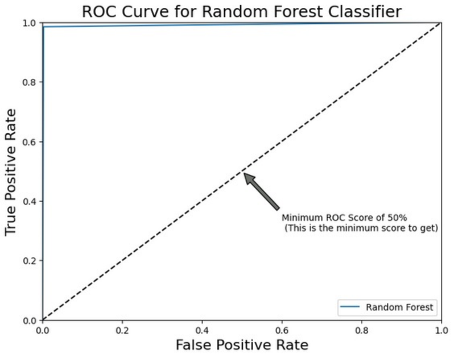

Figure 13. ROC Curve of Random Forest Classifier.

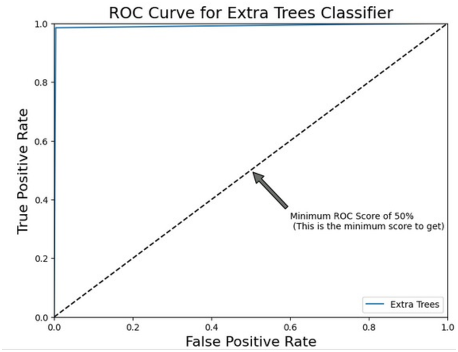

Figure 14. ROC Curve of Extra Tree Classifier.

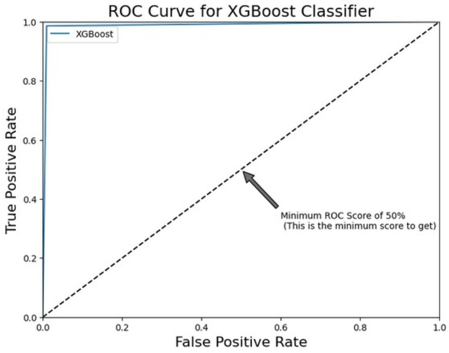

Figure 15. ROC Curve of XGBoost Classifier.

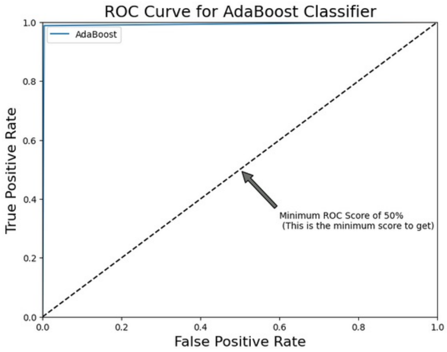

Figure 16. ROC Curve of AdaBoost Classifier.

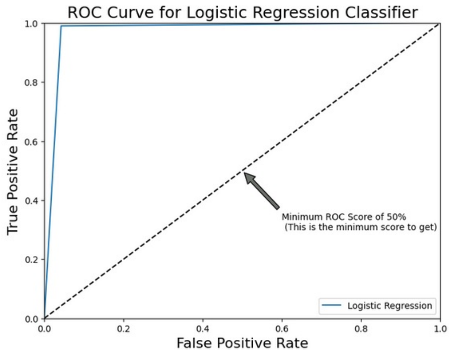

Figure 17. ROC Curve of Logistic Regression Classifier.

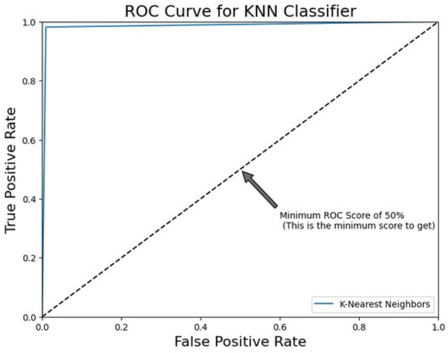

Figure 18. ROC Curve of K-Nearest Neighbors Classifier.

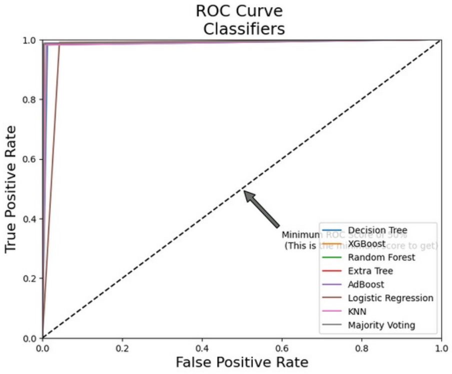

Figure 19. ROC Curve of All Classifier.

5.3. Discussion

This paper has focused on the detection of fake social media profiles, a critical task in the realm of online security and social media management. Specifically, our approach leverages the Majority Voting technique to aggregate predictions from multiple classifiers, which has proven to be a robust and reliable method for tackling this challenging problem. The ensemble approach significantly outperforms individual classifiers, achieving a remarkable detection accuracy of 99.12%, a precision of 99.12%, a recall of 99.12% and an F1-score of 99.12%. This high level of accuracy indicates the potential of our technique to play a vital role in identifying deceptive profiles and maintaining the trustworthiness of social networking platforms.

