**FAKE PROFILE DETECTION USING ML AND NLP**

A PROJECT REPORT

*Submitted by*

**NAMRATHA M [RA2111003010240]**

**KEERTHANA K [RA2111003010250]**

*Under the Guidance of*

**Dr. RAMKUMAR JAYARAMAN**

Associate Professor, Department of Computing Technologies

*in partial fulfillment of the requirements* *for the degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE ENGINEERING**



**DEPARTMENT OF COMPUTING TECHNOLOGIES**

**COLLEGE OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR- 603 203**

**NOVEMBER 2024**

**Department of Computing technologies**

**SRM Institute of Science & Technology**

**Own Work Declaration Form**

|  |  |
| --- | --- |
|  |  |
| **Degree/ Course**    **Student Name** | **: B. TECH / COMPUTER SCIENCE AND ENGINEERING**  **: NAMRATHA M & KEERTHANA K** |

**Registration Number : RA2111003010240 & RA2111003010250**

## Title of Work : FAKE PROFILE DETECTION USING ML AND NLP

We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is my / our own except where indicated, and that We have met the following conditions:

* Clearly referenced / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

We understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| We are aware of and understand the University’s policy on Academic misconduct and plagiarism and we certify that this assessment is our own work, except where indicated by referring, and that we have followed the good academic practices noted above.  Date: |
| M. Namratha K. Keerthana  RA2111003010240 RA2111003010250 |



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203**

**BONAFIDE CERTIFICATE**

Certified that 18CSP107L - Minor Project report titled “**FAKE PROFILE DETECTION USING ML AND NLP”** is the bonafide work of “**NAMARTHA M [RA2111003010240], KEERTHANA K [RA2111003010250]”** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE SIGNATURE**

**Dr. RAMKUMAR JAYARAMAN Dr. G NIRANJANA**

Associate Professor  **PROFESSOR & HEAD**

Department of Computing Department of Computing

Technologies Technologies

# ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. T. V. Gopal,** Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Com puting, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson, School of Computing and **Dr. C.Lakshmi,** Professor and Associate Chairperson, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department, **Dr. G. Niranjana**, Professor, Department of Computational Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, **Dr. S. Jagadeesan** and Panel Members,Dr. Ramkumar Jayaraman and Dr. A. Manju, Department of Computational Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor**,Dr. M. Eliazer**, Department of Computational Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. Ramkumar Jayaraman**, Department of Computational Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his / her mentorship. He / She provided us with the freedom and support to explore the research topics of our interest. His / Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computational Technologies, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement

**NAMRATHA[RA2111003010240]**

**KEERTHANA[RA2111003010250]**

# ABSTRACT

The rise of social media and online platforms has also spawned spammers and fake profiles to a large extent. This also calls for high-level security and trust in the online world. The new methodology for detecting fake profiles will be presented in this paper through ML and NLP. It will propose a multifaceted framework of both user metadata and content with text for the study. In the first stage, we extract features from user profiles -such as age of the account and follower-tofollowing ratios. Simultaneously, using NLP techniques, we analyze the linguistic patterns in the user-generated content including posts and comments in terms of sentiment analysis, coherence, and authenticity markers. We train a prediction model using machine learning classifiers, from the simple Random Forest and Support Vector Machines to the deep learning methods, in order to identify a true profile from false ones. The results of the experiments can be compared: the accuracy of the system is much higher than for traditional detection systems; our F1 score exceeds 90%. This framework enhances the reliability of online platforms and gives insight into the nature of fake profiles. The results prove that integrating ML and NLP can enhance digital ecosystems against fraudulent practices. Specifically, because more individuals have opened their accounts in various social media and online platforms, fraudulent profiles have increased dramatically as well, which will impair trust and security in interactions. This paper introduces a novel approach toward the identification of spurious profiles using ML and NLP techniques. We put forward a multi-faceted framework that analyzes user metadata alongside textual content. The first part is featuring extraction from user profiles, including account age, follower-to-following ratios, and engagement metrics. Meanwhile, we apply NLP to analyze the linguistic patterns in user-generated content such as posts and comments, in terms of sentiment analysis, coherence, and authenticity markers.

# TABLE OF CONTENTS

# ABSTRACT v

# LIST OF FIGURES vii

# LIST OF ABBREVIATIONS viii

# 1.INTRODUCTION

# 1.1 Introduction to Online Social media platforms and the Mishaps 1

# 1.2 Motivation 3

# 1.3 Sustainable Goal of Project 4

# 1.4 Vision of the project 4

# 2.LITERATURE SURVEY 8

# 3.SYSTEM ANALYSIS

# 3.1 Existing System 11

# 3.2 Proposed System 13

# 3.3 Block diagram 14

# 4. METHODOLOGY

# 4.1 General Architecture 17

# 4.2 Flow Chart 18

# 4.3 Explanation of flowchart 19

# 4.4 Algorithms 21

# 5. IMPLEMENTATION AND TESTING 26

# 6.RESULTS AND DISCUSSIONS 28

**7.CONCLUSION** 29

**REFERENCES** 31

**APPENDIX**

**A CODE** 33

**LIST OF FIGURES**

**Figure No TITLE PAGE NO**

**1 Block Diagram 14**

**2 Flow Chart 18**

**3 Confusion Matrix for 21**

**Model Performance**

# LIST OF ABBREVIATIONS

ML MACHINE LEARNING

DFD DATA FLOW DIAGRAM

DT DECISION TREE

OSN ONLINE SOCIAL NETWORK

GUI GRAPHICAL USER INTERFACE

**CHAPTER 1**

# INTRODUCTION

**1.1 Introduction to Online social media platforms and the Mishaps:**

Social networking site: It is a website in which each user has a profile and keeps in contact with friends, shares their updates, meets new people who have the same interests. These OSN are connected through web2.0 technology, and through this, the users can interact. Social networking sites are booming up with speed and transforming the way in which the people keep in contact with each other. What they do is connect people that share the same interest thus making it easy for their users to get friends online. More so, these days, with ML being provided as a rather sophisticated tool, it exploits algorithms used for making distinctions between spurious profiles and real ones with surprisingly high accuracy. Successful recognition relies heavily on feature extraction-that is, transforming complicated data into manageable features. In general, some common attributes are user activity metrics, text consistency, and behavior in the network. It also involves sentiment analysis, whereby the post's emotional tone is measured; fake profile owners tend to rely on rather too platitudinous or enthusiastic language patterns for which the tools can identify and measure traces.

## Dataset

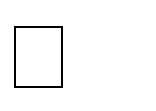
Data is a collection. Most of the time, using machine learning, we need large data for different purposes. The training dataset that is fed into our machine learning algorithms to train the model. T ese dataset is a sample of the dataset that is used in verification of the accuracy of the model but not used in specifying the model. It can be called the validation dataset. Users [11] and Boilers [12] are used to support this analysis. It contains 1,482 real profiles and 1,338 fraud profiles, and the data is available in CSV format for automatic extraction.

## Data Preprocessing

It is worth noting that in real life, data and images are usually incomplete, inconsistent, and full of errors. To make sure that the data is ready for the model, there are a few steps you can take.

## Data Cleaning

Removal of Data Adding or classification of the wrongly added data is termed data cleaning. Sometimes the gathered data contains a lot of unwanted data and null values. Data may be there in un-structured formats as well. The unwanted data will be replaced by an approximated data. Some null values will be filled by adding static values. There are also scenarios like that where only garbage value data sets are gathered only. These are some steps in pre-processing. After data cleaning, it is now ready for the next step...

