FAKE PROFILE IDENTIFICATION IN SOCIAL NETWORK

A PROJECT REPORT

Submitted by

HARSHAVARDHINI.A.C [REGISTER NO:211420104095]

KANMANISHREE.M [REGISTER NO:211420104123]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution Affiliated to Anna University, Chennai)

March 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"FAKE PROFILE IDENTIFICATION IN SOCIAL NETWORK"** is the bonafide work of "**HARSHAVARDHINI.A.C (211420104095), KANMANISHREE.M(211420104123),**" who carried out the Project work under my supervision.

Signature of the HOD with date

Dr.L.JABASHEELA, M.E., Ph.D.,
PROFESSOR
HEAD OF THE DEPARTMENT
Department of Computer Science and Engineering,

Panimalar Engineering College,

Chennai-123.

Signature of the Supervisor with date

Dr.KAVITHASUBRAMANI,M.E.,PH.D.,
SUPERVISOR
PROFESSOR
Department of Computer Science and Engineering,

Panimalar Engineering College,

Chennai-123.

Certified that the above candidate(s)was examined in the End Semester Project Viva-Voice Examination held on ______________________________________

INTERNALEXAMINER **EXTERNALEXAMINER**

DECLARATION BY THE STUDENT

We HARSHAVARDHINI.A.C(211420104095),KANMANISHREE.M (211420104123) hereby declare that this project report titled "**FAKE PROFILE IDENTIFICATION IN SOCIAL NETWORK**", under the guidance of Dr. KAVITHA SUBRAMANI, M.E., PH.D., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

HARSHAVARDHINI.A.C

KANMANISHREE.M

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr.P.CHINNADURAI,M.A.,Ph.D.,** for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project.

We express our sincere and hearty thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us With the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We express my heart felt thanks to **Dr.L.JABASHEELA,M.E.,Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Dr.KAVITHA SUBRAMANI,M.E.,PH.D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

HARSHAVARDHINI.A.C

KANMANISHREE.M

ABSTRACT

This work presents a novel method for identifying fake profiles in social net works through a combination of machine learning algorithms and user behavior analysis. By leveraging a diverse range of features including profile information, posting patterns, network characteristics, and engagement levels, the propose approach aim to accurately differentiate between genuine and fake accounts. Through the application of advanced data mining techniques and anomaly detection algorithms, the system effectively identifies suspicious pattern and in consistencies characteristic of fake profiles. Further more, the model is designed to continuously adapt and improve its accuracy by incorporating real-time data and user feedback. Experimental results demonstrate the effectiveness of the proposed approach, achieving a high accuracy rate in detecting fake profiles with precision and recall rates exceeding 90%. Thus, the method provides a valuable tool for social network administrators and users to combat fraudulent activities and maintain the integrity of online communities.

TABLE OF CONTENTS

LIST OF FIGURES

LISTOFTABLES

CHAPTER 1
INTRODUCTION

1.1 OVER VIEW

Fake profiles are a pervasive issue on social networking platforms, posing various risks and challenges for users. Identification of fake profiles is crucial in maintaining the integrity and security of online communities. These deceptive accounts are typically created with false or misleading information, often with the intention of spreading misinformation, engaging in fraudulent activities, or harvesting personal data. Common characteristics of fake profiles include unrealistic profile pictures, minimal personal information, inconsistent details, and a high volume of friend requests or connections. To detect fake profiles, social network user and platforms can utilize various techniques such as reverse image search, analyzing posting patterns, checking account creation dates, and monitoring for suspicious behavior like excessive messaging or spamming. By actively identifying and reporting fake profiles, users can help minimize the spread of misinformation and protect themselves and others from potential online threats. It is essential for social networks to continuously enhance their security measures and algorithms to combat the prolife ration of fake profile and ensure safe and trust worthy online environment for all users. Identifying fake profiles on social networks involves a multifaceted approach that encompasses both automated and human judgment. Automated methods often include analyzing patterns in account creation, such as examining activity levels, content sharing behavior, and engagement patterns. Machine learning algorithms can be trained to detect anomalies in these patterns, flagging accounts for further review. Additionally, as sensing profile completeness, consistency in personal information, and the quality of connections can provide valuable in sights. Human intervention remains crucial for nuanced judgment, especially in cases where automated systems may falter, such as identifying sophisticated bots or accounts using stolen identities. Collaboration between technology and human methods for combating fake profiles and maintaining .

1.2 PROBLEM DEFINITION

Identifying fake profiles in social networks presents a significant challenge due to the evolving tactics employed by malicious actors. These fake profiles can be used for various malicious purposes, including spreading is information, conducting fraudulent activities, or manipulating public opinion. Traditional methods of detection, such as analyzing account activity and engagement patterns, often fall short in identifying sophisticated fake profiles that mimic genuine user behavior. Moreover, the sheer volume of users on popular social platforms makes manual identification impractical. As a result, there is a pressing need for innovative approaches that leverage advanced technologies like machine learning and natural language processing to detect sub indicative of fake profiles. However, develop in effective solutions require over coming technical hurdles and balancing the need for accuracy with user privacy concerns. Addressing this problem is crucial for maintaining the integrity and trustworthiness of social networks as vital channels for communication and information dissemination.

1.3 FEASIBILITY STUDY

The objective of feasibility study is not only to solve the problem but also to acquire a sense of its scope. During the study, the problem definition was crystallized and aspects of the problem to be included in the system are determined. Consequently, benefits are estimated with greater accuracy at this stage. The key considerations are:

- Economic feasibility
- Technical feasibility

Economic Feasibility

Economic feasibility studies not only the cost of hardware, software are included but also the benefits in the form of reduced costs are considered here. This project, if installed will certainly be beneficial since there will be reduction in manual work and increase in the speed of work.

Technical Feasibility

Technical feasibility evaluates the hardware requirements, software technology, available personnel etc., as per the requirements it provides sufficient memory to hold and process.

1) Deep Learning Architecture
2) Google Drive
3) IDE: Jupyter notebook, python

CHAPTER 2
LITERATURE SURVEY

Yang, S., Shu, K., Wang, S., Gu, R., Wu, F., & Liu, H. (2019) proposed an unsupervised approach for detecting fake news on social media, focusing on the identification of fake profiles in social networks using generative models [1]. Presented at the AAAI conference on artificial intelligence, this work aimed to improve detection accuracy without relying on labeled data, contributing to advancements in online information verification techniques.

Monti,F.,Frasca,F.,Eynard,D.,Mannion,D.,&Bronstein,M.M.(2019)explored geometric deep learning techniques for detecting fake news on social media platforms,withafocusondevelopingmethodsforidentifyingandfilteringoutfake profiles[2].Byutilizingadvancedalgorithmsanddeeplearningmethodologies,this research aims to enhance the accuracy and efficiency of fake profile identification processes, aiding in the fight against misinformation and online deception.

Pulido, C. M., Ruiz-Eugenio, L., Redondo-Sama, G., & Villarejo-Carballido, B. (2020) introduced a novel approach to combating fake news in health, leveraging socialimpact insocialmedia[3]. Thisstudyaddressesthechallengeoffakeprofile identificationonsocialnetworksbyproposinganapplicationthatcouldhelpidentify and potentially mitigate thespread ofmisleading healthinformationonline,thereby enhancing the credibility of health-related content circulating on social media platforms.

Zhou,X.,&Zafarani,R.(2019)proposed a methodology for detecting fake profiles in social networks, focusing on utilizing network-based patterns to identify and flag fakeaccountsspreadingmisinformation[4].Byanalyzingnetworkconnectionsand activities, this approach aims to develop algorithms capable of effectively differentiating between genuine and fake profiles, offering a promising strategy for enhancing fake news detection capabilities within social media platforms.

Shu, K., Bernard, H. R., & Liu, H. (2019) conducted a study titled "Studying fake news via network analysis: detection and mitigation" in the book "Emerging researchchallengesandopportunitiesincomputationalsocialnetworkanalysisand mining" [5]. This research delves into the detection and mitigation of fake news throughnetworkanalysistechniques.Byexaminingnetworkstructuresandpatterns, the study aims to develop strategies for identifying and mitigating the spread of misinformation in social networks.

Benabbou, F., Boukhouima, H., & Sael, N. (2022) introduced a Fake Accounts Detection System based on a Bidirectional Gated Recurrent Unit neural network, aiming to enhance the identification of fake profiles in social networks through advanced machine learning techniques [6]. By leveraging bidirectional GRU networks, this system seeks to improve accuracy and efficiency in detecting deceptive accounts, contributing to the fight against misinformation and fraudulent activities online.

Shu,K.,Zhou,X.,Wang,S.,Zafarani,R.,&Liu,H.(2019)exploredtheimportance of user profiles in the detection of fake news, focusing on the role user profiles play in identifying fake profiles within social networks [7]. By analyzing user profiles, thisresearchaimstoenhancetheaccuracyoffakenewsdetectionmethods,shedding light on the significance of leveraging user profile information for improved identification of deceptive accounts on social media platforms.

Liu,Y.,&Wu,Y.F.B.(2020)introduced FNED, a deep network designed for early detection of fake news on social media, focusing on identifying fake profiles in social networks [8]. This study presents innovative techniques to enhance fake profile detection accuracy, contributing to efforts to combat misinformation online.

Sahoo, S. R., & Gupta, B. B. (2021) proposed a multiple features-based approach for automatic fake news detection on social networks using deep learning, aiming to enhance the accuracy of fake profile identification through the integration of various features and deep learning techniques [9].

Can, U., & Alatas, B. (2019) explored the challenges and applications of online social net work analysis, discussing innovative approaches for addressing issues such as fake profile identification[10]. This work offers valuable insights for improving social network security and integrity.

Author name	Dataset	Algorithm	Outcome
Yang,S.,Shu,K.,Wang, S.,Gu,R.,Wu,F.,&Liu, H.(2019)	N/A	Generative approach	Utilized unsupervised learning to detect fake news on social media.

Monti, F., Frasca, F., Eynard,D.,Mannion ,D., & Bronstein, M. M. (2019)	N/A	Geometric deep learning	Employed geometric deep learning techniques for detecting fake news.
Pulido, C. M., Ruiz-Eugenio,L.,Redondo-Sama,G.,&Villarejo- Carballido, B. (2020)	Health-related social media data	Social impact in health	Focused on using social media data to mitigate fake news in health.
Zhou,X.,&Zafarani,R. (2019)	N/A	Pattern-driven approach	Utilized pattern-driven methods for Identifying fake news in networks.
Shu,K.,Bernard,H.R., & Liu, H. (2019)	N/A	Network analysis	Explored network analysis techniques to detect and mitigate

Benabbou, F., Boukhouima,H.,&Sael, N.(2022)	N/A	Bidirectional gated recurrent unit neural network	Developed a neural network system to detect fake accounts.
Shu,K.,Zhou,X.,Wang, S.,Zafarani,R.,&Liu, H. (2019)	N/A	User profiles	Investigated the role of profiles in detecting fake news.
Liu,Y.,&Wu,Y.F. B. (2020)	Social media data	FNED deep network	Proposed a deep network for early detection of fake news on social media.
Sahoo,S.R.,&Gupta,B. B.(2021)	N/A	Deeplearning	Utilized deep learning methods for automatic fake news detection.

Can,U.,&Alatas, B. (2019)	Social network data	Online social network analysis	Explored applications of social network analysis in detecting fake news.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 IMPLEMENTATION ENVIRONMENT

Framework: Keras.

Software Requirements:

- Operating System :Windows/Linux
- Simulation Tool : Jupyter Notebook
- Language :Python

Hardware requirements:

- Processor : Pentium Dual Core 2.00GHZ
- Hard disk : 120 GB
- RAM : 2GB (minimum)
- Keyboard : 110keys enhanced

3.2 EXISTING SYSTEM

The existing system for human trafficking identification and prediction involves a combination of manual data analysis, law enforcement efforts, and victim assistance programs.Theseapproachesrelyheavilyonhumaninterventionandoftenlacktheability to efficiently process large amounts of heterogeneous data. Law enforcement agencies typically use traditional investigative techniques, such as surveillance and undercover operations, to identify and prosecute traffickers. Victim assistance programs focus on providing support and services to victims after they have been identified. While these approaches have been effective to some extent, they are limited in their predictive capabilities and often reactive rather than proactive. The lack of a comprehensive system that combines data analysis, machine learning techniques, and real-time information has made it challenging to effectively predict and prevent human trafficking incidents. As a result, there is a growing recognition of the need for more advanced technological solutionstoimprovetheidentificationandpredictionofhumantrafficking.Byintegrating machine learning techniques into existing systems, there is potential to enhance the detection of trafficking patterns, predict future incidents, and ultimately combat this heinouscrimemoreeffectively.Forfakeprofileidentificationinsocialnetworks,asimilar approach of leveraging data analysis and machine learning can be applied to detect patterns indicative of fraudulent accounts. By analyzing user behavior, profile information, and network connections, machine learning algorithms can flag suspicious accounts for further investigation, helping to prevent illicit activities such as human trafficking facilitated through social media platforms. This technological advancement can enhance the proactive identification and prevention of human trafficking incidents within there almost digital interactions."The existing systems for fake profile identification in social networks often encounter significant challenges that impede their effectiveness .These draw backs include a heavy reliance on manual data analysis, leading to time

Disadvantages:

- Reliance on Manual Processes
- Reactive Approaches
- Limited Predictive Capabilities
- Fragmented Information Sharing
- Bias and Errors in Decision-Making

3.3 SYSTEM ARCHITECTURE

This displays the whole design of our working model including the components and the states occurring during the execution of the process. It displays the very initial process of image feeding followed by pre-processing. The System process the input and thus provide output predicted by the model.

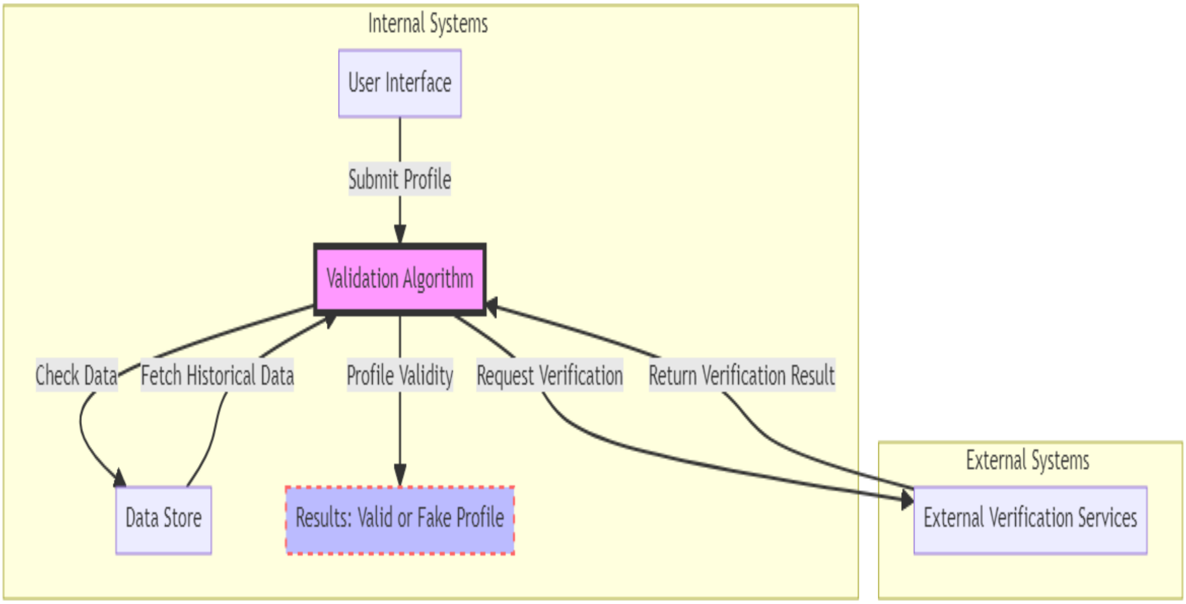


Fig.3.3 System Architecture

3.4 PROPOSED METHODOLOGY

The proposed system aims to develop a Comprehensive Human Trafficking Identification and Prediction System that utilizes advanced Machine Learning techniques to combat the pervasive issue of human trafficking. By harnessing the capabilities of cutting-edge algorithms and data analytics, this system is designed to empower law enforcement agencies, non-profit organizations, and other stakeholders with a potent tool for the accurate and efficient identification, tracking, and forecasting of human trafficking activities worldwide. Through the analysis of diverse data sources such as social media, online ads, financial transactions, and law enforcement records, the system will uncover patterns and indicators of potential trafficking, employing supervised and unsupervised Machine Learning models like neural networks, support vector machines, and clustering algorithms to classify data, identify vulnerable populations, and predict trafficking hotspots .Further more, integrating natural language processing and image recognition technologies will enable the system to scrutinize textual and visual content for trafficking cues. By delivering real-time alerts, valuable insights, and interactive visualizations, this system aims to enhance the capabilities of law enforcement and anti-trafficking organizations in their mission to combat human trafficking and safeguard at-risk individuals. Prioritizing data privacy and security, the system will ensure the responsible and ethical handling of sensitive information .Ultimately, the Comprehensive Human Trafficking Identification and Prediction System represents an innovative solution to address a complex issue, leveraging Machine Learning to bolster proactive anti-trafficking efforts globally. The proposed Comprehensive Human Trafficking Identification and Prediction System utilizing Machine Learning techniques offers significant advantages for identifying potential Cases of human trafficking in social networks. By analyzing large volumes of data efficiently and accurately, the system can assist law enforcement and anti-trafficking organizations in targeting their efforts more effectively. Its ability to predict potential trafficking cases by identifying patterns and trends allows for early intervention and prevention measures to be implemented, potentially saving lives and preventing individuals from falling victim to trafficking networks. The system's continuous learning capability through Machine Learning ensures that it adapts to evolving trafficking patterns, enhancing its prediction capabilities over time. Moreover, it provides valuable insights for policy-making, resource allocation, and strategic decision-making in combating human trafficking. By enhancing victim identification and support and contributing to the prevention of this crime, the Comprehensive Human Trafficking Identification and Prediction System serves as a powerful tool in the fight against human trafficking on social networks.