6. Conclusion

In the age of online communication and networking, detecting false online social networking identities is critical. In this paper, we have explored the effectiveness of the Majority Voting technique as a powerful tool for identifying fake social media profiles. The results and insights gained from our work hold significant implications for enhancing the trustworthiness of social media platforms and online communities.

The findings of our study have shown that the Majority Voting approach is effective in detecting counterfeits profiles. With an accuracy of 99.12%, a precision rate of 99.12%, a recall rate of 99.12% and an F1-score of 99.12%, our approach significantly outperforms individual classifiers. This high level of accuracy instills confidence in the reliability of our method to identify and mitigate the risks associated with fraudulent profiles.

While the Majority Voting technique exhibits promise, we acknowledge that the battle against fake social media profiles is an ongoing one. The dynamic nature of online platforms and the continuous evolution of deceptive tactics require constant vigilance and research. Addressing emerging challenges, such as adversarial attacks and the identification of deepfake profiles, represents an essential area for future exploration.

References

[1] Facebook: fake account removal as of Q4 2022, Published by Stacy Jo Dixon, March 9, 2023. Available online, https://www.statista.com/statistics/1013474/facebook-fake-account-removal-quarter.

[2] Twitter Inc. (2013-2020). Twitter annual reports. Twitter Inc. Available online, https://www.annualreports.com/Company/twitter-inc.

[3] Snap Inc. Snap annual report 2020. Available online, https://s25.q4cdn.com/442043304/files/doc_presentations/presentation/2021/Snap-Inc.-2020-Annual-Report.pdf.

[4] LinkedIn. Annual report 2015. Available online, https://news.linkedin.com/2016/linkedin-announces-fourth-quarter-and-full-year-2015-results.

[5] Ramalingam D, Chinnaiah V. Fake profile detection techniques in large-scale online social networks: A comprehensive review. Computers & Electrical Engineering. 2018 Jan 1;65:165-77.

[6] Ezarfelix J, Jeffrey N, Sari N. A Systematic Literature Review: Instagram Fake Account Detection Based on Machine Learning. Engineering, MAthematics and Computer Science (EMACS) Journal. 2022 Feb 5;4(1):25-31.

[7] Kaubiyal J, Jain AK. A feature based approach to detect fake profiles in Twitter. InProceedings of the 3rd international conference on big data and internet of things 2019 Aug 22 (pp. 135-139).

[8] Mohammadrezaei M, Shiri ME, Rahmani AM. Identifying fake accounts on social networks based on graph analysis and classification algorithms. Security and Communication Networks. 2018 Aug 5;2018.

[9] Joshi UD, Vanshika, Singh AP, Pahuja TR, Naval S, Singal G. Fake social media profile detection. Machine Learning Algorithms and Applications. 2021 Aug 19:193-209.

[10] Ajesh F, Aswathy SU, Philip FM, Jeyakrishnan V. A hybrid method for fake profile detection in social networkusing artificial intelligence. Security Issues and Privacy Concerns in Industry 4.0 Applications. 2021 Jun 15:89-112.

[11] Khaled S, El-Tazi N, Mokhtar HM. Detecting fake accounts on social media. In2018 IEEE international conference on big data (big data) 2018 Dec 10 (pp. 3672-3681). IEEE.

[12] Xiao C, Freeman DM, Hwa T. Detecting clusters of fake accounts in online social networks. In Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security 2015 Oct 16 (pp. 91-101).

[13] Kodati S, Reddy KP, Mekala S, Murthy PS, Reddy PC. Detection of Fake Profiles on Twitter Using Hybrid SVM Algorithm. In E3S Web of Conferences 2021 (Vol. 309, p. 01046). EDP Sciences.

[14] Sahoo SR, Gupta BB. Fake profile detection in multimedia big data on online social networks. International Journal of Information and Computer Security. 2020;12(2-3):303-31.

[15] Akyon FC, Kalfaoglu ME. Instagram fake and automated account detection. In2019 Innovations in intelligent systems and applications conference (ASYU) 2019 Oct 31 (pp. 1-7). IEEE.

[16] Bordbar J, Mohammadrezaie M, Ardalan S, Shiri ME. Detecting fake accounts through Generative Adversarial Network in online social media. arXiv preprint arXiv:2210.15657. 2022 Oct 25.

[17] Samreen S, Joshna A, Neelima B, Varsha C, Nikitha G. ARTIFICIAL NEURAL NETWORKS FOR FAKE ACCOUNT DETECTION FROM SOCIAL MEDIA. Turkish Journal of Computer and Mathematics Education (TURCOMAT). 2023 Jul 6;14(03):53-64.

[18] Ali AK, Abdullah AM. Fake accounts detection on social media using stack ensemble system. International Journal of Electrical and Computer Engineering (IJECE). 2022 Jun 1;12(3):3013-22.