 Extract features based on profile meta-data, posting behavior, linguistic features, and network characteristics.

**Natural Language Processing:**

Apply NLP models or techniques to the text content in the profiles, such as posts and comments and find out the linguistic markers which could identify fake profiles.

## Feature Extraction

1) Tokenization: It is breaking words in a text in a way that the machine can understand. In essence, it breaks texts down into smaller pieces which are termed as tokens that can either be words, characters, or sub-words. It is the primary step in modeling text information. Tokenization can be put to use in generating features for model training by considering texts as tokens which allow having counts of tokens.

1. A Stemmer based on the natural language processing algorithm called Porter Stemmer.
2. Application of NLP to convert words into their base or root forms. It is commonly referred to as stemming, which involves the removal of inflectional and morphological endings from words to aid in the simplification of text data and normalization for analysis purposes.

**1) Main Characteristic of Porter Stemmer: Rule-Based:**

A) It is rule-based: Applying a chain of rules so that the suffixes are stripped off from words. These have been developed to cover as many word forms as appear in the English language.

Iterative processing: The process of stemming is carried out in terms of multiple steps or iterations where all the rules are applied in a consecutive manner to strip away the word further.

C) Another NLP tool that can be used for the normalization of text is the WordNet Lemmatizer, which normalizes words to their base or dictionary form, known as the "lemma". Unlike stemming, lemmatization ensures that the result of the reduction is a valid word by utilizing linguistic knowledge.

**2)Main Features of WordNet Lemmatizer :**

Linguistically Aware: WordNet Lemmatizer uses the information provided in WordNet, the lexical database of English, about word meanings and accordingly selects the appropriate lemma of a word.

As such, it can bring the words to their root word status with an eye for its part of speech.

POS Sensitive: To lemmatize correctly, the WordNet Lemmatizer needs the part of speech of the wordSentiment

**Classification:**

1) By sentiment extraction, we can say that which polarity of the sentence or word is going to determine. Classification is a technique in which we classified the text information into a given number of classes. The special task of text classification, who intended to classify a text according to the emotional words of sentimental polarities, is called sentiment classification.

# 1.2 Motivation

Fake profiles present a significant challenge across social media, dating platforms, and professional networks, often contributing to misinformation, harassment, and fraud. The motivation behind this project stems from the pressing need to combat these negative impacts that erode trust and safety online. With the exponential growth of digital spaces, individuals, companies, and society at large face increasing risks from fake profiles, which can mislead, scam, or otherwise harm authentic users. By creating a robust Fake Profile Identification system, we aim to support platforms in maintaining the integrity of their user base and promoting positive interactions. This project is an attempt to leverage advanced technology to improve the online experience, uphold digital ethics, and protect both individual privacy and reputation.

# 1.3 Sustainable Development Goal of the Project

The Fake Profile Identification project aligns with several UN Sustainable Development Goals, primarily Goal 16: Peace, Justice, and Strong Institutions. By detecting fake profiles, the system will help reduce cybercrimes, online harassment, and other forms of digital misconduct, contributing to a safer and more accountable online environment. Additionally, the project supports Goal 9: Industry, Innovation, and Infrastructure, as it promotes the development of resilient digital systems and innovative tools that address real-world challenges. Implementing a sustainable, efficient, and fair solution to identify fake profiles will not only reduce harmful digital behavior but also encourage responsible technology use and help ensure ethical standards in AI and machine learning applications, laying a foundation for future developments in digital ethics and online safety.

# 1.4 Vision of the Project: Fake profile Identification

The Fake Profile Identification system aims to provide an intelligent, reliable, and user-friendly solution for detecting and mitigating the spread of fraudulent accounts across social and professional networks. By leveraging machine learning algorithms and natural language processing (NLP), this system will empower platform administrators and users alike to identify profiles exhibiting suspicious or inauthentic behaviour. The end goal is to enhance online trust, create safer communities, and reduce harmful online interactions. The platform will also ensure that genuine users are protected from false positives by offering high accuracy and transparency in how it flags suspicious activity. Ultimately, this system will set a new standard for maintaining authenticity in digital spaces, building a trusted ecosystem for interactions, business, and information sharing.

## 1. Audience

In any digital environment, users seek safety, transparency, and authenticity. The target audience for the Fake Profile Identification system includes platform administrators, users, and businesses who rely on the credibility of online interactions.

* **Platform Administrators:** Social media managers, security teams, and content moderators need tools to quickly and accurately identify fake profiles to maintain platform integrity. They face challenges from the sheer scale of user interactions, making it difficult to track suspicious behavior without automated tools.
* **End Users:** Regular users of social media, dating, and professional networking sites are vulnerable to interaction with fake profiles, leading to potential emotional harm, financial scams, or even identity theft. This audience requires a trustworthy system that safeguards their experience without compromising privacy.
* **Businesses and Brands:** Companies using social platforms for marketing and customer engagement risk reputational damage if their online presence is tainted by fraudulent activity. Fake profiles can skew metrics, deceive genuine customers, and damage brand reputation. These stakeholders need a solution that ensures authenticity in brand interaction and customer engagement.

## 2. Needs

Each audience group has unique but interconnected needs. Understanding these needs helps shape

the product’s design, functionality, and overarching goals.

* **Need for Safety and Trust:** Users need to trust the profiles they interact with and have confidence in the platform’s ability to protect them from harmful interactions. This means identifying fake profiles accurately while maintaining user privacy and data security.
* **Need for Efficiency and Scale:** Platform administrators require a solution capable of handling millions of profiles, processing data in real time, and integrating seamlessly with their existing systems. A high level of automation is essential for efficiently flagging, analyzing, and taking action against suspicious profiles.
* **Need for Business Continuity and Metrics Integrity:** Businesses need accurate metrics to assess user engagement and avoid misleading data caused by interactions with fake profiles. A solution that identifies and removes fake profiles allows businesses to gain accurate insights into their audience, ensuring marketing efforts are well-targeted.

## 3. Product Overview

The Fake Profile Identification system will employ machine learning (ML) algorithms and natural language processing (NLP) to detect fake profiles accurately and efficiently.

* **Core Features and Functionalities:** The system will analyze multiple data points, such as profile activity, language patterns, follower behaviors, and interaction history, to detect inauthentic profiles. Advanced ML models like Support Vector Machines (SVM), Random Forests, and NLP-based sentiment analysis will work together to identify anomalies that signal fake profiles.
* **Automated Reporting and Actions:** For platform administrators, the product will provide automated reporting, allowing them to review flagged profiles and take action (e.g., warnings, account suspension). It will also include a dashboard for detailed insights on flagged profiles, reasons for flagging, and historical data.
* **Privacy Protection:** The system will operate under strict privacy standards, analyzing data with minimal user intrusion and ensuring that any flagged profiles are reviewed with transparency to avoid bias or wrongful identification.

## 4. Values

The Fake Profile Identification system is built on core values that guide its development,

use, and contribution to the digital ecosystem:

* **Integrity:** The system upholds the integrity of online interactions by preventing fake profiles from misleading or harming genuine users. It serves as a reliable tool to maintain the trustworthiness of online communities and brand-user relationships.
* **Transparency and Fairness:** The system will prioritize fairness in profile flagging, using unbiased algorithms and offering platform administrators clear insights into the reasoning behind each flagged profile.
* **Privacy and Ethical AI:** Ethical standards guide every stage of product development, ensuring privacy protection for users and transparency in how data is used. The system will comply with global data privacy regulations, giving users and administrators confidence in its ethical operation.
* **Sustainability:** As a digital tool designed to improve social media ecosystems, the system contributes to sustainable digital spaces where safety, trust, and respectful interactions can flourish.