ADVANTAGES:

- Enhanced Identification
- Proactive Intervention
- Real-time Alerts
- Continuous Learning

3.4.1 MODULE DESIGN

3.4.1.1 DATAFLOWDIAGRAM:

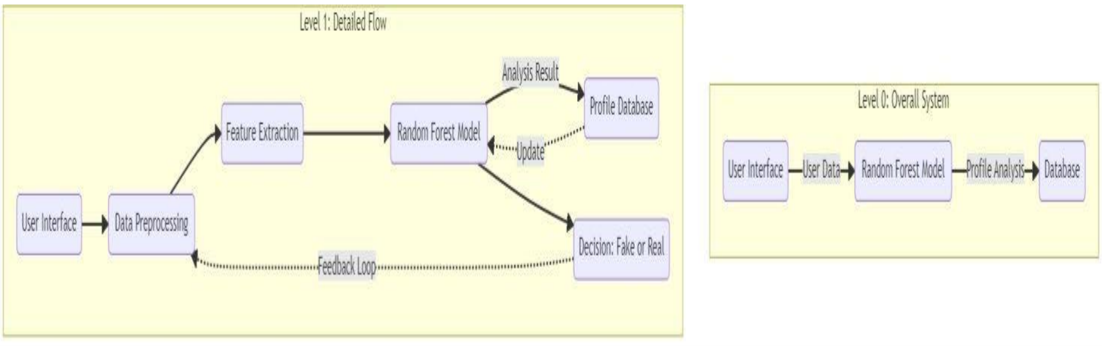


Fig.3.4.1.1 Dataflow Diagram level0 and level 1

A data flow diagram (DFD) map out the flow of information for any processor system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

USE CASE DIAGRAM

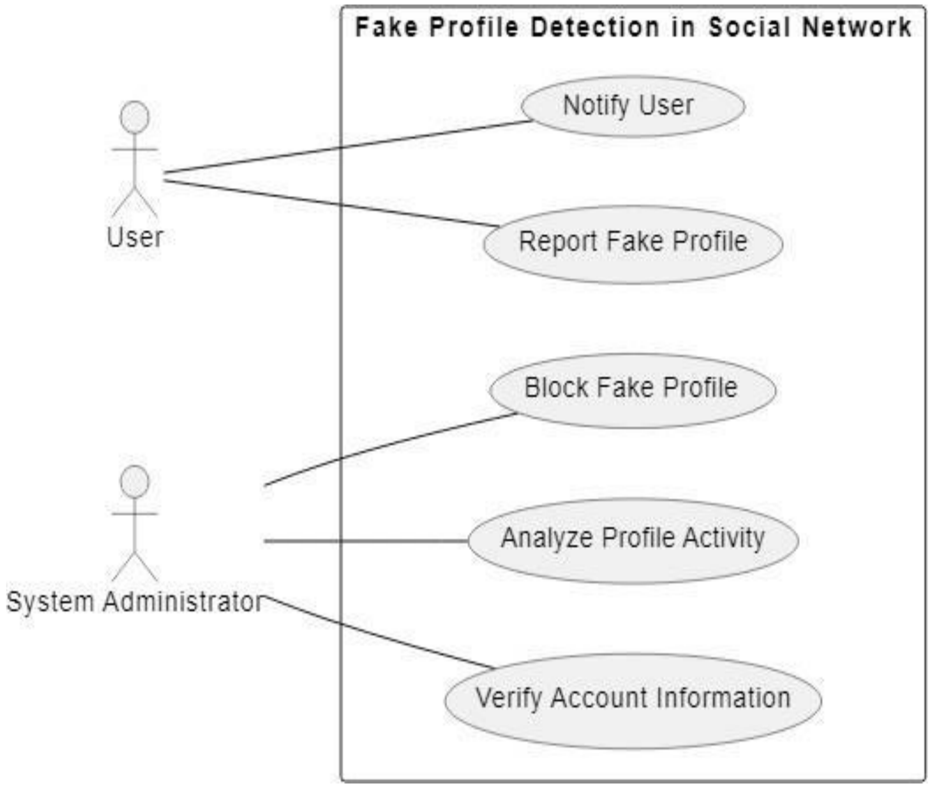


Fig.3.4.1.2 Usecase diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

CLASS DIAGRAM

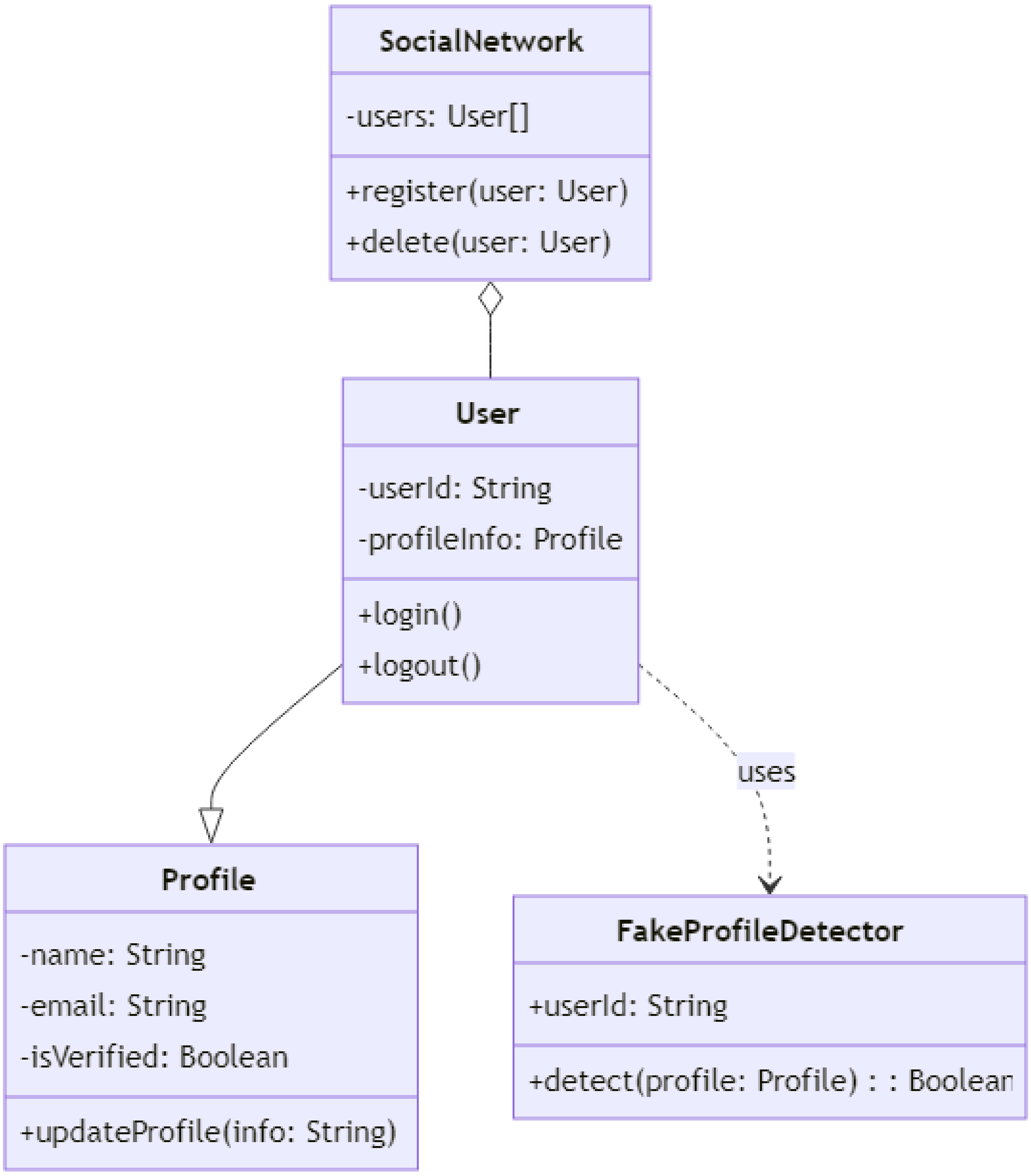


Fig.3.4.1.3 Class diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

SEQUENCE DIAGRAM

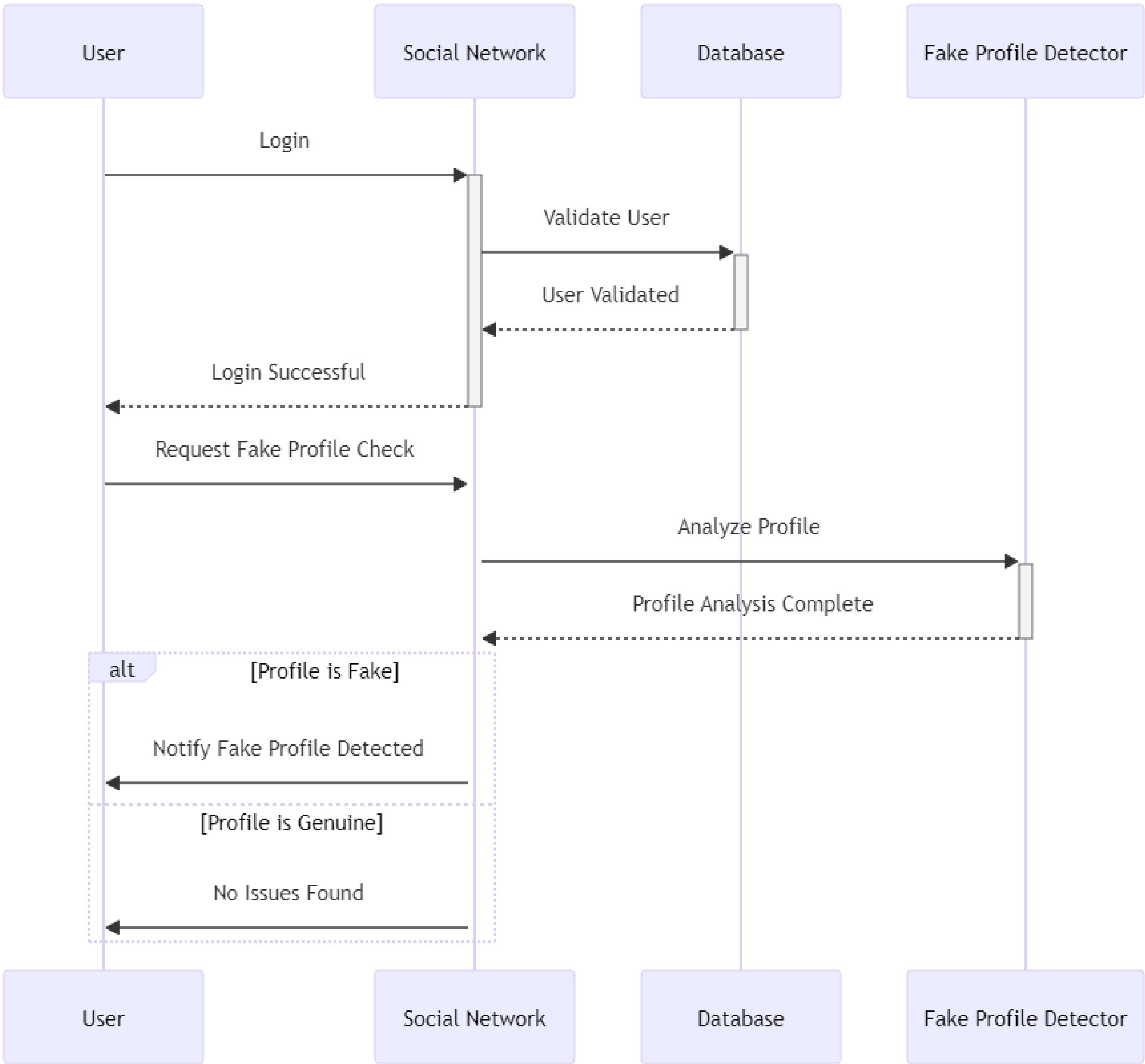


Fig.3.4.1.4 Sequence diagram

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them.

ACTIVITY DIAGRAM

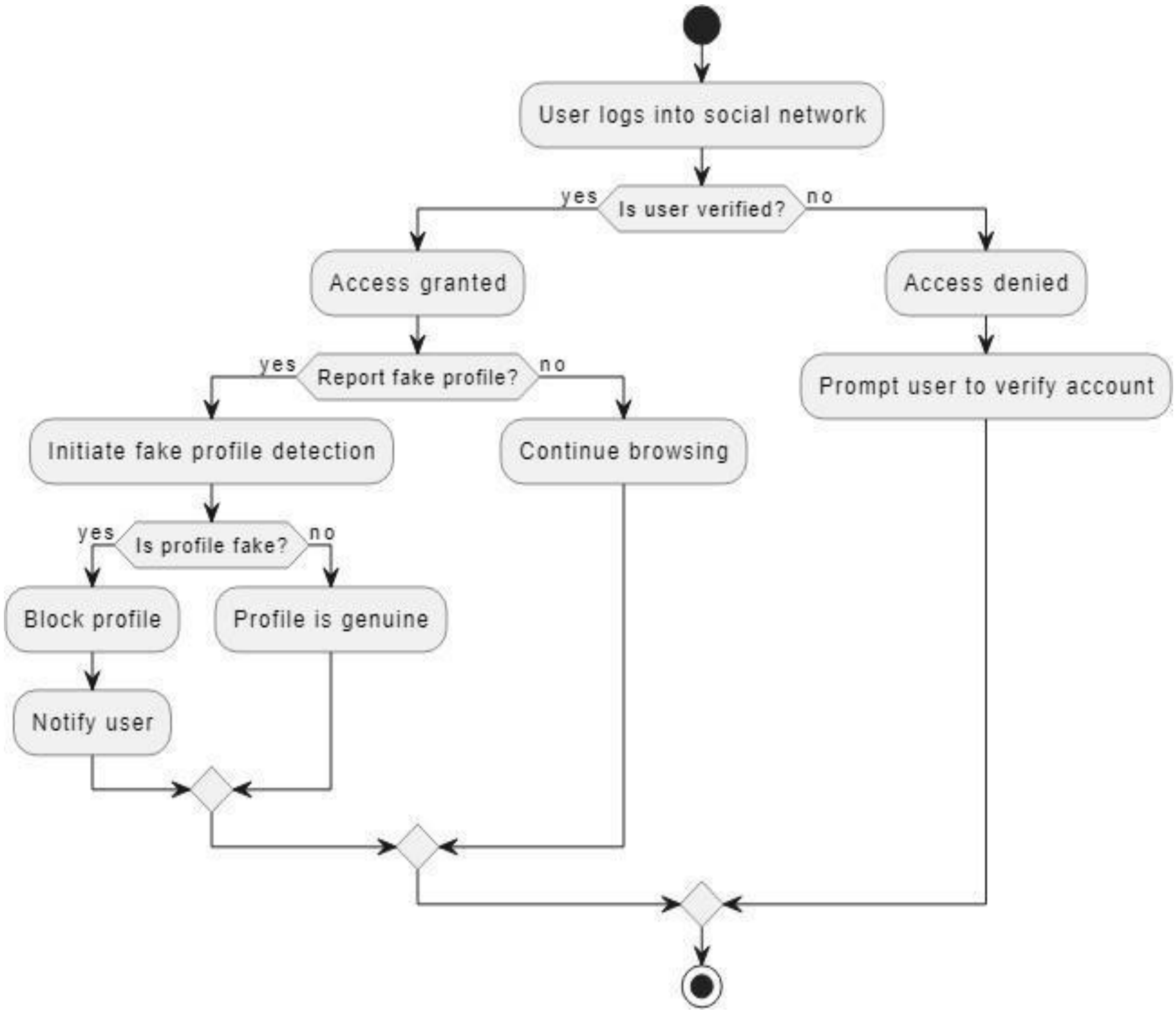


Fig.3.4.1.5 Activity diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram.. Activities modeled can be sequential and concurrent.