[19] Muñoz SD, Pinto EP. A dataset for the detection of fake profiles on social networking services. In2020 International Conference on Computational Science and Computational Intelligence (CSCI) 2020 Dec 16 (pp. 230-237). IEEE.

[20] Rostami RR, Karbasi S. Detecting Fake Accounts on Twitter Social Network Using Multi-Objective Hybrid Feature Selection Approach. Webology. 2020 Jun 1;17(1).

[21] Kadam N, Sharma SK. Social media fake profile detection using data mining technique. Journal of Advances in Information Technology Vol. 2022 Oct;13(5).

[22] Al-Qurishi M, Rahman SM, Hossain MS, Almogren A, Alrubaian M, Alamri A, Al-Rakhami M, Gupta BB. An efficient key agreement protocol for Sybil-precaution in online social networks. Future Generation Computer Systems. 2018 Jul 1;84:139-48.

[23] Conti M, Poovendran R, Secchiero M. Fakebook: Detecting fake profiles in on-line social networks. In2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2012 Aug 26 (pp. 1071-1078). IEEE.

[24] Gurajala S, White JS, Hudson B, Matthews JN. Fake Twitter accounts: profile characteristics obtained using an activity-based pattern detection approach. InProceedings of the 2015 international conference on social media & society 2015 Jul 27 (pp. 1-7).

[25] Gurajala S, White JS, Hudson B, Voter BR, Matthews JN. Profile characteristics of fake Twitter accounts. Big Data & Society. 2016 Oct;3(2):2053951716674236.

[26] Meligy AM, Ibrahim HM, Torky MF. Identity verification mechanism for detecting fake profiles in online social networks. Int. J. Comput. Netw. Inf. Secur.(IJCNIS). 2017 Jan 1;9(1):31-9.

[27] Jin L, Takabi H, Joshi JB. Towards active detection of identity clone attacks on online social networks. In Proceedings of the first ACM conference on Data and application security and privacy 2011 Feb 21 (pp. 27-38).

[28] Nagariya HG, Dhanotiya N, Joshi S, Jain S. Identifying Fake Profile in Online Social Network. In ACM Workshop on Advances in Computational Intelligence at ISIC 2021.

[29] Uppada SK, Manasa K, Vidhathri B, Harini R, Sivaselvan B. Novel approaches to fake news and fake account detection in OSNs: user social engagement and visual content centric model. Social Network Analysis and Mining. 2022 Dec;12(1):52.

[30] Du, Jiahua, Jia Rong, Hua Wang, and Yanchun Zhang. Neighbor-aware review helpfulness prediction. Decision Support Systems 148 (2021): 113581.

[31] Du, Jiahua, Liping Zheng, Jiantao He, Jia Rong, Hua Wang, and Yanchun Zhang. An interactive network for end-to-end review helpfulness modeling. Data Science and Engineering 5 (2020): 261-279.

[32] Zhou, Yuzhong, Zhengping Lin, Liang Tu, Junkai Huang, and Zifeng Zhang. Analysis and design of standard knowledge service system based on deep learning. EAI Endorsed Transactions on Scalable Information Systems 10, no. 2 (2022).

[33] Reddy, Shiva Shankar, Mahesh Gadiraju, N. Meghana Preethi, and VVR Maheswara Rao. A Novel Approach for Prediction of Gestational Diabetes based on Clinical Signs and Risk Factors. EAI Endorsed Transactions on Scalable Information Systems 10, no. 3 (2023).

[34] Cresci S, Di Pietro R, Petrocchi M, Spognardi A, Tesconi M. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In Proceedings of the 26th international conference on world wide web companion 2017 Apr 3 (pp. 963-972).

[35] Elyusufi Y, Elyusufi Z, Kbir MH. Social networks fake profiles detection using machine learning algorithms. In Innovations in Smart Cities Applications Edition 3: The Proceedings of the 4th International Conference on Smart City Applications 4 2020 (pp. 30-40). Springer International Publishing.

[36] Maimon O, Rokach L, editors. Data mining and knowledge discovery handbook. New York: Springer; 2005 Sep 1.

[37] Breiman L. Random forests. Machine learning. 2001 Oct;45:5-32.

[38] Stoltzfus JC. Logistic regression: a brief primer. Academic emergency medicine. 2011 Oct;18(10):1099-104.

[39] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and

data mining 2016 Aug 13 (pp. 785-794).

[40] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Machine learning. 2006 Apr;63:3-42.

[41] Peterson LE. K-nearest neighbor. Scholarpedia. 2009 Feb 21;4(2):1883.

[42] Schapire RE. Explaining adaboost. In Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik 2013 Oct 9 (pp. 37-52). Berlin, Heidelberg: Springer Berlin Heidelberg.

[43] Ruta D, Gabrys B. Classifier selection for majority voting. Information fusion. 2005 Mar 1;6(1):63-81.

[44] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Information processing & management. 2009 Jul 1;45(4):427-37.