## 5. Product Goal

The ultimate goal of the Fake Profile Identification system is to create a more secure, transparent, and authentic online experience for all stakeholders. By providing a reliable tool that identifies, analyzes, and takes action against fake profiles, the system seeks to enhance digital trust and foster environments where real connections can thrive.

**CHAPTER 2**

# LITERATURE SURVEY

In the article titled "Social Media Malware Detection System," Nambouri et al. [1] propose a method to help detect and prevent criminal activities that appear on social media platforms. Community sites pose a real threat in today's online environment. Several models have been proposed that can identify misinformation. Here they use machine learning to focus on monsters and troll characters. The information from diverse information blogs will then be used for progress by witches and Trolls. The information which then becomes problematic, then they put it in; cleaning after that, the re-storage, and later the presentation of the features of fakes after the cleaning because the missing place is where the fake feature will then be found. Sometimes, after cleaning the storage, that specific area is readied, and the information is placed in a non-institutional file for further use; hence, it is the procedure by which false information can be ruled out. [2] Compare similar performance and volume areas to identify fake accounts in social media sites. One of the advantages of the application function is determining fake money with much precision. The combination of five indicators of 400 fake and real money in the machine learning algorithms dataset is the result of this program. Here you can find 200 fake numbers and 200 real numbers used to calculate the accuracy of supervised machine learning calculations. On the other hand, skin calculation can also be used. According to the true value of appearance, the image is collected to calculate the fake true calculation. The dataset was run under 10-fold cross-validation through machine learning. Factsbook13 Key Points for Leasing Personal Data. Provide 60% 80% accuracy and 20% error. KNN calculates 60% accuracy. Other classifiers like Bayesian and Selection Tree classifiers contribute up to 80% accuracy. BrEmbedded deep learning algorithm Identifies which image in the set of images are human. If the photo has more than 13 percent of skin exposure percentage, it is a fake profile because most fake accounts own this percentage If the photo shows more skin, then the account is counted as a fake one. Leave this time to harm notThis time the message will be cleaned andsaved again before cleaningAfter cleaning, fake people and incomplete sites are wrong Self. Sometimes recently cleaned storage space is prepared and data is stored in irrelevant files for future use, which plays an important role in eliminating false data. [2] Use similar performance and volume area calculations to identify fake accounts on dating sites. The benefit of the application function is that it will detect fake money with an accuracy rate. The output of this program utilizes 400Five machine learning algorithm data set, fake, and real money. Here you can find 200 fake numbers and 200 real numbers utilized to compute supervised accuracy machine learning calculations. Conversely, skin calculation can also be used. The data set was cross-validated 10-fold in a machine learning algorithm based on appearance true value by gathering the image to calculate the fake true account. Factbook13 Key points for Leasing Personal data. Give accuracy of up to 60% 80% error. KNN computes accuracy up to 60%. Other classifiers like Bayesian and Choice tree classifiers give an accuracy level of up to 80%. br>Embedded deep learning algorithmDetermine whether these images in the picture are human. If photo possesses more than 13 percent exposure percentage of skin then consider it as a fake profile because most fake profile contains this percentage account to be a fake account.

Tran et al. [3] presented their work on using machine learning, Fluffy rules and artificial intelligence to detect spam. This method overcomes spam's limitations by using semi-learning methods. Therefore, in this paper, fuzzy logic can be used to process large datasets with efficiency and detect spam emails within a short period. It saves the labor cost, time and usage of complicated procedures. Fake accounts on social media can send spam. And can reach spam profiles and start responding to them. Accept some posts and spam notifications containing account URLs. [4 has used machine learning and NLP methods for improving the accuracy of the detection of fake accounts. Many issues such as privacy issues, cyber bullying, trolls etc. are sent to the social sites by support vector and naivety statements. The money that is collected for making an application is not released at the social sites. This paper is using Facebook dataset for detecting illegal accounts. Here in addition to NLP pre-processing strategy computational machine learning is used for analysing the dataset and the contours classification is done with the use of Support Vector Machine classifier which is highly essential to know whether the contour is a false one or honest. br> variation. [5] It concentrates on the system engineering. The two design concepts in the initial design will analyze the details of the account and prepare the indicators using TGA. According to the verification process, if the user pays to two or more accounts, then they should be asked to give security. The concept of Bolster Vector Machine BOW (bag of words) is applied in the second architecture, proving the readability of a word. Harmful Word Count, harmful words written in records. BOW has SVM (Inverse Vector Machine) prediction mode, combines the datasets and separate datasets for testing and planning. It is used to count the numbers of injuries in an account and find the wrong amount based on the details shown in the account at that point of time send warning messages to give real advice> Continue accounting.

The account should be used only when the count is more than three, after verifying the customer.

The most significant issue that social organizations face is that the users are not authenticated before information sharing. Infrastructures such as fake posts' friend names and follower names are developed to access the false information that people develop through social media platforms. Location, and profile picture to authenticate whether a social media account is indeed owned by a human. Human and bot accounts share common characteristics like names. The engineering function produces money generated by bots. Identify three machine learning modules applied to detect false information in the article Entropy refers to the description of the engineering characteristics that either determine success in account or not. The current entropy is used in describing the engineering characteristics that determine the success of the account or not. So the FI score is 49.7% by chance and by itself. It would be estimated at 50%, which is not ideal. This device Sometimes MLAs are associated with diseases spread across a large dataset. [7] Determine whether the online presence is true or false. These functional models have been used to define profiles using various classification algorithms, including SVM, naive Bay, and decision tree.