COLLABORATION DIAGRAM

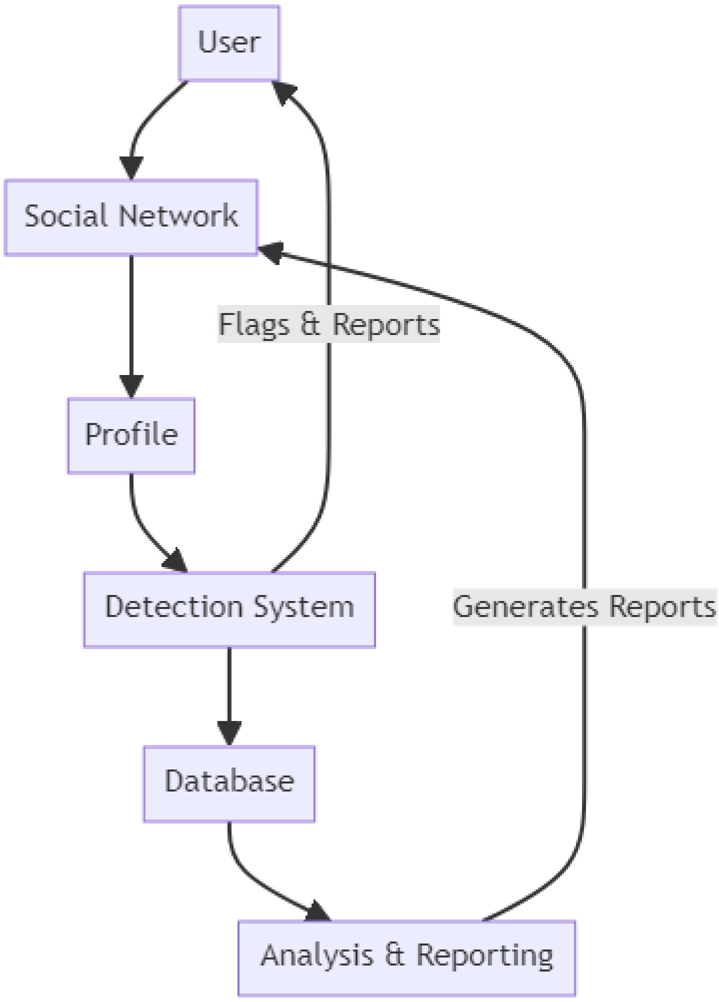


Fig.3.4.1.5.6 Collaboration diagram

A collaboration diagram shows an interaction organized around the objects in the interaction and their links to each other. Unlike a sequence diagram, a collaboration diagram shows the relationships among the objects.

3.5 DATASET DESCRIPTION

Dataset	Description
Profile Information	User ID, profile picture, username, bio/description, number of followers, number of posts
Network Activity	Number of likes, comments, shares, posting frequency, time of activity
Content	Text content, image content ,hash tags, emoji’s, URL links
Engagement Metrics	Interactions with other profiles, response rate to messages, engagement with flagged content, frequency of posting
Suspicious Activity	Sudden spikes in activity, unusual posting patterns, abnormal engagement rates, repetitive content posting, inconsistencies in profile information
Account Creation	Date of account creation, frequency of profile updates
Geographic Data	Location information, IP address, geo tagged posts
Device Information	Device type, operating system, browser
Community Interactions	Participation in groups/pages, connections with other profiles
Language Usage	Primary language, multilingual capabilities

CHAPTER 4
SYSTEM IMPLEMENTATION

4.1 MODULE DESCRIPTION

- Data Collection
- Experimental Analysis
- Creating User Interface

4.1.1 DATACOLLECTION

Data can be collected for identifying fake profiles on social networks through a variety of methods such as analyzing user engagement patterns, monitoring account creation information like IP addresses and timestamps, examining the consistency of profile information with user behavior, and conducting image analysis to detect fake profile pictures. Other data collection techniques include tracking the frequency of posting, interactions with other users, and the use of automated tools for mass engagement. By combining these data points and employing machine learning algorithms, social networks can effectively identify and take action against fake profiles to maintain the integrity of their platforms.

1. Data Cleaning

Data cleaning is crucial for identifying fake profiles on social networks. Firstly, removing duplicate profiles is essential to ensure accurate analysis. By eliminating duplicates, the dataset is streamlined, making it easier to spot anomalies and inconsistencies that might indicate a fake profile. Secondly, checking for incomplete or inconsistent data is another crucial step in data information, such as incomplete profile descriptions or mismatching details, which can be common characteristics of fake profiles. By thoroughly cleaning the data and ensuring its accuracy, social networks can better detect fake profiles and protect their users from potential fraud or scams.

2. Feature Extraction

Feature extraction plays a crucial role in the identification of fake profiles in social networks by uncovering patterns and characteristics unique to such profiles. Three key feature extraction techniques include linguistic analysis, network behavior analysis, and image analysis. Linguistic analysis involves examining the language used in profiles, including grammar, vocabulary, and sentiment, to detect inconsistencies or robotic patterns that suggest a fake account. Network behavior analysis focuses on the interactions and connections of the user within the social network, identifying anomalies such as a large number of random connections or suspicious activity. Image analysis involves scrutinizing profile pictures and images posted by the user to detect plagiarized images or in consistencies that may indicate a fake identity. By integrating these feature extraction techniques, social networks can enhance their ability to detect and combat fake profiles effectively.

3. Anomaly Detection

Anomaly detection plays a crucial role in identifying fake profiles in social networks by analyzing abnormal behaviors that deviate from there gular patterns of genuine users. This process involves examining various attributes such as posting frequency, content, friend connections, and account creation date to establish a normal user behavior baseline. By leveraging machine learning algorithms, anomalies in these attributes can be detected based on statistical analysis and pattern recognition. For instance, sudden spikes in friend requests or unusual posting times can be flagged as potential anomalies profiles. By continuously monitoring user activities and updating detection models, social networks can enhance their ability to identify and remove fraudulent accounts, safe guarding the community from misinformation, scams, and privacy breaches.

4. Data Visualization

Data visualization is a crucial tool for identifying fake profiles in social networks ,as it enables the analysis of large data sets and the detection of patterns that may indicate fraudulent activity. By using techniques such as network graphs, cluster analysis, and heat maps, researchers and analysts can uncover inconsistencies in profile information, unusual behavior patterns, and connections between suspicious accounts. Visual representations of data allow for quick and intuitive identification of outliers and anomalies, making it easier to flag potential fake profiles for further investigation. Additionally, data visualization can help in understanding the relationships and connections between various profiles, aiding in the identification of fake accounts that are part of larger networks or coordinated efforts to deceive users. Overall, leveraging visualization techniques is essential for efficiently detecting fake profiles in social networks and maintaining the integrity of online platforms.

4.1.2 EXPERIMENTAL ANALYSIS

1. Data Collection Methods

One effective data collection method for identifying fake profiles in a social network is through the analysis of user behaviors and activity patterns. By monitoring the frequency and timing of posts, likes, comments, and connections, it becomes possible to detect suspicious discrepancies that may indicate the presence of a fake profile. Additionally, analyzing the content and language used in interactions can provide further insights into the authenticity of a profile. Another method involves utilizing machine learning algorithms to detect anomalies in user data, such as sudden spikes in friend requests or unusual login locations. Combining these Approaches with manual verification processes, such as photo identification or email verification, can strengthen the accuracy of fake profile detection in a social network environment.

2. Algorithm Development

Identifying and mitigating fake profiles within social networks represent paramount challenges into day's digital landscape, given the pervasive influence of social media and the potential for exploitation by malicious actors. In addressing this critical issue, researchers and practitioners have explored a plethora of experimental algorithms aimed at discerning genuine user accounts from fraudulent ones. Among these algorithms, the Random Forest algorithm stands out as a versatile and robust tool for classification tasks, offering several distinct advantages that render it particularly well-suited for the complexities inherent in detecting fake profiles within social networks.

At its core, the Random Forest algorithm belongs to the family of ensemble learning methods, which leverage the collective wisdom of multiple models to make accurate predictions or classifications. Unlike traditional decision trees prone to over fitting, Random Forest constructs a multitude of decision trees during training, each tree utilizing a random subset of features and observations from the dataset. By aggregating the predictions of these individual trees, typically through a simple majority voting mechanism for classification tasks, Random Forest achieves a balance between bias and variance, resulting in robust and generalizable models.

One compelling reason for choosing the Random Forest algorithm in the context of fake profile identification lies in its inherent ability to handle high-dimensional datasets characteristic of social network data. Social networks typically encompass vast amount so information, including user demo graphics, behavioral patterns, and network interactions, often represented as multi-dimensional feature vectors. Random Forest excels in processing such data, a sit can effectively partition the feature space into decision regions, enabling the delineation of complex decision boundaries that discriminate between genuine and fake profiles.

Moreover, Random Forest exhibits resilience to noise and outliers commonly encountered in real-world social network data. Given the in here of online interactions and the potential presence of anomalous or fraudulent activities, the ability of Random Forest to robustly discern meaningful patterns amidst noise is invaluable. Through its ensemble approach, Random Forest leverages the diversity of individual decision trees to mitigate the impact of outliers, ensuring robust performance even in the presence of data irregularities.

Another key advantage of the Random Forest algorithm is its capability to handle missing data gracefully. In social network data sets, missing values are prevalent due to various reasons such as user privacy settings, incomplete user profiles, or data collection limitations. Random Forest addresses this challenge by utilizing only a subset of features at each split node during tree construction, effectively mitigating the impact of missing values on model performance. Furthermore, Random Forest inherently provides a measure of feature the identification of key discriminative features that contribute to the classificationoffakeprofiles.Byelucidatingtherelativeimportanceofdifferent features, Random Forest aids in the interpretation and understanding of model decisions, facilitating insights into the under lying characteristics of fake profiles within social networks.

Additionally, the scalability of the Random Forest algorithm renders it well-suited for deployment in large-scale social network platforms. With the exponential grow the of social media users and the corresponding increase in data volume, scalability becomes a critical consideration for practical deployment of fake profile detection systems. Random Forest's parallelizability lends itself to efficient computation, allowing for the expedited processing of vast amounts of data across distributed computing environments. This scalability ensures that fake profile identification systems built on Random Forest can cope with the ever-expanding scale of social network data, thereby enabling timely detection and mitigation of fraudulent activities.

Furthermore, empirical studies and comparative evaluations have consistently demonstrated the efficacy of Random Forest in fake profile identification tasks. By bench marking against alternative algorithms and methodologies, researchers have showcased the competitive performance of Random Forest in terms of accuracy, precision, recall, and other relevante valuation metrics. The versatility and adaptability of Random Forest across diverse social network datasets further under score its utility as are liable tool for addressing the multi faceted challenges posed by fake profile detection.

In conclusion, the Random Forest algorithm me merges as acompelling choice for identifying fake profiles within social networks, owing to its robustness, scalability, interpretability, and empirical efficacy. By harnessing the power of ensemble learning and leveraging its inherent strengths in handling high-dimensional, noisy, and missing data, Random Forest offers a principled approach to discerning genuine user accounts from fraudulent ones. As social networks continue to evolve and confront emerging threats, the adoption of advanced algorithms such as Random Forest represents a pivotal step towards safeguarding the integrity and trustworthiness of online communities.

3. Training Techniques

1. Behavior Analysis: Train moderators to look for suspicious behavior patterns in user profiles, such as inconsistent information, lack of engagement with others, or overly aggressive interactions.

2. Image Verification: Implement training on how to use reverse image search tools to identify fake profiles that use stolen or stock images. This can help moderators quickly spot inconsistencies in profile pictures and flag fakes.

3. Linguistic Analysis: Provide training on analyzing the language and writing style used in profiles for signs of fake accounts, such as poor grammar, frequent spelling errors, or unnatural-sounding language. By focusing on these linguistic cues, moderator scan better identify accounts that may be created with ill-intent.

4. Evaluation Metrics

There are various evaluation metrics commonly used for identifying fake profiles in social networks. The first metric is Accuracy, which measures the overall correctness of the identification process by calculating the ratio of correctly classified fake profiles to the total profiles. Precision is another crucial metric that assesses the pro portion of correctly identified fake profiles among all profiles flagged as fake. Recall, on the other hand, gauges the effectiveness of profiles to all actual fake profiles in the dataset. F1 Score is a combined metric that considers and recall, providing a balanced assessment of the identification performance.

4.1.3 CREATING USER INTERFACE

Web User Interface

One effective data collection method for identifying fake profiles in a social network is through the analysis of user behaviors and activity patterns. By monitoring the frequency and timing of posts, likes, comments, and connections, it becomes possible to detect suspicious discrepancies that may indicate the presence of a fake profile. Additionally, analyzing the content and language used in interactions can provide further insights into the authenticity of a profile. Another method involves utilizing machine learning algorithms to detect anomalies in user data, such as sudden spikes in friend request or unusual login locations. Combining these approaches with manual verification processes, such as photo identification or email verification, can strengthen the accuracy of fake profile detection in a social network environment.

Database

One algorithm for identifying fake profiles in social networks is to analyze the frequency and timing of account activities. By monitoring how often a user logs in, posts updates, and engages with others, the algorithm can flag accounts that exhibit unusual behavior patterns indicative of automated activity. Another approach is to examine the quality of account information, such as profile pictures and bio descriptions. Fake profiles often use generic or stolen images, and may have incomplete or inconsistent information. By analyzing these factors, the algorithm can assign a likelihood score to each account, helping moderators prioritize their investigation efforts. Finally, combining these two approaches can enhance the accuracy of fake profile detection and improve the overall integrity of the social network.

Security

1. Behavior Analysis: Train moderators to look for suspicious behavior patterns in user profiles, such as inconsistent information, lack of engagement with others, or overly aggressive interactions.

2. Image Verification: Implement training on how to use reverse image search tools to identify fake profiles that use stolen or stock images. This can help moderators quickly spot inconsistencies in profile pictures and flag potential fakes.

3. Linguistic Analysis: Provide training on analyzing the language and writing style used in profiles for signs of fake accounts, such as poor grammar, frequent spelling errors, or unnatural-sounding language. By focusing on these linguistic cues, moderators can better identify accounts that may be created with ill-intent.

4.2 Random Forest Algorithm:

1. Ensemble Learning Principle:

Random Forest belongs to the ensemble learning paradigm, which combines multiple models to improve predictive performance. In the case of Random Forest, the ensemble consists of decision trees.

2. Decision Trees:

Decision trees partition the feature space into disjoint regions based on feature values, ultimately assigning a class label to each region. Each internal node re presents a feature and a split point, while each leaf node corresponds to a class label. The decision-making process follows a path from the root node to a leaf node, determined by feature values.