**CHAPTER 3**

# SYSTEM ANALYSIS

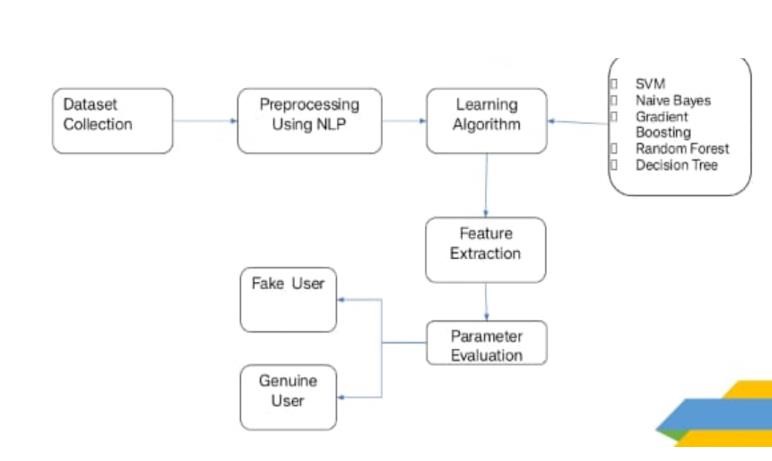
## 3.1 EXISTING SYSTEM

There have been several studies done to review techniques for detecting compromised accounts, thus offering a basis for future research in this field and forming a need for thorough analysis [3]. The survey conducted by Drury et al. [4] has the objectives of recording social media crimes, classifying crimes with similarities, and giving recommendations on further research using publicly available datasets. A literature review discussed recent developments in ML methods for both detection and classification in social media platforms such as Facebook, Instagram, LinkedIn, Twitter, and Weibo [15]. In any application, feature selection in classification problems is important in dealing with high-dimensional data. Bahassine et al. [7] proposed a new Arabic text classification approach based on ImpCHI and SVM classifiers, showing an improvement of 90.50% compared with the traditional metrics for feature selection. A study has been conducted utilizing dynamic feature selection and ML algorithms in the identification of spam users on Twitter since malicious activities have risen [10]. Purba et al. used 17 features for fake Instagram followers; they proposed supervised ML models that classify authentic and fake users with the highest accuracy coming from the RF algorithm [8]. The use of ML techniques for fake profile detection has received much more attention in recent times. A model is proposed to classify a cluster of accounts as malicious or legitimate using user-generated text, pattern within the cluster, and cross-cluster comparison among the overall user base. Hassan et al. [6] proposed supervised learning algorithms with SVM and RF classifiers for detecting fake social media accounts. The results show that the proposed model performs better as compared to other techniques for protection against online threats in the networks. Liang et al. [11] worked on spam detection on Sina Weibo, a Chinese microbiological website using ML where it showed that LR models outperform NB and DT J48 in terms of precision, recall, F-measure, and AUC value. ML algorithms, a cost-sensitive genetic algorithm for automated accounts, and the SMOTE algorithm for unevenness in the fake profile dataset were used to identify Instagram fake and automated accounts in a study [14]. Elyusufi et al. [9] applied DT and NB algorithms to classify user profiles as genuine and fake with the purpose of detecting fake profiles on social media. Sallah et al. [16] proposed a machine learning architecture for the detection of fake Instagram accounts by applying some techniques such as bagging and boosting, synthetic minority over-sampling technique SMOTE, and SHapley Additive exPlanations (SHAP) values, with a combined accuracy of 96% using XGBoost and RF models. However, most of these studies lack accuracy and extensive evaluation and comparison with state-of-the-art classifiers. Aydin et al. [17] used ML techniques to detect fake accounts on social networks like Twitter, with LR being the most effective classification method. A research in Facebook data employed DM methods to detect fake profiles: ID3 was the method with the highest precision with regards to misuse accounts for malicious activities [18]. Sahoo and Gupta [19] presented an ML approach to identify fake profiles from Facebook and Twitter through multimedia big data analysis of both content and profile features for its potential. A study on cybercrime prevalence, particularly against women, has been done through ML algorithms for the detection of spam in Instagram accounts [20]. It results in a declaration that RF outperforms LR with a result to detect spam profiles. An algorithm has been proposed for spammers and fake account detection in Instagram by Kaushik, achieving a precision and accuracy of 93% and 91% that more focused on the problems of security. The author also reviewed the use of different detection methodologies, such as deceptive, predictive, linguistic, and clustering; however, these methods are appropriate only for fake news detection and not for fake profile detection. Deep learning has emerged as a potential method in the field of fake profile identification on OSNs. Traditional systems struggle with fake profiles because of synthesized features, and deep learning can learn complex patterns from raw data. RunFake introduced a dynamic CNN that classifies malicious accounts using general activation function called RunMax by enhancing the training and testing accuracy [2]. A method was developed that is more effective with features involving user profile data. A heterogeneous stacking-based ensemble learning framework is proposed to improve spam detection in social networks. It employs six base classifiers and cost-sensitive learning which boosts the detection rates on imbalanced datasets which forms a backbone for the proposed work [5]. A study has proposed the use of digital face-processing authentication as a double-factor authentication method for OSN and achieved 95% using deep learning classification and SVM at 97.8% for fake profile detection.

**3.2 PROPOSED SYSTEM:**

Many studies have been focused on using LSTM-RNN for fake profile identification. Such studies often focus on the use of LSTM-RNN to analyze various characteristics about a profile, such as the text within a profile description, behavior patterns associated with the profile, or the interprofiles. The goal is to identify the patterns of fake profiles and design models that can better classify the profiles as fake or real. Results from these studies are encouraging, with LSTM-RNN being highly accurate in the detection of fake profiles. But more improvements are still achievable, and the research continues, trying to make the models even more accurate and robust. In addition, some of the proposed methods utilize the application of natural language processing techniques to analyze the text content people post in forms of posts and comments to improve precision for detecting fake profiles. Some of the recent works suggest that multi-modal learning techniques can be applied together with both visual and textual information for fake profile detection. That way, it may enhance model robustness and increase a high distinction between real and spurious profiles as well as profile impersonation. Hence, in a nutshell, the proposed system works for the identification of fake social media profiles based on LSTM-RNN has focused mainly to adopt the use of user metadata and text content posted by the users for training a model and recently made an effort.

**BLOCK DIAGRAM**



**Figure 1 Block Diagram**

**PROJECT FEATURES**

1. Low cost.
2. High speed networking.
3. Low power consumption.
4. Light weight network
5. Broadcast communication
6. As a security purpose.

## CHAPTER 4

**METHODOLOGY**

Identifying the methodology that can create a machine learning algorithm for detecting fake profiles based on natural language processing encompasses the following major key steps. Here's a structured methodology:

### 1.Problem Definition

Objective: Clearly define what a fake profile is (such as bot accounts, impersonation and etc.). Scope: Identify the types of platforms targeted ( e.g., social media, dating apps) and types of profiles targeted.

### 2. Data Collection

Dataset Sources: Gather datasets from social media, public profiles or synthesize them. Know both real and fake profiles.

Label: Proper label of the profile as real or fake best achieved by a mixture of manual and automated methods.

#### 3. Data Preprocessing

Text Cleaning Such information contains irrelevant material that needs to be removed from textual data, like URLs and special characters. **Feature Extraction:**

NLP Features: Extract features like word count, sentiment score, and style of language with methods like TF-IDF and word embeddings such as Word2Vec, GloVe. Profile Features: These include metadata like the age of account, number of friends or followers, or even analysing profile pictures.

#### 4. Exploratory Data Analysis (EDA)

Graphs and charts help understand the distribution of data and relationships between features:

Visualizations.

Statistical Analysis Patterns and Differences Identification between Real and Fake Profile Pictures

### 5. Feature Selection

This technique selects the most impactful features by using techniques such as correlation analysis, feature importance from models like Random Forest, or recursive feature elimination.

#### 6. Model Selection

Choose appropriate ML algorithms depending on problem size or complexity and data size.

Options available are:

Supervised Learning: Decision Trees, SVMs, Random Forest, Gradient Boosting, Neural Network.

Unsupervised learning: clustering techniques for anomaly detection.

#### 7. Model Training

Divide the data into a training set and test set, say 80/20. Train selected models using the training set and hyperparameters that lead to best performance.

#### 8. Model Evaluation

Apply appropriate metrics like accuracy, precision, recall, F1-score, and AUC-ROC for assessing model performance on test sets.

#### 9. Model Interpretation

Use cross-validation techniques for better generalizability: SHAP or LIME to understand how different features influence model prediction.

#### 10. Deployment

Integration of the model in a real-time system to detect the fake profile and ensure it can process and handle the new data. Intensive performance monitoring for updating the model to work more accurately.

#### 11. Feedback Loop

Collect user feed and new data for the fine-tuning of the model; feedback mechanism should be integrated to ensure continuous learning.

#### 12. Ethical Considerations

Ensure normal privacy and regulations such as GDPR compliance.

Transparency of the limitations of the model and their limitation in performing the task so that bias in the training data is not developed.