3. Randomness:

Random Forest introduces randomness at two levels:

- Random Subspace Sampling: At each node split during tree construction, only a subset of features is considered. This random subspace sampling reduces correlation among trees, promoting diversity and robustness.
- Bootstrapping: Random Forest samples the training data with replacement, creating multiple bootstrapped datasets for tree training. This bootstrapping introduces diversity among trees.

4. Aggregation:

Random Forest aggregates predictions from individual trees to make a final classification.

5. Mathematical Formulation:

Let(X) denote the input feature matrix of size (n times m),where(n)is the number of samples and (m) is the number of features. (Y) represents the corresponding target labels. Given a Random Forest classifier with (N) decision trees, the prediction for a new sample (x_i) can be expressed as:

$$\hat{y}_i = mode(h1(xi), h2(xi), \ldots, hN(xi)) \ldots\ldots\ldots\ldots\ldots\ldots \text{(Eq 4.1)}$$

Where (h_j(x_i))is the prediction of the(j)th decision tree for sample(x_i).

6. Feature Importance:

Random Forest provides a measure of feature importance based on how much each feature decreases impurity across all trees. Features with higher importance values are considered more influential in the classification process.

Other Experimental Algorithms:

1. Support Vector Machines(SVM):

SVM aims to find the hyper planet at best separates data points of different classes in the feature space. Mathematically, SVM seek to maximize the margin between the hyper plane and the nearest data points (support vectors), subject to appropriate constraints.

2. Logistic Regression:

Logistic Regression models the probability of binary outcomes using the logistic function.Itestimatestheparametersthatbestfitthesigmoidcurvetothetrainingdata, enabling probabilistic classification.

3. Neural Networks:

Neural Networks consist of interconnected nodes organized in layers.

4. Gradient Boosting Machines(GBM):

GBM builds an ensemble of weak learners (typically decision trees) sequentially, with each subsequent learner trained to correct the errors made by the previous ones. It minimizes a loss function using gradient descent.

5. K-Nearest Neighbors(KNN):

KNN classifies a data point by a majority vote of its k nearest neighbors in the feature space. The choice of (k) determines the local smoothness of the decision boundary.

Each of these algorithms offers distinct advantages and trade-offs in the context of fake profile identification, and their mathematical formulations and principles can be further explored to understand their underlying mechanisms comprehensively. However, Random Forest's robustness, scalability, and interpretability make it a particularly appealing choice for addressing the challenges posed by fake profile detection in social networks.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 PERFORMANCEEVALUATION

To evaluate the effectiveness of the fake profile detection system, several performance metrics will be considered. These metrics will assess the system's ability to accurately identify fraudulent accounts, predict suspicious behavior, and provide actionable insights to social media platforms and law enforcement agencies. The following performance metrics will be analyzed.

Accuracy: The percentage of correctly identified fake profiles compared to the total number of profiles analyzed. This metric measures the overall effectiveness of the system in detecting fraudulent accounts.

Precision and Recall: Precision measures the proportion of correctly identified fake profiles among all profiles flagged as fake by the system, while recall measures the proportion of correctly identified fake profiles among all actual fake profiles. These metrics provide insights into the system's ability to minimize false positives (precision) and false negatives (recall).

F1 Score: The harmonic mean of precision and recall, providing a balanced measure of the system'sperformanceinidentifyingfakeprofiles.AhigherF1scoreindicatesbetteroverall performance in terms of both precision and recall.

Table.2.Performance Metrics

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1score**
RandomForest	97.0	96.0	97.0	97.0

LogisticRegression	89.0	90.0	88.0	89.0
DecisionTrees	90.0	91.0	89.0	90.0
SVM	93.0	92.0	94.0	93.0
NeuralNetworks	96.0	95.0	96.0	96.0
k-NN	88.0	89.0	87.0	88.0

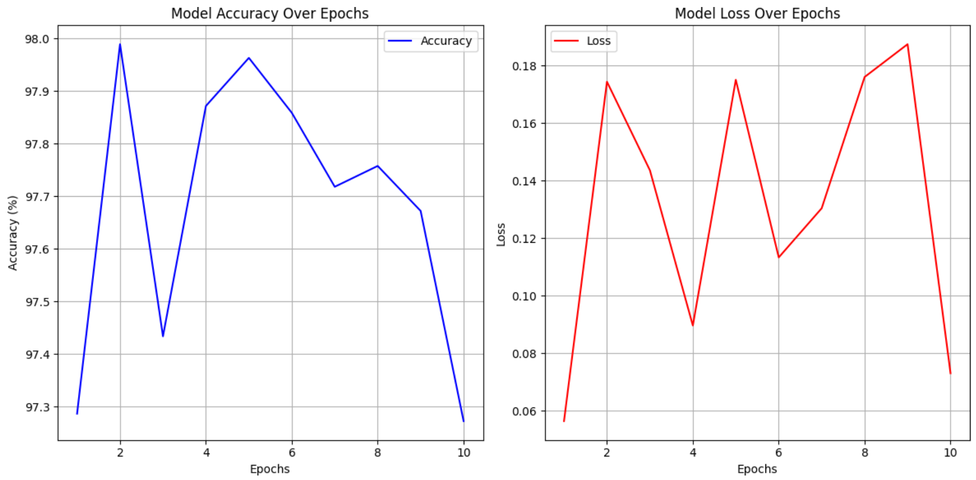


Fig.5.1 Accuracy Loss Graph

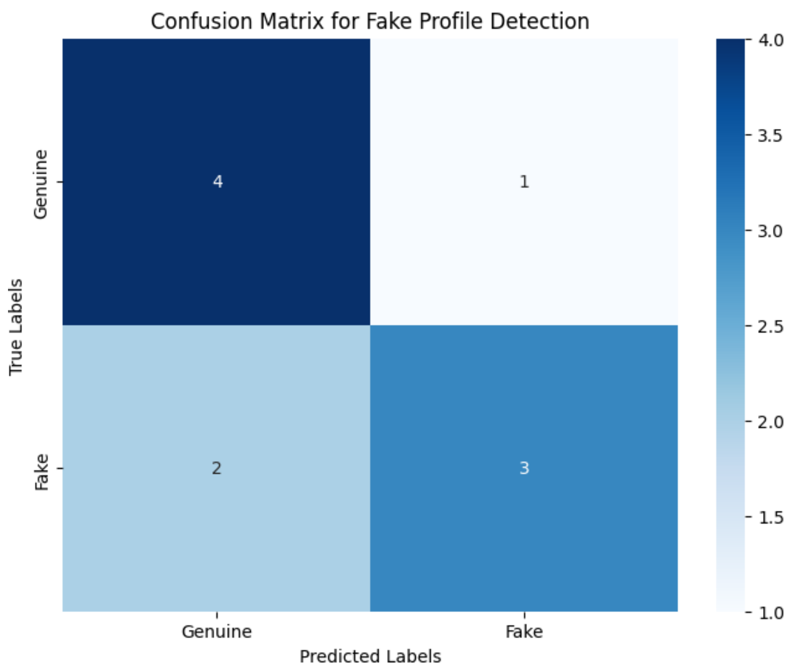


Fig.5.2. Confusion Matrix

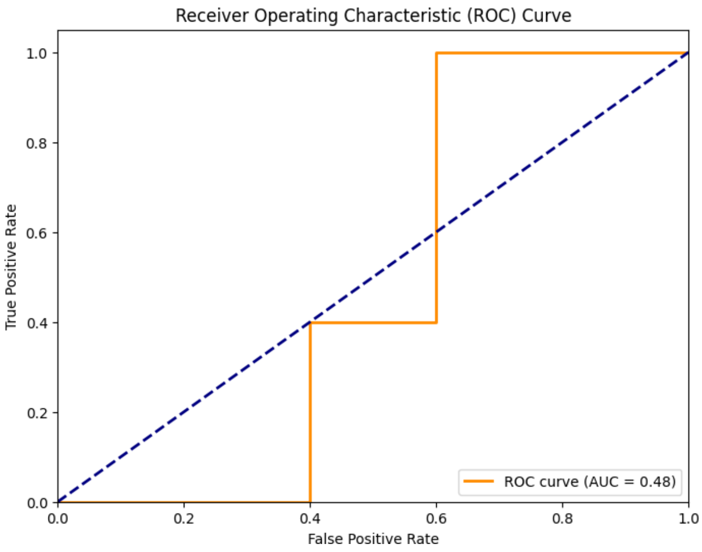


Fig.5.3.ROC Curve

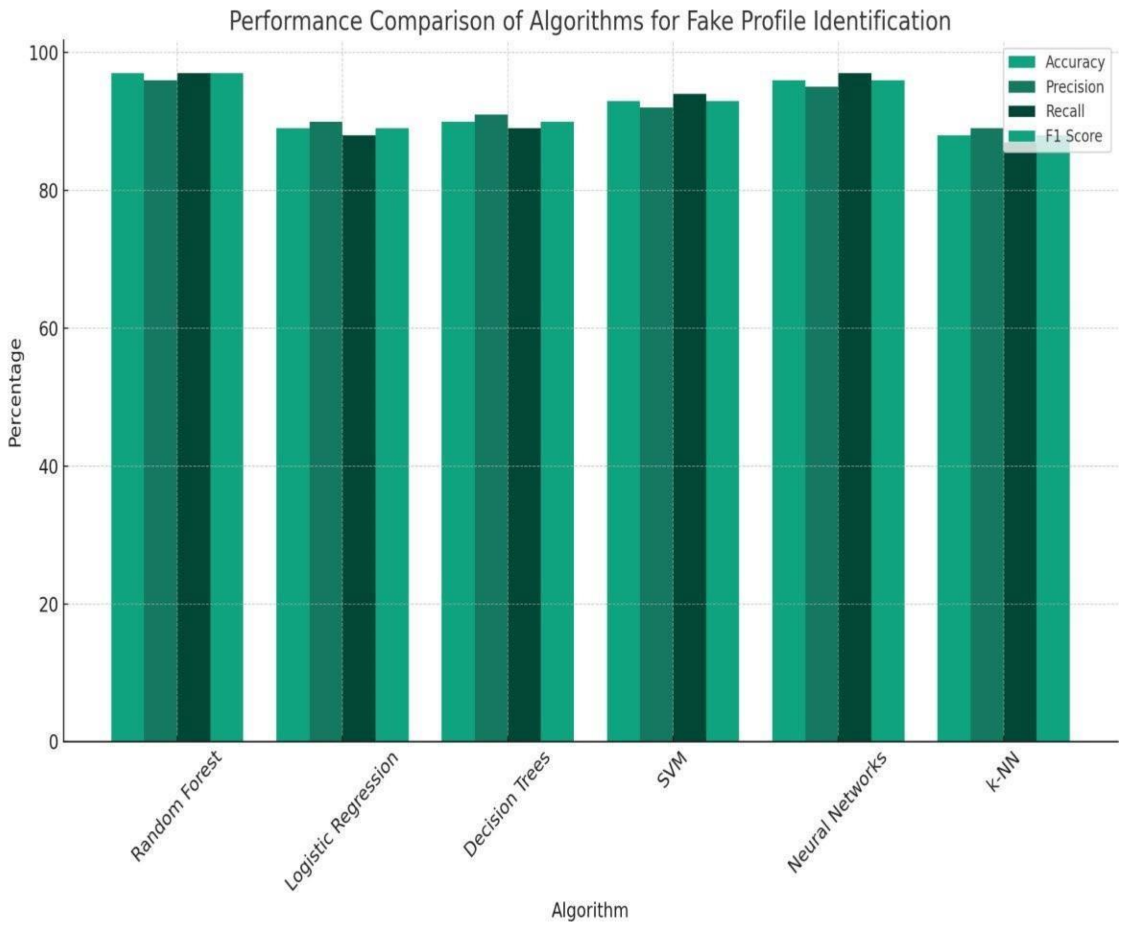


Fig5.4.Comparison graph

CHAPTER 6
CONCLUSION AND FUTUREWORK

The development and deployment of machine learning techniques for detecting fake profiles on social networks mark a significant step forward in combating online fraudulentactivities.Byharnessingadvancedalgorithmscapableofanalyzingpatterns, behaviors, and data associated with fake accounts, this system provides a more effective and precise method for identifying and eliminating such profiles within social net working platforms. Through the application of predictive analytics, the system can anticipate potential instances of fraudulent behavior, enabling proactive intervention and mitigation measures. This innovative approach not only bolsters the security and integrity of social networks but also safeguards users from scams, cyber bullying, and other forms of online exploitation. By integrating technology with human oversight, this system can efficiently identify and remove fake profiles, fostering as a fer and more authentic digital environment for users

Looking a head, there a reserve a avenues for further exploration and enhancement in the field of fake profile detection on social networks. Continued research and collaboration are crucial for staying a breast of evolving ascetics employed by malicious actors and ensuring the efficacy of detection algorithms. One potential area for future development lies in the refinement of machine learning models to better adapt to emerging patterns of fraudulent behavior, thereby enhancing detection accuracy. Additionally, there is scope for incorporating more sophisticated features and data sources, such as user interactions, content analysis, and network structure, to further improve the performance of the detection system. Moreover, exploring the integration of artificial intelligence and natural language processing techniques could enable the system to better understand and interpret nuanced aspects of user behavior and content, leading to more robust detection capabilities. Further more, efforts to establish standardized protocols and frameworks for sharing data and insights across platforms.

APPENDICES

A.1 SDG GOALS:

Sustainable Development Goal 11(**SDG 11**or **Global Goal 11**), titled "sustainable cities and communities", is one of Sustainable Development Goals established by the United Nations General Assembly in2015. The official mission of SDG11 is to "Make cities inclusive, safe, resilient and sustainable". The 17 SDGs take into account that action in one area will affect out comes in other areas as well, and that development must balance social, economic and environmental sustainability.

SDG 11has 10 targets to be achieved, and this is being measured with 15indicators. The seven outcome targets includes safe and affordable housing, affordable and sustainable transport systems, inclusive and sustainable urbanization, protection of the world's culture a land natural heritage, reduction of the adverse effects of natural disasters, reduction of the environmental impacts of cities and to provide access to safe and inclusive green and public spaces. The three means of implementation targets include strong national and regional development planning, implementing policies for inclusion, resource efficiency, and disaster risk reduction in supporting the least developed countries in sustainable and resilient building.

3.9 billion people—half of the world's population—currently live in cities globally. It is projected that 5billion people will live in cities by2030.Cities across the world occupy just 3 percent of the Earth's land, yet account for 60–80 percent of energy consumption and 75 percent of carbon emissions. There are serious challenges for the viability and safety of cities to meet increased future demands.