### 4.1 GENERAL ARCHITECTURE

1. **Data Collection Module**

Collect data from different sources (social media APIs, web scraping). End

Store the retrieved data in a structured manner, for example, within databases.

1. **Data Preprocessing Module** 
   * Cleans and preprocesses text data
   * Extracts features for NLP and profile-based features
2. **Exploratory Data Analysis (EDA) Module** 
   * Draws analysis and visualization of distribution
   * Generates insights based on feature selection
3. **Model Training Module** 
   * Split the data into training and testing sets.
   * Train the selected ML models using the training data.
   * Hyperparameter tuning for the best performance.
4. **Model Evaluation Module** 
   * Assess the model's performance using the test data.
   * Extract the performance metrics such as accuracy, precision, etc.
5. **Model Interpretation Module** 
   * Interpretation techniques like SHAP, LIME, for better understanding of predictions.
6. **Deployment Module** 
   * Deploy the trained model into a production environment.
   * Monitor performance and gather feedback.
7. **Feedback Loop Module** 
   * Update the model according to new data and user feedback.
   * Tune data gathering and preprocessing based on the findings.

# 4.2FLOW CHART

**+-------------------------+**

**| Data Collection | +---------------------**

**----+**

**|**

**+-------------------------+**

**| Data Preprocessing | +------------------**

**-------+**

**|**

**+-------------------------+**  **| Exploratory Data |**

**| Analysis (EDA) | +----------------------**

**---+**

**|**

**+-------------------------+**

**| Feature Selection | +--------------------**

**-----+**

**|**

**+-------------------------+**

**| Model Training | +---------------------**

**----+**

**|**

**+-------------------------+**

**| Model Evaluation | +-------------------**

**------+**

**|**

**+-------------------------+**

**| Model Interpretation | +----------------**

**---------+**

**|**

**+-------------------------+**

**| Deployment | +---------------------**

**----+**

**|**

**+-------------------------+**  **| Feedback Loop |**

**+-------------------------+**

**Figure 2 Flow Chart**

## 4.3 EXPLANATION OF THE FLOW

**Data Collection:** Gather data from relevant sources so that there is a balance between actual and spam ones.

**Data Preprocessing:** Cleaning and preparing the data to feature extraction, which would be used in analysis.

**EDA:** Analysis to understand characteristics about the data and guide feature selection.

**Feature Selection:** Determine the most important features for use in the ML models

**Model Training:** Train multiple models with the preprocessed dataset and selected features.

**Model evaluation:** Evaluate the models against various metrics to find the best one.

**Model interpretation:** Apply interpretive techniques to understand what the model is deciding based on.

**Deploy**: The best model must be deployed in a production environment for real-time identification.

**Feedback Loop:** The model should continue learning from new data and feedback to evolve with time.

It scans fake profiles on social media or other online platforms through a form of analyzing user behavior, content, and metadata. Using several machine learning techniques and algorithms for pattern recognition with various data analysis methods will indicate which accounts are suspicious. Important parts of it are:

**Profile completeness analysis**: This checks if there are essential details missing from the profile, such as a profile picture, bio, or recent posts. Fake accounts usually miss these or include generic images and text.

**Behavioral patterns:** Analyze user behavior, including login frequency, posting patterns, and interactions. Fake accounts display erratic or automated behavior in posting at irregular intervals or liking numerous posts in a short period.

**Quality and consistency of posts:** Analyzes the quality and consistency of posts. Fake accounts may post repetitious content from other accounts or post irrelevant or spammy stuff.

**Network Analysis of Friends/Connections:** Reviews the profile's friend or follower network. Fake accounts generally have very few followers or too many follow requests sent in than it interacts with actual users.

**Language Analysis:** NLP to recognize patterns of common phrases, generic comments, or even unusual linguistic patterns that often characterise bots and spammers.

**Image and Metadata Scanning:** Look at profile images to detect the use of stock photos; duplication across multiple accounts; low-quality images. The system can also check data for a strange recent creation date, as well as an unsually high activity profile.

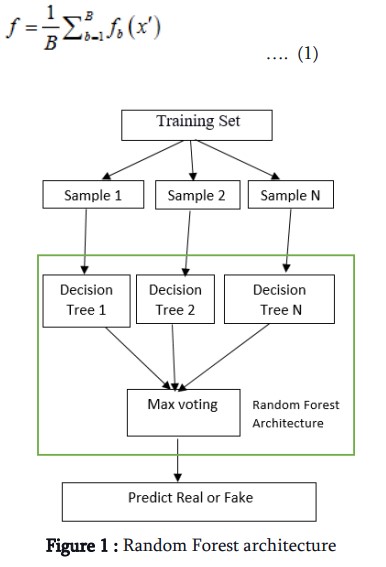
**Machine Learning-Based Prediction:**It relies on the use of supervised learning models that are informed by historical fake and real profiles to predict the likelihood of a new profile belonging to a fraudulent category. Account age, interaction type, and content patterns will be fed into the model.

**Giving the Trust Score**: All these factors lead to giving a "trust score" to the profile from the code with regard to the likelihood of it being authentic. Low score triggers an alert or even suspends the account automatically to be reviewed further.

**4.4 Algorithms:**

**Random Forest:**

This approach includes an example like ensemble learning: the method of random forest (or randomdecision forest). It is used in machine learning merely because it can be effectively used for both classification as well as regression problems. Like in Figure above, the random forest takes predictions from each tree and predicts by majority votes of the projections instead of pure use of one decision tree. The random-forest, however, produces nearly all decision trees unlike a decision-tree approach, so the outcome seems to be the sum of nearly all decision trees produced. Random-forest approach for profile detection. The model accepts the data and gives relevant outputs. The trees (FB) are fitted to the sample for the given set of 1 2, nX x x= and 1 2, NY y y= responses using the bootstrap aggregating process. At fixed regular intervals, a random sample is chosen (B times). The method utilized in finding the outputs given for any sample(x') following the training is by



**Gradient Booster:**

Ensemble learning is a method of the random forest method for regression, using sub-samples with varying stochastic optimization settings. The limitation of random forest is that it's most useful when full inputs are present, or when no missing data exists. To counter this problem, the author uses gradient boosting strategy.The idea in gradient boosting itself is to combine many weak learners to form a strong rule. One of the main advantages of this approach is that it still makes reasonably accurate predictions in case some input factors are missing. Here the decision trees are summed together, and their mistakecorrecting their predecessors'. This means that such misclassified instances are treated with higher priority in further trees. Apart from this, XG Boost can evaluate the importance of different features in the data set, where the most impacting features can be presented, which greatly influence its decision-making. XG Boost classifiers are very wide algorithms and are among the most famous for being very accurate and efficient in any classification task. They make up the basis of the solution. A labeled dataset is used for the training of the model, and in such a dataset, authentic and fraudulent profiles are distinctly labeled. By the application of hyperparameter tuning techniques like grid search or random search, more optimum performance is achieved from the model. All aspects of the effectiveness of the model can be completely understood with the help of evaluation metrics like ac-curacy, precision, recall, and F1-score.

**Decision Tree:**

Another popular algorithm that can be effectively used to identify fake profiles is a decision tree. This works by learning a sequence of simple decision rules based on features of a profile in order to classify it as real or fake. How a decision tree can be applied to this problem:

**1. Feature Selection for Fake Profile Identification:**

Similar to other algorithms, before building a decision tree, you need to select relevant features that can differentiate between fake and real profiles. Some potential features include:

•Account Age: Number of days since account was created.