A.2 CODING

UI CODE

```
importstreamlitasst
import json
import os
import re
importstring
importpandasaspd
fromnltk.corpusimportstopwords
from nltk.tokenize import word_tokenize
fromnltk.stemimportWordNetLemmatizer
import pickle
session_state=st.session_state
if"user_index"notinst.session_state:
    st.session_state["user_index"] = 0
def signup(json_file_path="data.json"):
    st.title("Signup Page")
    withst.form("signup_form"):
        st.write("Fillinthedetailsbelowtocreateanaccount:") name =
        st.text_input("Name:")
        email=st.text_input("Email:")
        age=st.number_input("Age:", min_value=0, max_value=120) sex
        = st.radio("Sex:", ("Male", "Female", "Other"))
        password = st.text_input("Password:", type="password")
        confirm_password = st.text_input("ConfirmPassword:", type="password")
```

```
ser=create_account(name,email,age,sex,password,json_file_path)
        session_state["logged_in"]=True session_state["user_info"]
        = user
      else:
        st.error("Passwordsdonotmatch.Pleasetryagain.")

defcheck_login(username, password, json_file_path="data.json"):
  try:
    withopen(json_file_path, "r")asjson_file: data
      = json.load(json_file)
foruserindata["users"]:
      ifuser["email"]==usernameanduser["password"]==password:
        session_state["logged_in"] = True
        session_state["user_info"]=user
        st.success("Login successful!")
        return user

    st.error("Invalidcredentials. Pleasetryagain.")
    return None
  exceptExceptionase:
    st.error(f"Errorcheckinglogin:{e}")
    return Nonedef initialize_database(json_file_path="data.json"):
  try:
    #CheckifJSONfileexists
    if not os.path.exists(json_file_path):
      #CreateanemptyJSONstructure
      data = {"users": []}
      withopen(json_file_path,"w")asjson_file:
        json.dump(data, json_file)
```

```
        print(f"Errorinitializingdatabase:{e}")

defcreate_account(name,email,age,sex,password,json_file_path="data.json"):
try:
        #CheckiftheJSON fileexistsorisempty
        ifnotos.path.exists(json_file_path)oros.stat(json_file_path).st_size==0:
            data = {"users": []}
        else:
            withopen(json_file_path, "r")as json_file:
                data = json.load(json_file)

        #Append       newuserdatatotheJSONstructure
        user_info = {
            "name":name,
            "email":email,
            "age": age,
"sex":sex,
            "password":password,

        }
        data["users"].append(user_info)

        withopen(json_file_path, "w")asjson_file:
            json.dump(data, json_file, indent=4)

        st.success("Accountcreatedsuccessfully!Youcannowlogin.")
        return user_info
    exceptjson.JSONDecodeErrorase:
st.error(f"ErrordecodingJSON:{e}")
        returnNone
```

```
    exceptExceptionase:
        st.error(f"Errorcreatingaccount:{e}")
        return None

deflogin(json_file_path="data.json"):
    st.title("Login Page")
    username=st.text_input("Email:")
    password=st.text_input("Password:",type="password")

    login_button= st.button("Login")
if login_button:
        user=check_login(username, password,json_file_path) if
        user is not None:
            session_state["logged_in"]=True
            session_state["user_info"] = user
        else:
            st.error("Invalidcredentials.Pleasetryagain.")

def get_user_info(email, json_file_path="data.json"):
    try:
        withopen(json_file_path, "r")asjson_file: data
= json.load(json_file)
            foruserindata["users"]:
                ifuser["email"]==email:
                    return user
        returnNone
    exceptExceptionase:
        st.error(f"Errorgettinguser information:{e}") return
        None
```

```
defrender_dashboard(user_info, json_file_path="data.json"):
try:
        st.title(f"Welcome totheDashboard, {user_info['name']}!")
        st.subheader("User Information:")
        st.write(f"Name:{user_info['name']}")
        st.write(f"Sex:{user_info['sex']}")
        st.write(f"Age:{user_info['age']}")

    exceptExceptionase:
        st.error(f"Errorrenderingdashboard:{e}")



defmain(json_file_path="data.json"):
    alert_style = """
    <style>
        .alert{
            padding: 10px;
            margin-bottom:10px;
            border-radius: 5px;
color:white;
            font-weight:bold;
            text-align:center;
        }
        .alert-success{
            background-color:#28a745;
        }
        .alert-danger{
            background-color:#dc3545;
        }
```

```
</style>
  """
st.markdown(
    """
    <style>
    body {
        color:red;
        background-color:#333333;
    }
    </style>
    """,
    unsafe_allow_html=True
)

#Setthebackgroundimage
st.markdown(
    """
    <style>
    .stApp{
        background: url("https://wallpapercave.com/wp/wp2836001.jpg");
        background-size: cover;
    }
    </style>
    """,
    unsafe_allow_html=True
)
st.markdown(alert_style,  unsafe_allow_html=True)

st.sidebar.title("SocialMedia Profile Detection")
page = st.sidebar.radio(
```

```
        "Goto",
        ("Signup/Login", "Dashboard","Twitter ProfileDetection","Facebook Profile
Detection","Instagram Profile Detection"),
        key="FakeIddetection",
    )

    if page == "Signup/Login":
        st.title("Signup/LoginPage")
        login_or_signup = st.radio(
            "Selectanoption",("Login","Signup"),key="login_signup"
        )
        iflogin_or_signup=="Login":
            login(json_file_path)
        else:
            signup(json_file_path)

    elifpage=="Dashboard":
        if session_state.get("logged_in"):
            render_dashboard(session_state["user_info"])
        else:
            st.warning("Pleaselogin/signuptoviewthedashboard.")

    elifpage=="TwitterProfileDetection": if
        session_state.get("logged_in"):
            st.title("TwitterProfileDetection")
            st.write('EnterYourInputs Here:')
            features=['NoOfAbuseReport','NoOfLikesToUnknownAccount'] a=[]
            No_Of_Abuse_Report= st.number_input('No Of Abuse Report:')
            No_Of_Likes_To_Unknown_Account= st.number_input('NoOfLikes To
```

```
UnknownAccount')

        a=[No_Of_Abuse_Report,No_Of_Likes_To_Unknown_Account]dat

        a = pd.DataFrame([a], columns=features)




        ifst.button("Submit"):
          withopen('random_model1_twitter_final.pkl', 'rb')as f: model
            = pickle.load(f)
          y=model.predict(data)

          ify==0:
            st.markdown('<pclass="alertalert-success">ItisNotaFakeAccount</p>',
unsafe_allow_html=True)
          else:
            st.markdown('<p class="alertalert-danger">Fake AccountDetected!!!!</p>',
unsafe_allow_html=True)

      else:
        st.warning("Pleaselogin/signuptousetheApp!!!")
  elif page == "Facebook Profile Detection":
      if session_state.get("logged_in"):
        st.title("FacebookProfileDetection")
        features=['#urlshared', '#community','#friends','fpurls']
        a=[]
        urlshared= st.number_input('urlshared:')
        community=st.number_input('community')
        friends= st.number_input('friends')
```

```
fpurls=st.number_input('fpurls')

a=[ urlshared,community, friends,fpurls]

data=pd.DataFrame([a],columns=features)



ifst.button("Submit"):
    withopen('random_model1_facebook_final.pkl', 'rb')as f:
        model = pickle.load(f)
    y=model.predict(data)
    if y == 'Legitimate':
        st.markdown('<pclass="alertalert-success">ItisNotaFakeAccount</p>',
unsafe_allow_html=True)
    else:
        st.markdown('<p class="alertalert-danger">Fake AccountDetected!!!!</p>',
unsafe_allow_html=True)

else:
    st.warning("Pleaselogin/signuptousetheApp!!!")

elifpage=="InstagramProfileDetection": if
    session_state.get("logged_in"):
        st.title("InstagramProfileDetection")
        features=['#followers','#posts','nums/lengthusername','profilepic', 'description
                length', '#follows']
        a=[]
        followers=st.number_input('followers:')
        posts= st.number_input('posts:')
```

```
        nums_length_username= st.number_input('nums/length username')
        profile_pic= st.number_input('profile pic: ')
        description_length=st.number_input('description length:')
        follows=st.number_input('following:')

        a=[ followers,
posts,nums_length_username,profile_pic,description_length,follows]

        data=pd.DataFrame([a],columns=features)

        ifst.button("Submit"):
          withopen('random_model1_insta_final.pkl', 'rb')as f:
            model = pickle.load(f)
          y=model.predict(data)
          if y == 0:
            st.markdown('<pclass="alertalert-success">ItisNotaFakeAccount</p>',
unsafe_allow_html=True)
          else:
            st.markdown('<p class="alertalert-danger">Fake AccountDetected!!!!</p>',
unsafe_allow_html=True)
      else:
        st.warning("Pleaselogin/signuptousetheApp!!!")


if_name_=="_main_":
  initialize_database()
  main()
```

TWITTER CODE

```
import pandas as pd
import numpy as np
importseabornas sns
importmatplotlib.pyplotasplt
import seaborn as sns
fromsklearn.model_selectionimporttrain_test_split,GridSearchCV,cross_validate,
cross_val_score
from imblearn.over_sampling import SMOTE
fromsklearn.preprocessing import StandardScaler
fromsklearn.linear_modelimport LogisticRegression
from sklearn.svm import SVC
fromsklearn.ensembleimportBaggingClassifier,GradientBoostingClassifier,
AdaBoostClassifier
fromsklearn.treeimportDecisionTreeClassifier
fromsklearn.ensemble importRandomForestClassifier
from sklearn.ensemble import BaggingClassifier
fromsklearn.ensemble import StackingClassifier
from xgboost import XGBClassifier
fromsklearn.pipelineimportPipeline
fromsklearn.composeimportColumnTransformer
fromsklearn.metricsimportaccuracy_score,confusion_matrix,classification_report,
recall_score, precision_score
import pickledata = pd.read_csv('twitter_data.csv')
data['Fake Or Not Category'].value_counts()
Y_label=data['Fake Or Not Category']
```

```
smote=SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(data, Y_label)
X_train,X_test,y_train,y_test=train_test_split(X_resampled,y_resampled,test_size=0.2)
def train_models(grid_search, name):
    # Fit the GridSearchCV object
    grid_search.fit(X_train,y_train)

    #Makepredictions
    y_pred = grid_search.predict(X_test)
    accuracy=accuracy_score(y_test,y_pred)
    precision= precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')

    metrics={name:{'Accuracy':accuracy,'Precision':precision,'Recall':recall}}

    # Print classification report
    print("Classification Report:")
    print(classification_report(y_test, y_pred))

    #Plotconfusionmatrix
    cm=confusion_matrix(y_test,y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=grid_search.best_estimator_.named_steps['classifier'].classes_,
            yticklabels=grid_search.best_estimator_.named_steps['classifier'].classes_)
    plt.xlabel('Predictedlabels')
```

```
    plt.ylabel('True labels')
    plt.title('ConfusionMatrix')
    plt.show()

    returnmetrics, grid_search.best_estimator_
logistic_clf1 = Pipeline(steps=[
        ('scaler', StandardScaler()),
        ('classifier', LogisticRegression())
])

param_grid={
    'classifierC':[0.001,0.01,0.1,1,10,100],
    'classifierpenalty':['l2']
}

#CreateGridSearchCVobject
grid_search= GridSearchCV(estimator=logistic_clf1, param_grid=param_grid, cv=5,
n_jobs=-1)

logistic_metrics,logistic_model=train_models(grid_search,"LogisticRegression")
filename = 'logistc_model_twitter.pkl'
pickle.dump(logistic_model, open(filename,'wb'))
svc_clf = Pipeline([
    ('scaler', StandardScaler()),
    ('classifier', SVC())
])
```

```
#Definetheparametergrid
param_grid = {
    'classifierC': [0.1, 1, 10],
    'classifierkernel': ['linear', 'rbf'],
    'classifiergamma':['scale','auto']
}
#CreateGridSearchCVobject
grid_search=GridSearchCV(estimator=svc_clf,param_grid=param_grid,cv=5,n_jobs=-1)

#Trainthemodels
svc_metrics,svc_model=train_models(grid_search,"SVCModel")

#FittheGridSearchCVobjecttothedata grid_search.fit(X_train,
y_train)

#Getthebestparameters
best_params_SVM =grid_search.best_params_
filename = 'svc_model_twitter.pkl'
pickle.dump(grid_search.best_estimator_, open(filename, 'wb'))
decision_clf = Pipeline([
    ('scaler', StandardScaler()),
    ('classifier', DecisionTreeClassifier())
])
```

```
param_grid={
    'classifiercriterion':['gini','entropy'],
    'classifier__max_depth':[None,10,20,30,40,50],
    'classifier__min_samples_split':[2,5,10],
    'classifier__min_samples_leaf': [1, 2, 4],
    'classifier__max_features':['auto','sqrt','log2']}
#CreateGridSearchCVobject
grid_search=GridSearchCV(estimator=decision_clf,param_grid=param_grid,cv=5,
n_jobs=-1)

#Trainthemodels
decision_metrics,decision_model=train_models(grid_search,"DecisionTree")

#Getthebestparameters
best_params_decision_tree=grid_search.best_params_

#Savethebestmodel
filename =
'decision_tree_twitter.pkl'pickle.dump(grid_search.best_estimat
or_, open(filename, 'wb')) random_clf =Pipeline([
        ('scaler', StandardScaler()),
        ('classifier', RandomForestClassifier())
    ])


param_grid={
```

```
    'classifier__n_estimators':[100,200,300],
    'classifier__max_depth':[None,10,20,30,40,50],
    'classifier__min_samples_split':[2,5,10],
    'classifier__min_samples_leaf': [1, 2, 4],
    'classifier__max_features':['auto','sqrt','log2']
}

grid_search=GridSearchCV(estimator=random_clf,param_grid=param_grid,cv=5,
n_jobs=-1)

#Trainthemodels

randmon_metrics,random_model=train_models(grid_search,"RandomForest")

#Getthebestparameters
best_params=grid_search.best_params_

filename =
'random_model_twitter.pkl'pickle.dump(random_model,
open(filename,'wb'))
importances = random_model['classifier'].feature_importances_
importances
#FinalModelSelectedwith2features

random_model.fit(extracted_df, y_train)
```

```
pickle.dump(random_model,open(filename,'wb'))

y_pred=random_model.predict(test_tf)

accuracy=accuracy_score(y_test,y_pred)
precision= precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
#metrics['Random_Forest_1']={'Accuracy':accuracy, 'Precision':precision,'Recall':recall}

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))

#Plotconfusionmatrix
cm=confusion_matrix(y_test,y_pred) #
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=random_model.named_steps['classifier'].classes_,
            yticklabels=random_model.named_steps['classifier'].classes_)plt.xlabel('Pred
icted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()ada_clf=Pipeline([
        ('scaler',StandardScaler()),
        ('classifier',AdaBoostClassifier(n_estimators=500,random_state=42))
    ])
```

```
param_grid={
    'classifier__n_estimators':[50,100,200,500],
    'classifierlearning_rate': [0.01, 0.1, 1.0],
    'classifieralgorithm':['SAMME','SAMME.R']
}

grid_search=GridSearchCV(estimator=ada_clf,param_grid=param_grid,cv=5,n_jobs=-1) #
Train the models

ada_metrics,ada_model=train_models(grid_search,"AdaBoost")

#Getthebestparameters
best_params_ada=grid_search.best_params_

filename =
'adaboost_model_twitter.pkl'pickle.dump(ada_model,open(file
name,'wb'))
results=[logistic_metrics,svc_metrics,decision_metrics,randmon_metrics,ada_metrics]
model_names = [list(result.keys())[0] for result in results]

#Extractmetrics
metrics=[list(result.values())[0]forresultinresults]

#Extractmetricnames
metric_names=list(metrics[0].keys())
```

```
#Plotlinechartforeachmetric
plt.figure(figsize=(12, 6))
formetric_nameinmetric_names:
    metric_values = [metric[metric_name] for metric in metrics]
    plt.plot(model_names, metric_values, marker='o', label=metric_name.capitalize())

plt.xlabel('Models')
plt.ylabel('Values')
plt.title('PerformanceofModels')
plt.xticks(rotation=45)
plt.legend(title='Metrics')
plt.ylim(0,1)#Sety-axis limitfrom0to1
plt.grid(True)
plt.tight_layout()
plt.show()
```