•Post Frequency: The number of posts made over a period of time (daily, weekly). •Follower/Following Ratio: A very high ratio of following compared to followers might indicate dubious behavior.

•Engagement Rate: This refers to the number of likes and comments per post, which would indicate a real interaction.

•Profile Completeness: Does the profile have a bio, picture, or other key elements?

•Content Consistency: Original, diverse content versus generic and repetitive content.

. These features serve as the input to the decision tree algorithm

**2. How a Decision Tree Works in Fake Profile Detection:**

It works like splitting the dataset into subsets based on feature values. This algorithm is recursive by nature and splitting the data into branches until it reaches a decision point at the end, called the leaf node, that indicates whether it is fake or real.

The steps are:

Splitting Criteria: The decision tree splits the data at each node based on a condition such as "Is the account age < 30 days?" The algorithm chooses a feature and a threshold that maximizes the difference between real and fake profiles for the split.

Impurity Measures: To select the best split, decision trees rely on measures like Gini Index or Entropy. As long as these measures are valid functions of a node's "impurity" – i.e., the degree to which the real and fake profiles in that subset of data have been "mixed up" together – the tree continues splitting until nodes are nearly pure, i.e., consist of mostly real or mostly fake profiles.

Tree Growth: The tree keeps growing based on this process in which every branch is representing a decision rule for example: "Account age < 30 days, and follower/following ratio > 10". At the leaf nodes, the tree outputs a classification — either "real" or "fake" profile.

**3. Example of Decision Tree on Fake Profile Detection:**

Let's take a very simple example:

* **Root Node**: It can check first the age of the account in the decision tree. If over 180 days old, then it may be real. Otherwise, it continues to the next condition if younger.
* **Second Node**: it checks if the following/follower ratio is more than 180 days. High ratio may be one of a spam or fake account.
* **Third Node:** The tree may check the engagement rate or content quality if the follower account has a good following/follower ratio but still appears unclear.

**Gaussian Naive Bayes:**

Gaussian Naive Bayes is a classification technique used in machine learning based on the probabilistic approach and Gaussian distribution. Gaussian Naïve Bayes (GNB) is a variant of the Naïve Bayes algorithm that assumes features follow a Gaussian (normal) distribution. GNB is a model that plays an extremely crucial role in fake profile detection. They classify profiles into fake or genuine based on some feature sets derived from user behaviors, characteristics of the profile, and network activities. For each feature, this model assumes the value is conditionally independent given the class label-that is, fake or legitimate. It further assumes that each feature follows a Gaussian distribution, which is suitable for continuous variables like the time between messages, number of posts per day, or the time spent online. Once a profile is detected, the GNB classifier classifies whether a new or existing profile is fake or legitimate based on the posterior probability for each class; the class with the highest probability is assigned to the profile.

**Support Vector Machine:**

Support vector machines or SVMs are a class of supervised learning methods applied on the classification, regression tasks and outliers detection. One can classify data with help of SVM by determining such hyperplane that splits points of one class from another and remains as far from given data points as possible on either side. The best or maximal margin hyperplane here in SVM means a case where the distance between two classes is at maxima. Similar to other machine learning models, SVM relies on extracting relevant features from user profiles for classification. Features could include:

**Profile attributes:** The length of the username, usage of profile picture, age, completeness of bio, etc.

**Behavioral features**: Posting frequency, interaction pattern, message characteristics, etc.

**Network feature:** Followers, Connection, reaching other people, etc.

A characteristic of fake-profile detection usually has imbalanced datasets, that is, far fewer fake profiles exist than real ones.

SVM can be tuned either through class weights or through the C parameter to accommodate such imbalances also, whereby the minority class gives importance, thereby preventing classifier bias toward the majority, legitimate profiles.Majority vote: After the trees are trained, the model classifies a profile as fake or legitimate based on the majority vote from all trees in the forest. The result is an ensemble decision that is typically more accurate than individual decision trees.

**Further judicious feature engineering techniques can improve the performance of RFC in the detection of fake profiles:**

**Behavioral analysis:** Features like the time between posts, frequency of comments, or patterns of interaction with other profiles can be engineered to help detect anomalies indicative of fake profiles.

**Textual features:** For social media website or review websites a model can analyze the actual content posted or the reviews in order to detect spammer language use or text patterns repetition.

**Image-based features:** Profiles can be flagged that carry no profile picture or low quality or a simply stock image by extracting and analyzing the related image features.

**Time-dependent features:** False profiles tend to have time patterns that are irregular, such as odd login times or spurring activity for long periods of no activity. These can be represented in features for RFC.

**9. Feature Importance in RFC for Fake Profile Detection**

One of the neat things about Random Forest is that it can produce a list of the importance of each feature in the classification decisions. This can prove quite useful in the task of fake profile detection by making it easier to comprehend what factors are the best predictors of fake profiles. Here is an example:

•**Post frequency:** Fake profiles may post far more often (or much less often) than legitimate profiles.

* **Follower/following ratio:** Fake profiles are those that follow a great many users but have fewer followers.
* **Profile completeness:** Fake profiles often tend to leave the most significant parts of their profiles, for example, the bio or profile pictures incomplete.
* **Engagement metrics**: In case of fake profiles, these engage with a much fewer unique users but in bulk numbers such as spamming a same person.

**CHAPTER 5**

# IMPLEMENTATION AND TESTING

This involves several stages, from setting up the environment to testing the model in terms of performance. Here's a step-by-step guide on how it is done:

1.Install Python and required libraries

1. Collect Data

Gather Profile Data from sources such as social media APIs or web scraping.

Store the data in a CSV file or database

1. Preprocess Data

Clean the Text: Remove URLs, punctuation, and any other information that's not critical.

Extract Features: For text data use TF-IDF and add profile metrics such as account age.

1. Split Data

Split your dataset into training data and test data (e.g., 80% for training and 20% for testing).

Here's a simplified step-by-step guide to implementing and testing a fake profile identification system:

6.Set Up Your Environment

Install Python and required libraries:

pip install pandas numpy scikit-learn nltk spacy

7.Collect Data

Profile data from sources like social media APIs or web scraping.

Store the data in a CSV file or database.

8.Preprocess Data

Clean the Text: Remove URLs, punctuation, and irrelevant information.

Feature Engineering: Apply methods like TF-IDF for text and add profile metrics like account age, etc.

9.Split Data

Divide your dataset into train and test sets (say 80-20%).

1. Train the Model

Choose a Machine Learning Model to train (say, Random Forest).

Fit the model to your training data.

1. Model Evaluation

Make Predictions on the test set.

Check Performance using metrics like accuracy, precision, and recall.

12.Model Interpretation

Use tools like SHAP to understand which features influence predictions.

13.Deploy the Model

Create an API using a framework like Flask to allow others to use your model.

14.Monitor and Improve collect feedback and new data to retrain the model as needed

15. Document the Process

Keep records of methods, code, and results to consult later.

# CHAPTER 6

**RESULTS AND DISCUSSIONS**

One of the concepts that could be utilized to measure the performance of the machine learning model is called the confusion matrix. It would be a table of four different values: true positive, false positive, true negative, and false negative, which can be inputted into the matrix. This can be done for the test dataset whose actual values are known beforehand.

* True positives (TP) is the value in which the examples are correctly determined as positive.
* TN: the value where examples are correctly deter-mined to be negative.

•False positives (FP) - the value in which the examples are negative but are actually determined as positive.

* FN False negatives: the value in which the examples are positive but are actually determined as negative.

From the confusion matrix, regression, accuracy, precision, fmeasure, and error of the distribution model are calculated fro m the confusion matrix. The formula for each performance measure listed above is as follows: TP equals TP/(TP + FN). (TP + TN + FP + FN) equals truth. + Return equals Fmeasure. Validity of perso nal data.

Interpretation of the XG Boost model and its base scores gives an idea about factors lying under the distribution decisions. If the model works satisfactorily, in real-time scenarios, it can actually be applied to detect fake information on online platforms. Again, here, the characteristics of fake data may change over time; hence, continuous monitoring and updating are critical so that the effectiveness of the model becomes perfect.

**CHAPTER 7**

## CONCLUSION

As a result, the achievement of this project was to show multiple applications of machine learning algorithms that come to identifying fake profiles in social media networking. In pursuing that goal, we used classification methods such as SVM, Random Forest, Gradient Boosting, Naïve Bayes, and Decision Tree classifiers with an aim of good accuracy and efficiency.

The Random Forest Classifier showed the best performance, giving 100% accuracy in the identification of fake profiles. This is a sign that it is strong and dependable in its decision-making capability. Its ensemble method by aggregating the decisions made by several decision trees explained its superior performance. Moreover, it can handle a large dataset with high dimensionality and does not overfit, which are the keys to achieving such high accuracy.

The Gradient Boosting Classifier also came out quite promising with 86% accuracy. Gradient Boosting, because it corrects errors from earlier models, presents another significant alternative in relation to Random Forest. However, it performs a little worse in those scenarios in which some data points are missing or profiles are incomplete since it demands more computing power and time to train.

On the basis of this comparison between the two models, it is apparent that selecting the appropriate algorithm in relation to the nature of the data and the specific objectives of the project should be determined. Whereas Random Forest will do great for general classification, there are indeed data sets and areas that need more precision, thus making Gradient Boosting a good choice.

Further work can include more advanced techniques like deep learning or multi-modal learning so that the developed model is upgraded to a much greater level of accuracy and scalability. Implementation of real-time systems with detection could add practical usability to this research, especially for large social media platforms dealing with millions of profiles.

It establishes a starting point for further research and development on this topic and has acted as a foundational understanding of how to utilize machine learning algorithms for the purpose of detecting fake profiles. The methods and findings reported here can help to act as a baseline for further

exploration into more complex detection systems and vastly more diverse datasets.

## REFRENCES

1. Nambouri Sravya et al. (2019). "Identify the Human or Bots Twitter Data using Machine Learning Algorithms". International Research Journal of Engineering and Technology (IRJET).
2. M. Smruthi & N. Harini (2019). "A Hybrid Scheme for Detecting Fake Accounts in Facebook". International Journal of Recent Technology and Engineering (IJRTE).
3. Tehlan, Pooja et al. (n.d.). "A Spam Detection Mechanism in Social Media using Soft Computing."
4. Rao, P. S. et al. (2018). "Fake profile identification in online social networks using machine learning and NLP". Int. J. Appl. Eng. Res.
5. Raturi, Rohit (2018). "Machine Learning Implementation for the Dilation of Fake Accounts on a Social Network". International Journal of Pure and Applied Mathematics.
6. Van Der Walt, Estée & Jan Eloff (2018). "Using machine learning to detect fake identities: bots vs humans". IEEE Access.
7. Kulkarni, Sumit Milind & Vidya Dhamdhere (2018). "Automatic detection of fake profiles in online social networks". Open access international journal of science and engineering.
8. Ala'M, Al-Zoubi et al. (2017). "Spam profile detection in social net-works based on public features". 2017 8th International Conference on information and Communication Systems (ICICS).
9. Elovici, Yuval & Gilad Katz (2017). "Method for detecting spammers and fake profiles in social networks". U.S. Patent No. 9,659,185.
10. Gurajala, Supraja et al. (2016). "Profile characteristics of fake Twitter accounts". Big Data & Society.
11. S. Venkatesan et al. (2017). "Detecting stealthy bot-nets in a resource-constrained environment using reinforcement learning". Proc. Workshop Moving Target Defence.
12. M. H. Arif et al. (2017). "Sentiment analysis and spam detection in short informal text using learning classifier systems". Soft Computing.
13. K. Thomas et al. (2011). "Suspended accounts in retrospect: An analysis of Twitter spam".

Proc. ACM SIGCOMM Conf. Internet Meas. Conf.

1. S. Cresci et al. (2015). "Fame for sale: Efficient detection of fake Twitter followers".

Decision Support Systems.

1. M. Ebrahimi et al. (2016). "Recognizing predatory chat documents using semi-supervised anomaly detection". Electron. Imag.
2. Gowroju, Swathi et al. (2022). "Deep Neural Network for Precise Age Group Prediction through Pupil Using the Optimized UNet Model". Mathematical Problems in Engineering.
3. Swathi, A. & Sandeep Kumar (2021). "A smart application to detect pupil for small dataset with low illumination". Innovations in Systems and Software Engineering.
4. Gowroju, Swathi & Sandeep Kumar (2022). "Review on secure traditional and machine learning algorithms for age prediction using IRIS image". Multimedia Tools and Applications.
5. Swathi, A. et al. (2023). A Reliable Novel Approach of Bio-Image Processing—Age and Gender Prediction. In: Computing Intelligence and Cybernetic Research. Advances in Intelligent Systems and Computing.
6. Gowroju, Swathi & Sandeep Kumar (2020). "Robust deep learning technique: U-net architecture for pupil segmentation". In 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).

**APPENDIX**

## A.SAMPLE CODING

**IMPORT LIBRARY:**

import numpy as np import pandas as pd import matplotlib.pyplot as plt import seaborn as sns import plotly.express as px

import pandas as pd import nltk from nltk.tokenize import word\_tokenize from nltk.corpus import stopwords from nltk.stem import PorterStemmer from nltk.stem import WordNetLemmatizer from wordcloud import WordCloud

import gender\_guesser.detector as gender

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import accuracy\_score,confusion\_matrix, classification\_report from sklearn.model\_selection import train\_test\_split from sklearn.naive\_bayes import GaussianNB from sklearn import svm from sklearn.ensemble import GradientBoostingClassifier from sklearn.ensemble import RandomForestClassifier from sklearn.tree import DecisionTreeClassifier

**LOAD DATASET:**

insta = pd.read\_csv('Copy.csv')

Index(['username', 'gender', 'profile\_pic', 'private\_or\_public', 'posts',

'followers', 'following', 'comment', 'bio', 'work', 'label'],dtype='object')

**# Calculate the length of the 'comment' column and create a new column 'comment\_len'** insta['comment\_len'] = insta['comment'].apply(len)

**# Calculate the length of the 'bio' column and create a new column 'bio\_len'** insta['bio\_len'] = insta['bio'].apply(len)

**EDA:**

fig = px.bar(insta,x="profile\_pic",y='followers') fig.show()

**# Create a pie chart for the 'private\_or\_public' column** fig\_pie = px.pie(insta, names='private\_or\_public', title='Distribution of Private vs Public Accounts') fig\_pie.show()

**# Create a pie chart for the 'private\_or\_public' column** fig\_pie = px.pie(insta, names='work', title='Distribution of work') fig\_pie.show()

**# Histogram of comment lengths** fig\_comment\_len = px.histogram(insta, x='comment\_len', nbins=20, title='Distribution of Comment Lengths')

fig\_comment\_len.show()