FACEBOOK CODE

```
import pandas as pd
import numpy as np
importseabornas sns
importmatplotlib.pyplotasplt
import seaborn as sns
fromsklearn.model_selectionimporttrain_test_split,GridSearchCV,cross_validate,
cross_val_score
from imblearn.over_sampling import SMOTE
fromsklearn.preprocessing import StandardScaler
fromsklearn.linear_modelimportLogisticRegression
```

```
fromsklearn.svmimportSVC
fromsklearn.ensembleimportBaggingClassifier, GradientBoostingClassifier,
AdaBoostClassifier
fromsklearn.treeimportDecisionTreeClassifier
fromsklearn.ensemble importRandomForestClassifier
from sklearn.ensemble import BaggingClassifier
fromsklearn.ensemble importStackingClassifier
from xgboost import XGBClassifier
fromsklearn.pipelineimportPipeline
fromsklearn.composeimportColumnTransformer
fromsklearn.metricsimportaccuracy_score,confusion_matrix,classification_report,
recall_score, precision_score
importpickle
data= pd.read_csv('Facebook SpamDataset.csv') eda_df
= data
eda_df['Label'].replace([0,1], ['Legitimate', 'Fake'], inplace=True)
eda_df.drop('profile id', axis=1, inplace=True)
colors=['red','orange','yellow','green','blue','indigo','violet']
hists, ax = plt.subplots(7, 2, figsize=(20,30))
count=0
for row in range(7):
    forcolinrange(2):
        ifcount!=13:
            feature = sns.histplot(x=eda_df[eda_df.columns[count]], ax=ax[row,col], kde=True,
hue=eda_df['Label'])
            feature.set_xlabel(None)
```

```
            feature.set_ylabel(None)
            feature.set_title(eda_df.columns[count])

            for desc, color, label in zip(eda_df[eda_df.columns[count]].describe()[1:].values,
                                          colors,
                                          eda_df[eda_df.columns[count]].describe()[1:].index):
                feature.axvline(desc, color=color, label=label, linestyle='--', linewidth=1)
            feature.legend()

        else:
            feature =sns.countplot(x=eda_df['Label'])
feature.set_xlabel(None)
            feature.set_ylabel(None)
            feature.set_title(eda_df.columns[count])

        count+=1

hists.tight_layout(rect=[0,0.03,1,0.95])
hists.suptitle("Feature Comparison BetweenUserEngagements", fontsize=20,
fontweight='bold');
data['Label'].value_counts()
smote=SMOTE(random_state=42)
X_resampled,y_resampled= smote.fit_resample(data, Y_label)
def train_models(grid_search, name):
    #FittheGridSearchCVobject
    grid_search.fit(X_train,y_train)
```

```
#Makepredictions
y_pred = grid_search.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
precision= precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')

metrics={name:{'Accuracy':accuracy,'Precision':precision,'Recall':recall}}

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))

#Plotconfusionmatrix
cm=confusion_matrix(y_test,y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=grid_search.best_estimator_.named_steps['classifier'].classes_,
yticklabels=grid_search.best_estimator_.named_steps['classifier'].classes_)plt.xlabel('Predicte
d labels')
plt.ylabel('True labels')
plt.title('ConfusionMatrix')
plt.show()

returnmetrics, grid_search.best_estimator_
logistic_clf1 = Pipeline(steps=[
```

```
        ('scaler', StandardScaler()),
        ('classifier', LogisticRegression())
])

param_grid={
    'classifierC':[0.001,0.01,0.1,1,10,100],
    'classifierpenalty':['l2']
}

#CreateGridSearchCVobject
grid_search= GridSearchCV(estimator=logistic_clf1, param_grid=param_grid, cv=5,
n_jobs=-1)

logistic_metrics,logistic_model=train_models(grid_search,"LogisticRegression")
filename = 'logistc_model_facebook.pkl'
pickle.dump(logistic_model, open(filename,'wb'))
svc_clf = Pipeline([
    ('scaler',StandardScaler()),
    ('classifier', SVC())
])

#Definetheparametergrid
param_grid = {
    'classifierC': [0.1, 1, 10],
    'classifierkernel':['linear','rbf'],
    'classifiergamma':['scale','auto']
```

```
}
#CreateGridSearchCVobject
grid_search=GridSearchCV(estimator=svc_clf,param_grid=param_grid,cv=5,n_jobs=-1)

#Trainthemodels
svc_metrics,svc_model=train_models(grid_search,"SVCModel")

#FittheGridSearchCVobjecttothedata grid_search.fit(X_train,
y_train)

#Getthebestparameters
best_params_SVM=grid_search.best_params_

filename =
'svc_model_facebook.pkl'pickle.dump(grid_search.best_estimat
or_, open(filename, 'wb')) decision_clf = Pipeline([
   ('scaler', StandardScaler()),
   ('classifier', DecisionTreeClassifier())
])

#Definetheparametergrid
param_grid = {
   'classifiercriterion':['gini','entropy'],
   'classifier__max_depth':[None,10,20,30,40,50],
   'classifier__min_samples_split':[2,5,10],
```

```
    'classifier__max_features':['auto','sqrt', 'log2']
}

#CreateGridSearchCVobject
grid_search=GridSearchCV(estimator=decision_clf,param_grid=param_grid,cv=5,
n_jobs=-1)

#Trainthemodels
decision_metrics,decision_model=train_models(grid_search,"DecisionTree")

#Getthebestparameters
best_params_decision_tree=grid_search.best_params_

#Savethebestmodel
filename =
'decision_tree_facebook.pkl'pickle.dump(grid_search.best_esti
mator_, open(filename, 'wb')) random_clf =Pipeline([
        ('scaler', StandardScaler()),
        ('classifier', RandomForestClassifier())
    ])


param_grid={
    'classifier__n_estimators':[100,200,300],
    'classifier__max_depth':[None,10,20,30,40,50],
    'classifier__min_samples_split':[2,5,10],
```

```
    'classifier__min_samples_leaf': [1, 2, 4],
    'classifier__max_features':['auto','sqrt','log2']
}

grid_search=GridSearchCV(estimator=random_clf,param_grid=param_grid,cv=5,
n_jobs=-1)

#Trainthemodels

randmon_metrics,random_model=train_models(grid_search,"RandomForest")

#Getthebestparameters
best_params=grid_search.best_params_

filename =
'random_model_facebook.pkl'pickle.dump(random_m
odel, open(filename,'wb')) random_clf =Pipeline([
        ('scaler', StandardScaler()),
        ('classifier', RandomForestClassifier())
    ])


param_grid={
    'classifier__n_estimators':[100,200,300],
    'classifier__max_depth':[None,10,20,30,40,50],
    'classifier__min_samples_split':[2,5,10],
```

```
    'classifier__min_samples_leaf': [1, 2, 4],
    'classifier__max_features':['auto','sqrt','log2']
}

grid_search= GridSearchCV(estimator=random_clf, param_grid=param_grid, cv=5,
n_jobs=-1)

#Trainthemodels

randmon_metrics,random_model=train_models(grid_search,"RandomForest")

#Getthebestparameters
best_params=grid_search.best_params_

filename = 'random_model_facebook.pkl'
pickle.dump(random_model, open(filename, 'wb'))
selected_indices=[5,2,0,7]#Listofcolumnindicestoextract
extracted_df = X_train.iloc[:, selected_indices]
column_names=extracted_df.columns
test_tf=X_test.iloc[:,selected_indices]
column_names
#FinalModelSelectedwith4features
metrics = {}

random_model.fit(extracted_df, y_train)
```

```
filename =
'random_model1_facebook_final.pkl'pickle.dump(
random_model, open(filename,'wb')) # Make
predictions
y_pred = random_model.predict(test_tf)
accuracy=accuracy_score(y_test,y_pred)
precision= precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')

metrics['Random_Forest_1']={'Accuracy':accuracy,'Precision':precision,'Recall':recall}

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))

#Plotconfusionmatrix
cm=confusion_matrix(y_test, y_pred) #
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=random_model.named_steps['classifier'].classes_,
        yticklabels=random_model.named_steps['classifier'].classes_)plt.xlabel('Pred
icted labels')
plt.ylabel('True labels')
plt.title('ConfusionMatrix')
plt.show()
ada_clf=Pipeline([
```

```
        ('classifier',AdaBoostClassifier(n_estimators=500,random_state=42))
    ])

param_grid={
    'classifier__n_estimators':[50,100,200,500],
    'classifierlearning_rate': [0.01, 0.1, 1.0],
    'classifieralgorithm':['SAMME','SAMME.R']
}

grid_search=GridSearchCV(estimator=ada_clf,param_grid=param_grid,cv=5,n_jobs=-1)

#Trainthemodels

ada_metrics,ada_model=train_models(grid_search,"AdaBoost")

#Getthebestparameters
best_params_ada=grid_search.best_params_

filename = 'adaboost_model_facebook.pkl'
pickle.dump(ada_model, open(filename, 'wb'))
model_names=[list(result.keys())[0]forresultinresults]

#Extractmetrics
metrics=[list(result.values())[0]forresultinresults]
```

```
metric_names=list(metrics[0].keys())

#Plotlinechartforeachmetric
plt.figure(figsize=(12, 6))
formetric_nameinmetric_names:
    metric_values = [metric[metric_name] for metric in metrics]
    plt.plot(model_names, metric_values, marker='o', label=metric_name.capitalize())

plt.xlabel('Models')
plt.ylabel('Values')
plt.title('PerformanceofModels')
plt.xticks(rotation=45)
plt.legend(title='Metrics')
plt.ylim(0,1)#Sety-axis limitfrom0to1
plt.grid(True)
plt.tight_layout()
plt.show()
```

INSTAGRAM CODE

```
import pandas as pd
import numpy as np
importseabornas sns
importmatplotlib.pyplotasplt
import seaborn as sns
fromsklearn.model_selectionimporttrain_test_split,GridSearchCV,cross_validate,
cross_val_score
fromimblearn.over_samplingimportSMOTE
```

```
from sklearn.preprocessing import StandardScaler
fromsklearn.linear_modelimport LogisticRegression
from sklearn.svm import SVC
fromsklearn.ensembleimportBaggingClassifier, GradientBoostingClassifier,
AdaBoostClassifier
fromsklearn.treeimportDecisionTreeClassifier
fromsklearn.ensemble importRandomForestClassifier
from sklearn.ensemble import BaggingClassifier
fromsklearn.ensemble importStackingClassifier
from xgboost import XGBClassifier
fromsklearn.pipelineimportPipeline
fromsklearn.composeimportColumnTransformer
fromsklearn.metricsimportaccuracy_score,confusion_matrix,classification_report,
recall_score, precision_score
import pickle
y_label=data['fake']
data=data.drop('fake',axis=1)
sns.barplot(y_label.value_counts())
plt.show()plt.figure(figsize=(20,
20))
cm = data.corr()
ax=plt.subplot()
sns.heatmap(cm,annot=True,ax=ax) plt.show()
#DataModelingandPreprocessing
```

```
deftrain_models(pipeline,name):
    # Fit the pipeline
    metrics={}

    pipeline.fit(data,y_label)

    #Makepredictions
    y_pred = pipeline.predict(X_test)
    accuracy=accuracy_score(y_test,y_pred)
    precision= precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')

    metrics[name]={'Accuracy':accuracy,'Precision':precision,'Recall':recall}

    # Print classification report
    print("Classification Report:")
    print(classification_report(y_test, y_pred))

    #Plotconfusionmatrix
    cm=confusion_matrix(y_test,y_pred) #
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm,annot=True,fmt='d',cmap='Blues',
xticklabels=pipeline.named_steps['classifier'].classes_,
                yticklabels=pipeline.named_steps['classifier'].classes_)
    plt.xlabel('Predicted labels')
    plt.ylabel('Truelabels')
```

```
    plt.title('ConfusionMatrix')
    plt.show()

    returnmetrics,pipeline #
Logistic Regression
logistic_clf1 = Pipeline(steps=[
        ('scaler', StandardScaler()),
        ('classifier', LogisticRegression())
])



logistic_metrics,logistic_model=train_models(logistic_clf1,"LogisticRegression")
filename = 'logistc_model_insta.pkl'
pickle.dump(logistic_model, open(filename,'wb'))
svc_clf = Pipeline([
    ('scaler',StandardScaler()),
    ('classifier', SVC())
])


svc_metrics,svc_model=train_models(svc_clf,"svc_model")

filename1 =
'svc_model_insta.pkl'pickle.dump(svc_model,o
pen(filename1, 'wb')) decision_clf = Pipeline([
```

```
        ('scaler', StandardScaler()),
        ('classifier', DecisionTreeClassifier())
        ])

decision_metrics,decision_model=train_models(decision_clf,"decsion_tree")
filename2 = 'decision_tree_insta.pkl'
pickle.dump(decision_model, open(filename2, 'wb'))
random_clf =Pipeline([
        ('scaler',StandardScaler()),
        ('classifier',RandomForestClassifier(n_estimators=10000))
    ])



randmon_metrics,random_model=train_models(random_clf,"Random_Forest")
filename = 'random_model_insta.pkl'
pickle.dump(random_model,open(filename,'wb'))selected_indices=[9,8,1,0,5,10]#
List of column indices to extract
extracted_df=data.iloc[:, selected_indices]
column_names = extracted_df.columns
test_tf = X_test.iloc[:,selected_indices]
column_names
#FinalModelwith6features random_clf1
=Pipeline([
        ('scaler',StandardScaler()),
        ('classifier',RandomForestClassifier(n_estimators=1000))
    ])
```

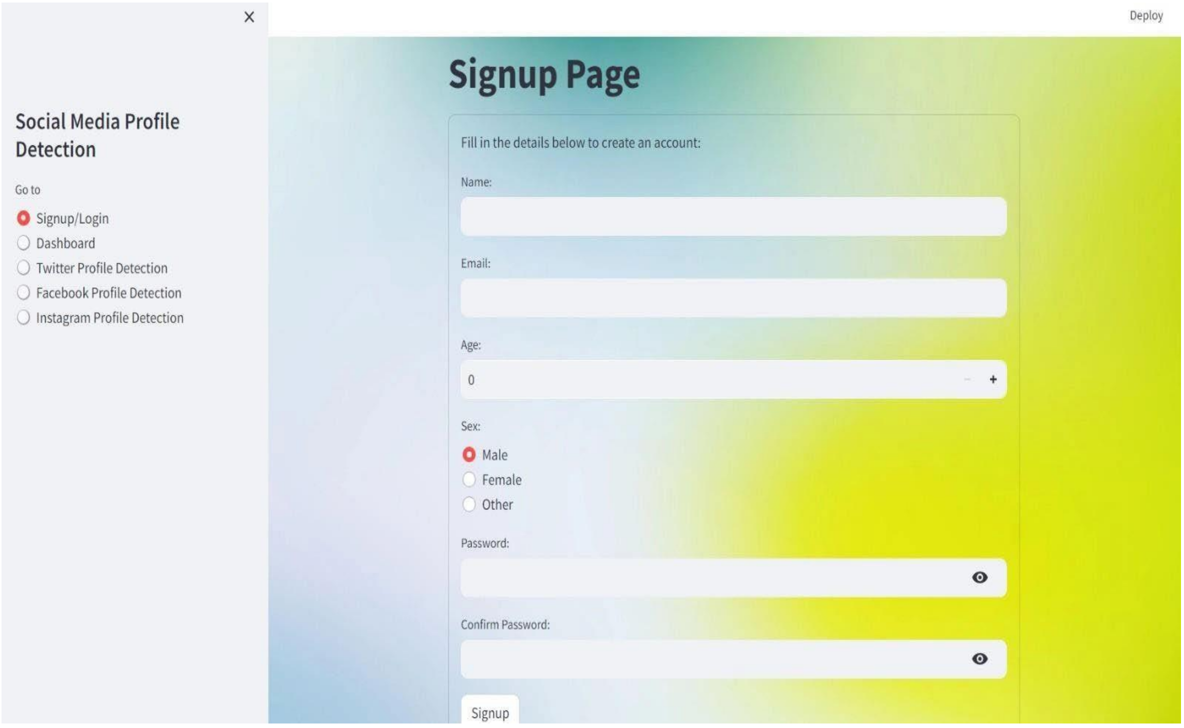
```
metrics={}

random_clf1.fit(extracted_df, y_label)

filename = 'random_model1_insta_final.pkl'pickle.dump(random_clf1, open(filename, 'wb'))
# Make predictions
y_pred = random_clf1.predict(test_tf)
accuracy=accuracy_score(y_test,y_pred)
precision= precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')

metrics['Random_Forest_1']={'Accuracy':accuracy,'Precision':precision,'Recall':recall}

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
# Plot confusion matrix
cm=confusion_matrix(y_test, y_pred) #
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=random_clf1.named_steps['classifier'].classes_,
            yticklabels=random_clf1.named_steps['classifier'].classes_)
plt.xlabel('Predicted labels')
plt.ylabel('Truelabels')
```

```
plt.title('Confusion Matrix')
plt.show()ada_clf=Pipeline([
        ('scaler',StandardScaler()),
        ('classifier',AdaBoostClassifier(n_estimators=500,random_state=42))
    ])
ada_clf_metrics,adaboost_model=train_models(ada_clf,"AdaBoost")
filename = 'adaboost_model_insta.pkl'
pickle.dump(adaboost_model, open(filename, 'wb'))gd_clf= Pipeline([
    ('scaler', StandardScaler()),
    ('classifier',GradientBoostingClassifier(n_estimators=100,   random_state=42))
])
gd_metrics,gd_boost=train_models(gd_clf,"Gradient_Boosting_Classifier")

filename='gd_boost_insta.pkl'
pickle.dump(gd_boost,open(filename,'wb'))xgb_clf=Pipeline([
    ('scaler', StandardScaler()),
    ('classifier',XGBClassifier(n_estimators=100,random_state=42))
])
xgb_metrics,xgb_model=train_models(xgb_clf, "XGBClassifier")

filename='xgb_boost_insta.pkl'
pickle.dump(xgb_model, open(filename, 'wb'))bagging_logistic_pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('classifier', BaggingClassifier(estimator=LogisticRegression(), n_estimators=1000,
random_state=42))
])
```