**# Histogram of bio lengths**

fig\_bio\_len = px.histogram(insta, x='bio\_len', nbins=20, title='Distribution of Bio Lengths') fig\_bio\_len.show()

**# Scatter plot for comment length vs bio length** fig\_scatter = px.scatter(insta, x='comment\_len', y='bio\_len', title='Comment Length vs Bio

Length',labels={'comment\_len': 'Comment Length', 'bio\_len': 'Bio Length'})

fig\_scatter.show()

**# Create a pie chart for the 'label' column (real vs fake profiles)**

fig\_pie\_label = px.pie(insta, names='label', title='Distribution of Real vs Fake Profiles') fig\_pie\_label.show()

**NLP:**

nltk.download('punkt') nltk.download('stopwords') nltk.download('wordnet')

**# Sample text data (replace these with actual column data)** texts = insta['comment'].astype(str).tolist()

**# Initialize NLP tools**

stop\_words = set(stopwords.words('english')) stemmer = PorterStemmer() lemmatizer = WordNetLemmatizer()

**# Tokenization, stop word removal, stemming, and lemmatization** def preprocess\_text(text): tokens = word\_tokenize(text.lower()) filtered\_tokens = [word for word in tokens if word.isalpha() and word not in stop\_words] stemmed\_tokens = [stemmer.stem(word) for word in filtered\_tokens] lemmatized\_tokens = [lemmatizer.lemmatize(word) for word in stemmed\_tokens] return ' '.join(lemmatized\_tokens)

**# Combine all processed text for word cloud**

all\_text = ' '.join(insta['comment'].tolist() + insta['bio'].tolist())

**# Generate and display word cloud**

Wordcloud=WordCloud(width=800,height=400,background\_color='white').generate(all\_text) plt.figure(figsize=(15, 5)) plt.imshow(wordcloud, interpolation='bilinear') plt.axis('off') plt.show()

**LABEL ENCODING:**

**# Initialize the LabelEncoder** label\_encoder = LabelEncoder()

**# Columns to be label encoded** columns\_to\_encode = ['profile\_pic', 'private\_or\_public', 'label','gender','work']

**# Apply label encoding to each selected column** for column in columns\_to\_encode: insta[column] = label\_encoder.fit\_transform(insta[column])

**MODEL IMPLEMENTATION:**

**GUASSIAN NB:**

**# Initialize and train the model** gnb = GaussianNB() gnb.fit(X\_train, y\_train)

**# Make predictions** y\_pred = gnb.predict(X\_test)

**# Calculate metrics**

accuracy = accuracy\_score(y\_test, y\_pred) conf\_matrix = confusion\_matrix(y\_test, y\_pred) class\_report = classification\_report(y\_test, y\_pred)

**# Print results** print('\nModel: GaussianNB') print(f'Accuracy: {accuracy:.4f}') print('Classification Report:') print(class\_report) plt.figure(figsize=(8, 6)) sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'], yticklabels=['0', '1']) plt.title('Confusion Matrix for GaussianNB') plt.xlabel('Predicted') plt.ylabel('True') plt.show()

**SVM:**

**# Initialize and train the model** svm\_model = svm.SVC() svm\_model.fit(X\_train, y\_train)

**# Make predictions** y\_pred = svm\_model.predict(X\_test)

**# Calculate metrics** accuracy = accuracy\_score(y\_test, y\_pred) conf\_matrix = confusion\_matrix(y\_test, y\_pred) class\_report = classification\_report(y\_test, y\_pred)

**# Print results** print('\nModel: SVM') print(f'Accuracy: {accuracy:.4f}') print('Classification Report:') print(class\_report**)**

**# Plot confusion matrix** plt.figure(figsize=(8, 6)) sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'], yticklabels=['0', '1']) plt.title('Confusion Matrix for SVM') plt.xlabel('Predicted')

plt.ylabel('True') plt.show()

**GRADIENTBOOSTING CLASSIFIER:**

**# Initialize and train the model** gbc = GradientBoostingClassifier() gbc.fit(X\_train, y\_train)

**# Make predictions** y\_pred = gbc.predict(X\_test)

**# Calculate metrics**

accuracy = accuracy\_score(y\_test, y\_pred) conf\_matrix = confusion\_matrix(y\_test, y\_pred) class\_report = classification\_report(y\_test, y\_pred)

**# Print results**

print('\nModel: GradientBoostingClassifier') print(f'Accuracy: {accuracy:.4f}') print('Classification Report:') print(class\_report)

**# Plot confusion matrix** plt.figure(figsize=(8, 6)) sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'], yticklabels=['0', '1']) plt.title('Confusion Matrix for GradientBoostingClassifier') plt.xlabel('Predicted') plt.ylabel('True') plt.show()

**RANDOM FOREST CLASSIFIER:**

**# Initialize and train the model** rf = RandomForestClassifier() rf.fit(X\_train, y\_train)

**# Make predictions** y\_pred = rf.predict(X\_test)

**# Calculate metrics**

accuracy = accuracy\_score(y\_test, y\_pred) conf\_matrix = confusion\_matrix(y\_test, y\_pred) class\_report = classification\_report(y\_test, y\_pred)

**# Print results**

print('\nModel: RandomForestClassifier') print(f'Accuracy: {accuracy:.4f}') print('Classification Report:') print(class\_report)

**# Plot confusion matrix** plt.figure(figsize=(8, 6)) sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'], yticklabels=['0', '1']) plt.title('Confusion Matrix for RandomForestClassifier') plt.xlabel('Predicted') plt.ylabel('True') plt.show()

**DECISION TREE CLASSIFIER:**

**# Initialize and train the model** dt = DecisionTreeClassifier() dt.fit(X\_train, y\_train)

**# Make predictions** y\_pred = dt.predict(X\_test)

**# Calculate metrics**

accuracy = accuracy\_score(y\_test, y\_pred) conf\_matrix = confusion\_matrix(y\_test, y\_pred) class\_report = classification\_report(y\_test, y\_pred) **# Print results**

print('\nModel: DecisionTreeClassifier') print(f'Accuracy: {accuracy:.4f}') print('Classification Report:') print(class\_report)

**# Plot confusion matrix** plt.figure(figsize=(8, 6)) sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['0', '1'], yticklabels=['0', '1']) plt.title('Confusion Matrix for DecisionTreeClassifier') plt.xlabel('Predicted') plt.ylabel('True') plt.show()

**PREDICTION:**

**# Assuming clf\_predict is your trained SVM classifier**

**# Define a function to get user input and make predictions** def predict\_attrition(selected\_features):

**# Create a dictionary to store user input**  input\_data = {}

**# Ask user to input values for selected features**  for feature in selected\_features: value = input("Enter value for {}: ".format(feature)) input\_data[feature] = value

**# Convert user input into a DataFrame with a single row**  input\_df = pd.DataFrame([input\_data])

**# Make prediction using the trained SVM classifier**  prediction = dt.predict(input\_df)

**# Print the prediction with explanation**  if prediction[0] == 0:

print("Predicted Attrition: Fake Account")

else:

print("Predicted Attrition: Real Account")

**# name = input("Enter a name: ")**

**# gender\_guess = d.get\_gender(name)**

**# print(f"The predicted gender for {name} is: {gender\_guess}") # Call the function to get user input and make predictions** predict\_attrition(selected\_features)

**GENDER PREDICTION:**

**# Initialize the gender detector** d = gender.Detector() name = input("Enter a name: ") gender\_guess = d.get\_gender(name)

print(f"The predicted gender for {name} is: {gender\_guess}")