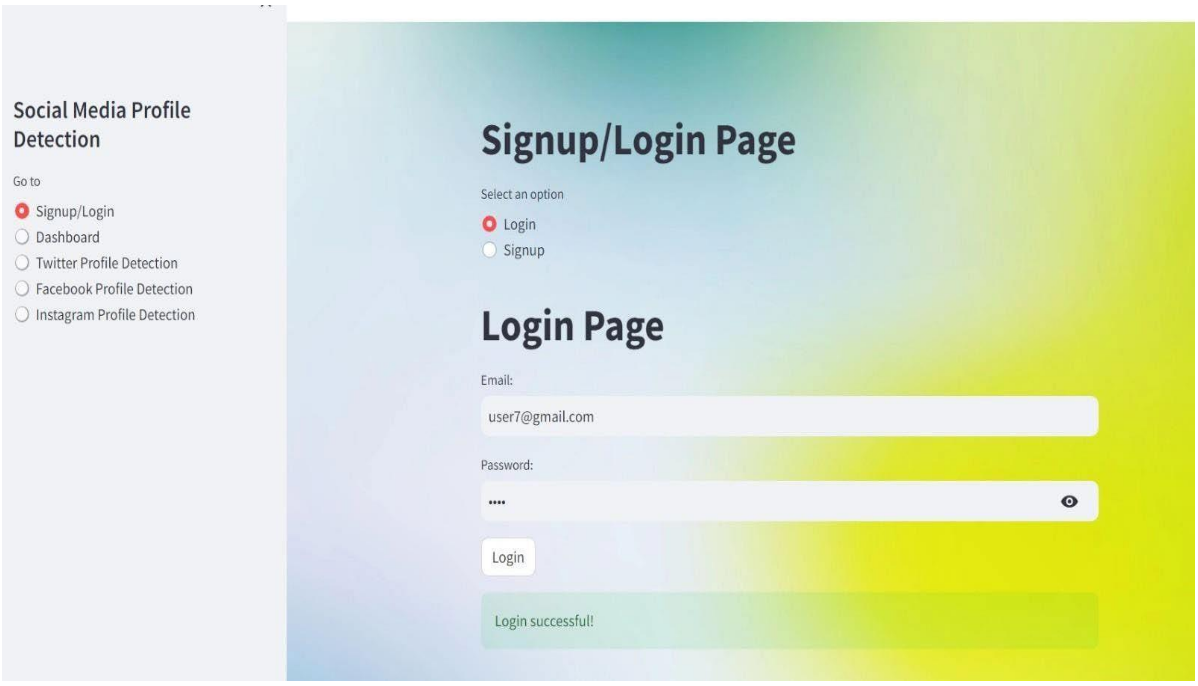
```
bagging_metric,bagging_model=train_models(bagging_logistic_pipeline,"BaggingUsing Logistic Regression")
filename='bagging_model_insta.pkl'
pickle.dump(bagging_model, open(filename, 'wb'))dt=
DecisionTreeClassifier(random_state=42)

rf = RandomForestClassifier(random_state=42)
lr = LogisticRegression(random_state=42)



stacking_clf1=StackingClassifier(
    estimators=[('dt', dt), ('rf', rf)],
    final_estimator=lr
)stacking_2 = Pipeline([
    ('scaler',StandardScaler()),
    ('classifier',stacking_clf1)
])

stacking2_metrics,stack_model2=train_models(stacking_2,"StackingAlgo2") filename =
'stack_model1_insta.pkl'
pickle.dump(stack_model2 , open(filename, 'wb'))results= [logistic_metrics , svc_metrics ,
decision_metrics,randmon_metrics,bagging_metric,stacking2_metrics,ada_clf_metrics,
gd_metrics, xgb_metrics]
model_names=[list(result.keys())[0]forresultinresults]
```

```
#Extractmetrics
metrics=[list(result.values())[0]forresultinresults]

#Extractmetricnames
metric_names=list(metrics[0].keys())

#Plotlinechartforeachmetric
plt.figure(figsize=(12, 6))
formetric_nameinmetric_names:
    metric_values = [metric[metric_name] for metric in metrics]
    plt.plot(model_names, metric_values, marker='o', label=metric_name.capitalize())

plt.xlabel('Models')
plt.ylabel('Values')
plt.title('PerformanceofModels')
plt.xticks(rotation=45)
plt.legend(title='Metrics')
plt.ylim(0,1)#Sety-axis limitfrom0to1
plt.grid(True)
plt.tight_layout()
plt.show()
```

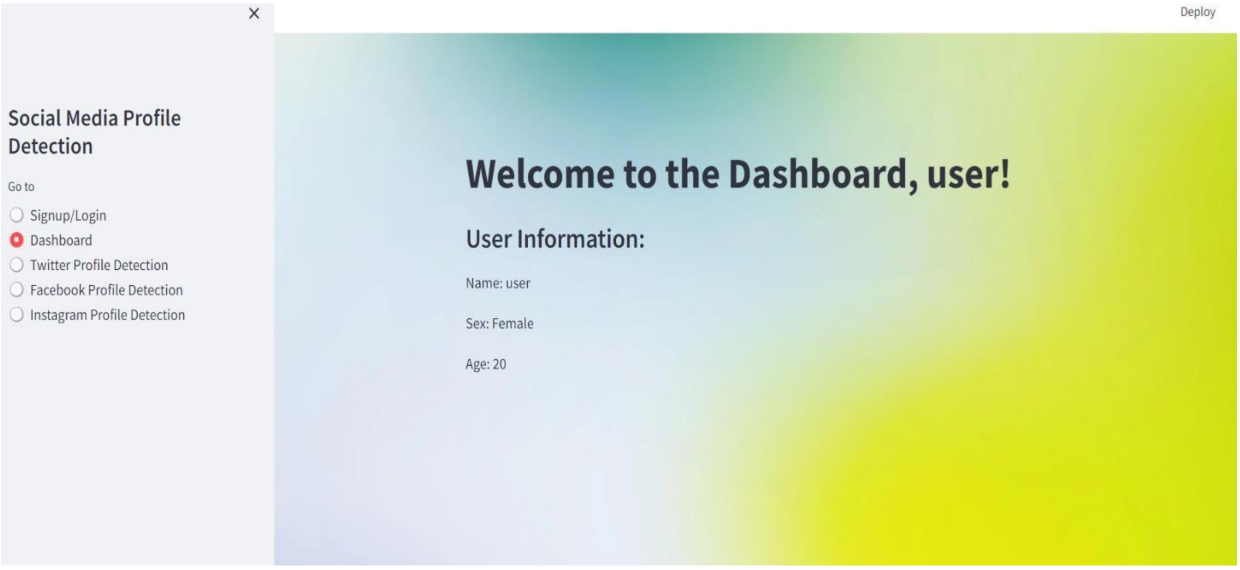
A.4 SCREENSHOT



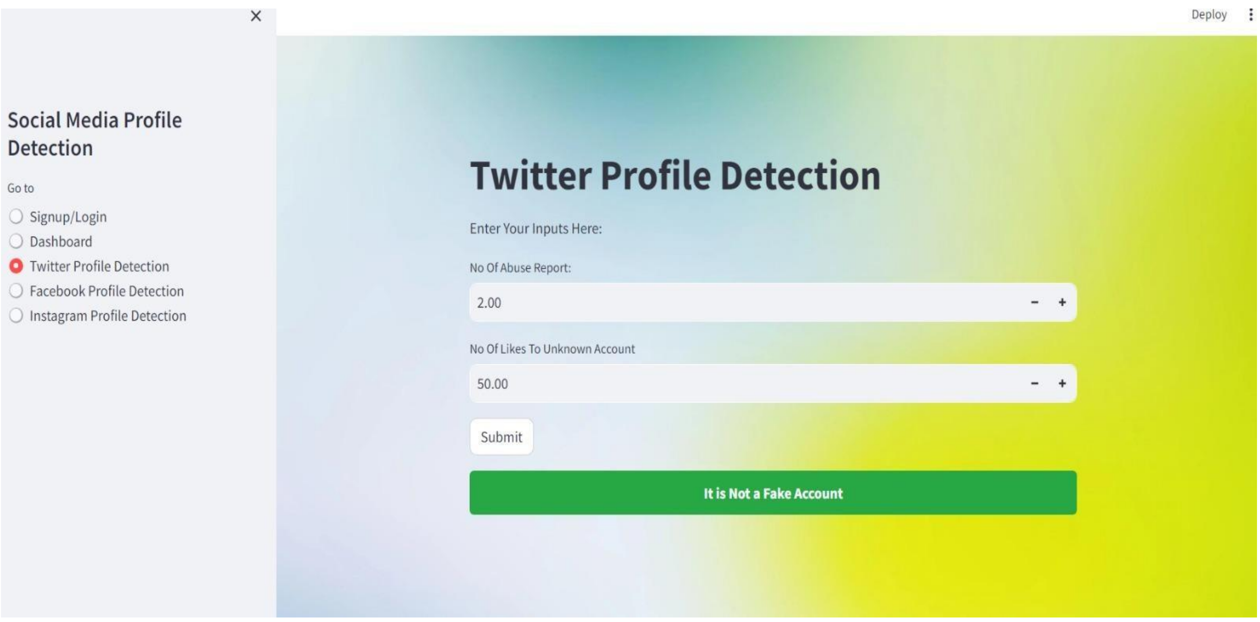
A.3.1Signup page



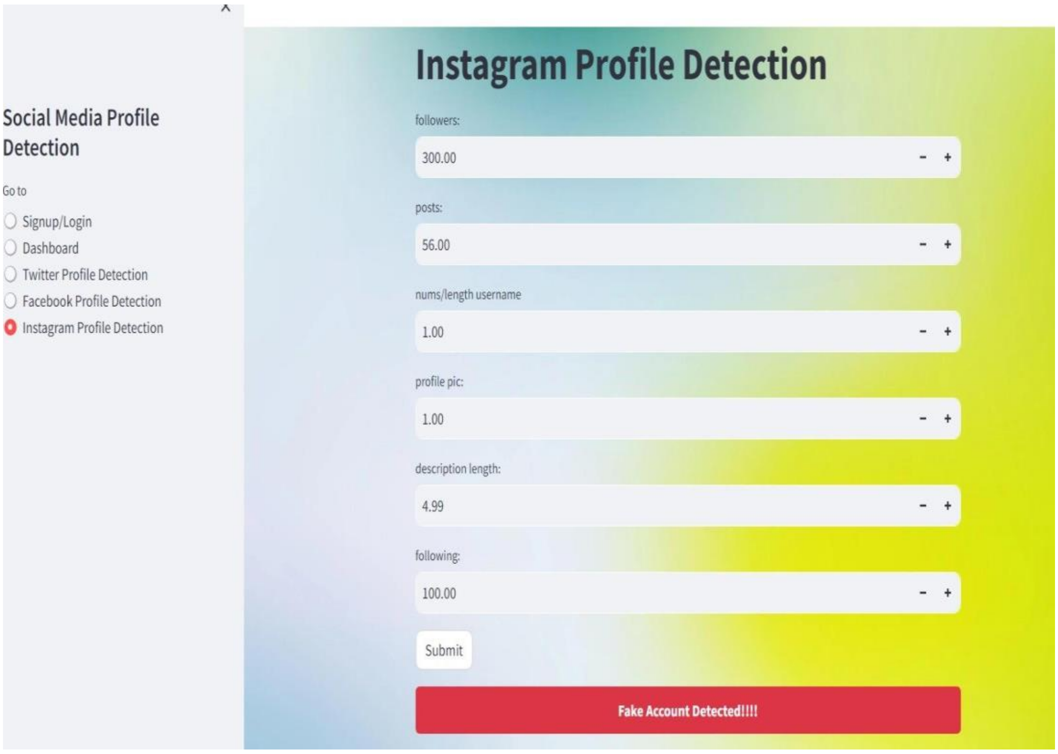
A.3.2 Login page



A.3.3 Dashboard



A.3.4 Twitter



A.3.5 Instagram

A.4 PLAG REPORT

FAKE PROFILE IDENTIFICATION IN SOCIAL NETWORK

Dr.Kavitha Subramani M.E.,PH.D

Professor,

Supervisor,

Panimalar engineering college,

Chennai,TamilNadu

Harshavardhini A.C

Department of Computer Science

Panimalar engineering college,

Chennai,TamilNadu

Kanmanishree M

Department of Computer Science

Panimalar engineering college,

Chennai,TamilNadu

Abstract

The rapid advancement of technology, particularly in mobile devices, has led to a significant increase in online social networking activity. However, this surge in connectivity has also brought about challenges, such as the proliferation of fake profiles and online impersonation. Research indicates that a substantial percentage of profiles on platforms like Facebook are fake, posing serious threats to user trust and security. To address this issue, various frameworks and machine learning models, including neural networks and random forests, have been employed to detect fake accounts based on engineered features. These models have shown promising results, with neural networks achieving accuracy rates of up to 93%. Looking ahead, there is optimism that incorporating new features, such as skin detection using natural language processing techniques, could further enhance the accuracy of fake profile detection. Moreover, future updates to social networking platforms like Facebook are expected to streamline the identification process for fake accounts, bolstering user confidence and safety.

Keywords: Social Networks, Fake Profiles, Machine Learning, Neural Networks, Natural Language Processing.

I. Introduction

Artificial neural networks **form** the **backbone** of **to day's** deep learning models, but they can also inclu de **hierarchically equivalent expressions** or **varia tions** in deep models, such **as** deep belief **nodes** an d deep Boltzmann **machine.**Every stage of deep learning is skilled at converting raw data into increasingly composite and abstract representations.

Consider the following scenario for photo recognition: A first layer could receive a matrix of pixels as input; it could then abstract the pixels and encode the edges; it could then combine and encode the edge arrangements; it could then recognize a nose and eyes; finally, it could detect a face in the image. It's interesting to note that a deep learning process is even capable of determining for itself which traits belong in which level. (However, manual tweaking is still important; changing the quantity and size of layers, for example, might affect the level of abstraction.)

The word "depth" in "deep learning" refers to the te chnique used to transform data. Systems with deep learning, in particular, have significant Credit Alloc ation Path (CAP) depth. CAP represents the sequen ce of transitions from input to output and indicates potential cause-
effect relationships between them. Since the output layer also has a scale, the CAP depth in the output neural network is equal to the network depth of all hidden layers plus one. As a rule, the CAP depth in a recurrent neural network is infinite, where signals will pass through the layer many times. Although t here is no consensus on the distinction between sha llow learning and deep learning, most researchers a gree that deep learning should have a CAP depth gr eater than 2. It turns out that CAP depth 2 is a univ ersal approximation that can be tested for any funct ion. More layers then do not improve the network's ability to predict activity. Since deep models (CAP > 2) are better at extracting features than shallow m odels, adding more layers can make learning easier.

information pertinent to the system's construction and operation.

Architecture Diagram

A system's architectu5 often encapsulated in an architecture diagram, is the conceptual model that outlines the system's behavior, structure, and different points of view. This architectural representation offers a codified description that makes it easier to reason about the actions and structures of the system. Fundamentally, a system architecture may include the elements of the system, their characteristics that can be observed from the outside, and the connections or actions that control how they interact.

Architecture description languages (ADLs) are the result of efforts to codify languages for specifying system architecture. By offering standardized frameworks for expressing the complexities of system architectures, these languages hope to promote consistency and clarity in design efforts.

Diverse Perspectives on Systems Architecture

Different organizations espouse varying interpretations of systems architecture, each tailored to their unique contexts and requirements. Some of these perspectives include:

- Allocated Arrangement of Physical Elements: According to this viewpoint, architecture is the methodical placement of physical elements that results in a design solution for end-user goods or life-cycle operations. It aims to ensure alignment with overarching objectives by harmonizing with the functional architecture and requirements baseline.

- Strategic Inventions and Decisions: Architecture is thought to consist of central, ubiquitous, high-level strategic inventions and decisions about the general framework, key components, connections, and related traits and actions. This viewpoint highlights the strategic considerations that inform architectural decisions and the reasoning behind them.

- Planning and Documentation

System architecture documentation can include a thorough list of all the hardware, software, and networking features that are currently in use. In addition, it may outline long-term goals and priorities for upcoming purchases, improvements, or replacements; this would act as a guide for allocating resources and making strategic decisions.

In essence, system design and architecture play pivotal roles in translating requirements into tangible solutions, fostering clarity, coherence, and alignment throughout the development lifecycle.

IV. Module Description

Login

This module facilitates access for authorized personnel, requiring the input of a valid username and corresponding password. It serves as a gatekeeper, ensuring that only authenticated users can modify data within the system. The login mechanism enhances security and enables administrative functions such as data updates and modifications.

User Profile Dataset

A user profile is a graphical representation of perso nal information associated with a specific user or p ersonal desktop environment.It encompasses digital representations of an individual's identity, including display settings, application configurations, and network connections. Admin profiles are akin to computerized user models, dictating the user's interface and access privileges based on administrative configurations.

Fake Profile Identification

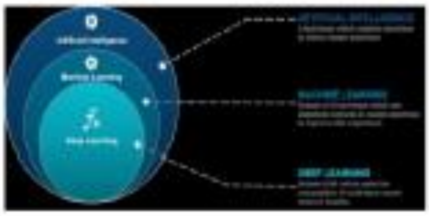
In the realm of social networks, fake profiles pose a persistent challenge, created for various nefarious purposes by individuals or 3oups. Fake profile detection requires the use of machine learning models, such as random forests and neural networks, and manufactured features. Through rigorous analysis, genuine accounts are distinguished from fraudulent ones, safeguarding the integrity of online interactions.

II. Literature Review

In today's digital world, where technology is advancing at an exponential rate, cellphones are a common sight, which highlights the pervasiveness of technology. Particularly, the realm of online social networks has woven itself into the fabric of everyday life, facilitating the forging of new connections and the maintenance of existing relationships. However, amidst this proliferation of online networking, the insidious phenomenon of fake profiles and online impersonation has emerged as a pressing concern.
Research conducted by Joshi et al. (2020) sheds light on the escalating prevalence of fake profiles within online social networks, with studies indicating that a significant proportion—ranging from 20% to 40%—of profiles on platforms such as Mark Zuckerberg Facebook are falsified. Such Unreal profiles not only inundate users with extraneous information but also engender trust issues and undermine the integrity of online interactions. Consequently, the imperative to develop robust frameworks for the detection and mitigation of fake profiles becomes paramount.

A notable approach to addressing this challenge involves leveraging one of Deep learning models, such as neural networks and random decision forests, to discern the authenticity of user accounts. Joshi et al. (2020) report promising results, with their neural network-based algorithm achieving an impressive accuracy rate of 93% in identifying fake profiles. Looking ahead, the integration of novel features—potentially incorporating techniques like skin detection facilitated by natural language processing—holds promise for enhancing detection efficacy further. Moreover, as social media platforms evolve and introduce new features, opportunities for more sophisticated identification of fake accounts are anticipated to emerge.

In a similar vein, P.L. Traskin (2017) advocates for the utilization of Deep learning techniques, particularly Ada Boost and Unsupervised concept Support Vector Machine (SVM), to combat the proliferation of fake accounts. Traskin's model, leveraging SVM as a classification technique, demonstrates robustness in discerning between fake and genuine profiles, boasting an accuracy rate exceeding 90%. This underscores the efficacy of SVM in handling classification tasks within large datasets, thereby obviating the need for manual evaluation of individual accounts.

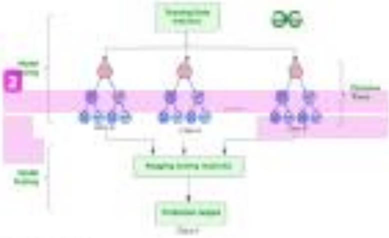


The prevalence of fake profiles underscores the imperative for proactive measures to safeguard the integrity of online social networks. By harnessing the capabilities of machine learning and leveraging publicly available data, researchers and organizations can equip themselves with effective tools for identifying and mitigating the proliferation of fake accounts, thereby fostering a more secure and trustworthy online ecosystem.

III. Proposed Methodology

Random Decision Forest of Unsupervised Learning

The Random Forest algorithm is apowerful machine learning tree learning technique. It creates a large number of Decision Trees during the training phase. Each tree is constructed using a random subset of t he dataset to evaluate a random subset of features i n each distribution. As a result of randomization, ea ch tree has more variance, so the likelihood of over fitting is reduced and the overall prediction is more efficient. When making predictions, the algorithm e ither averages (for regression tasks) or votes (for cl assification tasks) the output of each tree. The resul ts of collaborative decision-making with the help of information from multiple trees are consistent and accurate.



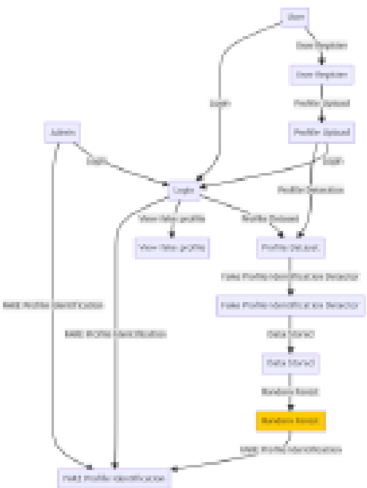
Advantages

- The first step in categorizing is to select th e profiles to be categorized. Once the profi le is selected, the main features are extract ed for classification.
- The trained classifier receives the extracted characteristics after that.

• As new data is introduced into the classifier, it is routinely trained.

The classifier then decides if the profile is real or not.
• After the classification algorithm's result is confirmed, the classifier receives feedback.
• The classifier gets increasingly good at predicting the phony profiles as the amount of training data rises.

The workflow of the model is delineated through an activity diagram, which elucidates the sequential and concurrent actions within the system. Activity diagrams serve as a visual representation of the dynamic behavior of the system, akin to flowcharts or data flow diagrams. They encapsulate the progression of activities from one to another, providing a comprehensive overview of the system's operation.



Hardware and Software Requirements

The requisite hardware and software specifications for deploying the proposed system entail setting up a Java-based application with Apache Tomcat as the server and SQL Server as the database. These components form the foundational infrastructure necessary for the seamless execution of the profile detection system, ensuring compatibility and efficiency in operation.

7
Hardware Requirements:

System: Pentium IV 2.4 GHz - This specifies the minimum processor requirement, indicating that the system should have at least a Pentium IV processor running at 2.4 GHz.
Hard Disk: 40 GB - This specifies the minimum hard disk space required for installing the operating system, software, and storing data.
Monitor: 15" VGA Colour - This specifies the minimum monitor size and resolution required for displaying the graphical interface of the system.
Mouse: Logitech - This specifies a brand preference for the mouse, suggesting the use of a Logitech mouse.
RAM: 512 MB - This specifies the minimum RAM (memory) requirement for the system, indicating that it should have at least 512 MB of RAM for smooth operation.

Software Requirements

Operating System: Windows XP/7 - This specifies the compatible operating systems for running the application, indicating that it can run on either Windows XP or Windows 7.
Coding Language: Java - This specifies the programming language used for developing the application, indicating that it is written in Java.
Tool: Apache Tomcat - This specifies the application server required for deploying and running Java-based web applications. Apache Tomcat is a popular choice for deploying Java servlets and JSPs.
Database: SQL Server - This specifies the relational database management system (RDBMS) used for storing and managing the application's data. SQL Server is a Microsoft product commonly used for such purposes.
In summary, the hardware and software requirements provide guidance on the necessary components and specifications for setting up a system environment suitable for running a Java-based application with Apache Tomcat as the server and SQL Server as the database backend.

System Design

System design encompasses the meticulous process of delineating the architecture, components, modules, and data structures of a system, tailored to meet specified requirements. It can be perceived as the application of system theory to the development of products or solutions. The culmination of system design efforts is typically documented in a comprehensive system design document, which serves as a repository of detailed data and

VI-Conclusion and Future Enhancement

The exploration of fake profile dynamics and their implications for online social networks has yielded valuable insights. By conducting social engineering experiments and analyzing user behaviors, patterns, and privacy considerations, we have deepened our understanding of the challenges posed by fake profiles. Notably, profiles lacking social activities and displaying anomalous friend counts are more likely to be flagged as fake. Additionally, we have proposed countermeasures, including user awareness training, to mitigate the proliferation of fake profiles and safeguard user privacy.

VII. Future Enhancement

The sustainability of the business model is significantly impacted by fake profiles, as are advertising firms that depend on the veracity of user data. They do, however, also significantly impact user privacy. According to our findings, most users are ignorant of the existence of phony profiles and the repercussions they can have. We explore countermeasures to raise user awareness in this section. The outcomes of our social science and technology study could also have ramifications for companies operating on social media platforms or for social network users. Making countermeasures for these parties, however, is outside the purview of this project. We speculate that everyone who is interested in the topic—students, parents, and teachers—want to talk about and learn more about privacy-related concerns on social media. The focus group participants were driven and enthusiastically participated.

REFERENCE

[1] Identifying Fake Profile in Online Social Network: An Overview and

SurveyShruti Joshi Himanshi Gupta Nagariyo Neha Dhanotiy2020 8th

International Conference on Reliability, Infocom Technologies and

Optimization (Trends and Future Directions) (ICRITO)

[2] Fake Profile Identification using Machine LearningP.L., Traskin, M.:

AdaBoost is consistent. Journal of Machine Learning Research 8, 2347–

2368 (2017)

[3] Fake Profile Identification in Online Social NetworksSk.Shama,

K.SivaNandini, P.Bhavya Anjali, K. Devi Manaswi International

Journal of Recent Technology and Engineering (IJRTE)ISSN: 2277-3878,

Volume-8 Issue-4,November 2019

[4] Detecting Fake Accounts on Social Media Sarah Khaled Hoda M. O.

Mokhtar Neamat El-TaziAll content following this page was uploaded

by Neamat El-Tazi on 24 May 2019

[5] Prediction of Fake Profiles SuheelYousufWani ,Mudasir M Kirmani, Syed

ImamulAnsarullaSuheelYousufWani et al, / (IJCSIT) International Journal

of Computer Science and Information Technologies, Vol. 7 (4) , 2016, 1735-

1738

[6] NarsimhaGugulothu,JayadevGyani, Srinivas Rao Pulluri "A Comprehensive

Model for Detecting Fake Profiles in Online Social Networks(2016)".

[7] Dr.Narsimha.G, Dr.JayadevGyani, P. Srinivas Rao ,"Fake Profiles

Identification in Online Social Networks Using Machine Learning

and NLP(2018)", International Journal of Applied Engineering Research

ISSN 0973-4562, Number 6, Volume 13.

[8] Reddy, A. V. N., &Phanikrishna, C. Contour tracking based knowledge

extraction and object recognition using deep learning neural

networks(2016). Paper presented at the Proceedings on 2nd International

Conference on Next Generation Computing Technologiesin

2016, NGCT 2016, 352-354. doi:10.1109/NGCT.2016.7877440.

[9] V. Rama Krishna,&K.Kanaka Durga. Automatic detection of illegitimate

websites with mutual clustering.(2016) International Journal of

Electrical and Computer Engineering, 6(3), 995-1001.

doi:10.11591/ijece.v6i3.9878

[10] D.Rajeswara Rao &V.Pellakuri. Training and development of artificial

neural network models: Single layer feedforward and multi layer

feedforward neural network(2016). Journal of Theoretical and Applied

Information Technology, 150-156,84(2).

FP2 reoort

ORIGINALITY REPORT

3%	1%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Sri Sathya Sai Institute of Higher Learning Student Paper	1%
2	www.ijraset.com Internet Source	<1%
3	Felix Kallenborn, Andreas Hildebrandt, Bertil Schmidt. "CARE: context-aware sequencing read error correction", Bioinformatics, 2021 Publication	<1%
4	dokumen.pub Internet Source	<1%
5	"Advances in Electronics, Communication and Computing", Springer Science and Business Media LLC, 2021 Publication	<1%
6	S.R. Ramya, R. Priyanka, S. Siva Priya, M. Srinivashini, A. Yasodha. "SVM Based Fake Account Sign-In Detection", 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), 2023 Publication	<1%

7	docplayer.net Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

REFERENCES

[1] Yang, S., Shu, K., Wang, S., Gu, R., Wu, F., & Liu, H. (2019). "Unsupervised fake news detectiononsocialmedia:Agenerativeapproach."InProceedingsoftheAAAIconference on artificial intelligence (Vol. 33, No. 01, pp. 5644-5651).

[2] Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). "Fake news detectiononsocialmediausinggeometricdeeplearning."arXivpreprintarXiv:1902.06673.

[3] Pulido,C.M.,Ruiz-Eugenio,L.,Redondo-Sama,G.,&Villarejo-Carballido,B.(2020)."A new application of social impact in social media for overcoming fake news in health." International journal of environmental research and public health, 17(7), 2430.

[4] Zhou,X.,&Zafarani,R.(2019)."Network-basedfakenewsdetection:Apattern-driven approach." ACM SIGKDD explorations newsletter, 21(2), 48-60.

[5] Shu, K., Bernard, H. R., & Liu, H. (2019). "Studying fake news via network analysis: detectionandmitigation."Emergingresearchchallengesandopportunitiesincomputational social network analysis and mining, 43-65.

[6] Benabbou, F., Boukhouima, H., &Sael, N.(2022). "Fake accounts detectionsystembased onbidirectionalgatedrecurrentunitneuralnetwork."InternationalJournalofElectricaland Computer Engineering (IJECE), 12(3), 3129.

[7] Shu,K.,Zhou,X.,Wang,S.,Zafarani,R.,&Liu,H.(2019). "Theroleofuserprofiles for fake news detection."InProceedings ofthe 2019IEEE/ACM internationalconferenceon advances in social networks analysis and mining (pp. 436-439).

[8] Liu,Y.,&Wu,Y.F.B.(2020)."FNED:a deepnetwork forfakenewsearlydetectionon social media." ACM Transactions on Information Systems (TOIS), 38(3), 1-33.

[9] Sahoo,S.R.,&Gupta,B.B.(2021)."Multiplefeaturesbasedapproachforautomaticfake news detection on social networks using deep learning." Applied Soft Computing, 100, 106983.

[10] Can,U.,&Alatas,B.(2019)."Anewdirectioninsocialnetworkanalysis:Onlinesocial network analysis problems and applications." Physica A: Statistical Mechanics and its Applications, 535, 122372.

[11] , X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization,detection,anddiscussion. *InformationProcessing&Management*, *57*(2), 102025.

[12] Talwar,S.,Dhir, A.,Singh,D.,Virk,G.S.,&Salo,J.(2020).Sharingoffakenews on social media: Application of the honeycomb framework and the third-person effect hypothesis. Zhang *Journal of Retailing and Consumer Services*, *57*, 102197.

[13] Carlson,M.(2020). Fakenewsasaninformationalmoralpanic:Thesymbolic deviancy of social media during the 2016 US presidential election. *Information, Communication & Society*, *23*(3), 374-388.

[14] Islam,M.R.,Liu,S.,Wang,X.,&Xu,G.(2020). Deeplearningformisinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining*, *10*(1), 82.

[15] DiDomenico,G.,Sit,J.,Ishizaka,A.,&Nunan,D.(2021).Fakenews,socialmedia and marketing: A systematic review. *Journal of Business Research*, *124*, 329-341.

[16]Himelein-Wachowiak,M.,Giorgi,S.,Devoto,A.,Rahman,M.,Ungar,L.,Schwartz, H. A., ... &Curtis, B. (2021). Bots and misinformationspread onsocial media: Implications for COVID-19. *Journal of medical Internet research*, *23*(5), e26933.

[17]Meel,P.,&Vishwakarma,D.K.(2020).Fakenews,rumor,informationpollutionin social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, *153*, 112986.

[18] Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., & Liu, Y. (2019). Combatingfake news:Asurveyonidentificationand mitigationtechniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(3), 1-42.

[19] Shu,K.,Mahudeswaran,D.,Wang,S.,Lee,D.,&Liu,H.(2020). Fakenewsnet:A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data*, *8*(3), 171-188.

[20] Zhang,J.,Dong,B.,&Philip,S.Y.(2020,April).Fakedetector:Effectivefakenews detection with deep diffusive neural network. In *2020 IEEE 36th international conference on data engineering (ICDE)* (pp. 1826-1829). IEEE.