A

PROJECT REPORT

ON

FAKE ACCOUNT DETECTION USING MACHINE LEARNING AND DATASCIENCE

Submitted to JNTU Anantapur for the Partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

G. SHIVAANI	**208U1A0528**
N. SWETHA	**208U1A0555**
N. RESHMA	**208U1A0550**
V. SRAVANI	**208U1A0573**

Under the Esteemed Guidance of

Mr.R.SHIVA SHANKAR REDDY.,M.Tech.,

Assistant Professor

(ESTD-2009)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GOUTHAMI INSTITUTE OF TECHNOLOGY & MANAGEMENT FOR WOMEN

(Approved by A.I.C.T.E., New Delhi, Affiliated to J.N.T. University,Anantapur) Sai Nagar, Peddasettipalli (V), PRODDATUR-516360, KADAPA Dist. A.P.

2020-2024

GOUTHAMI INSTITUTE OF TECHNOLOGY & MANAGEMENT FOR WOMEN

(Approved by A.I.C.T.E., New Delhi, Affiliated to J.N.T. University, Anantapur) Sai Nagar, Peddasettipalli (V), PRODDATUR-516360, KADAPA Dist. A.P.

CERTIFICATE

This is to certify that the project report entitled "FAKE ACCOUNT DETECTION USING ML AND DS" is the bonafide work done by G.SHIVAANI(208U1A0528),N.SWETHA (208U1A0555),N.RESHMA(208U1A0550)V.SRAVANI(208U1A0573).under the guidance of Mr.R.SIVA SANKAR REDDY.,M.Tech, in the department of **"COMPUTER SCIENCE & ENGINEERING"** at GOUTHAMI INSTITUTE OF TECHNOLOGY & MANAGEMENT FOR WOMEN (Affiliated to J.N.T.University, Anantapur) submitted in partial fulfillment of the requirement for the award of the degree of "Bachelor of Technology".

Project Guide

Mr.R.SHIVA SHANKAR REDDY., M.Tech
AssistantProfessor
Computer Science & Engineering

Head of the Department

Mr.S.YAKHOOB ALI., M.Tech
Assistant Professor
Computer Science &Engineering

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We manifest our deep sense of gratitude to our project guide **Mr.R.SHIVA SHANKAR REDDY.,**M.Tech Assistant Professor for giving enormous cooperation in bringing out this project with artistry. His erudite knowledge and Technical expertise have largely contributed to the success of this work.

We thank our honorable correspondent Smt **K.C.BABAMMA** garu, for providing us with good faculty and for her moral support throughout the course.

We thankful to our principal **Dr. M.RAMA SUBBAMMA** Prof & principal who has encouraged and motivated us to complete the project by providing all necessary facilities to carry out the work in the college.

We would like to express our deep sense of gratitude to **Mr.S.YAKHOOB ALI.,**M.Tech **,** HOD, Dept. of CSE, for providing us with an opportunity to do this project. We would like to thank our institution and all the faculty members of CSE department for their help and guidance. They have been great sources of inspiration to us and we thank them from the bottom of our heart.

We would like to thank all our friends and especially our classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. We have enjoyed their companionship so much during my stay at GITAM, Proddatur. Last but not least we would like to thank our parents and well-wishers.

PROJECT ASSOCIATES

G.SHIVAANI	**208U1A0528**
N.SWETHA	**208U1A0555**
N.RESHMA	**208U1A0550**
V.SRAVANI	**208U1A0573**

INDEX

FIG NO FIGURE NAME PAGE NO

ABSTRACT

Nowadays the usage of digital technology has been increasing exponentially. At the same time the rate of malicious users has been increasing. Online social sites like Facebook and Twitter attract millions of people globally. This interest in online networking has opened to various issues including the risk of exposing false data by creating fake accounts resulting in the spread of malicious content. Fake accounts are a popular way to forward spam, commit fraud, and abuse through online social network. These problems need to be tackled in order to give the user a reliable online social network. In this project, we are using different ML algorithms like Random Forest Classifier. Along with these algorithms we have used two different normalization techniques such as Z-Score, and Min-Max, to improve accuracy. We have implemented it to detect fake Instagram accounts. Our approach achieved high accuracy and true positive rate for Random Forest. Keywords: Data collection, Classification, Random Forest, Normalization

Keywords: Machine Learning, Classification, Random forest Classifier, Mormalization, Accuracy Matrix.

CHAPTER-1
INTRODUCTION

1.INTRODUCTION

1.1 Introduction to Machine Learning

A subset of artificial intelligence known as machine learning focuses primarily on the creation of algorithms that enable a computer to independently learn from data and previous experiences. Arthur Samuel first used the term "machine learning" in 1959. It could be summarized as follows: Without being explicitly programmed, machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things.

Machine learning algorithms create a mathematical model that, without being explicitly programmed, aids in making predictions or decisions with the assistance of sample historical data, or training data. For the purpose of developing predictive models, machine learning brings together statistics and computer science. Algorithms that learn from historical data are either constructed or utilized in machine learning. The performance will rise in proportion to the quantity of information we provide

In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role of Machine Learning.

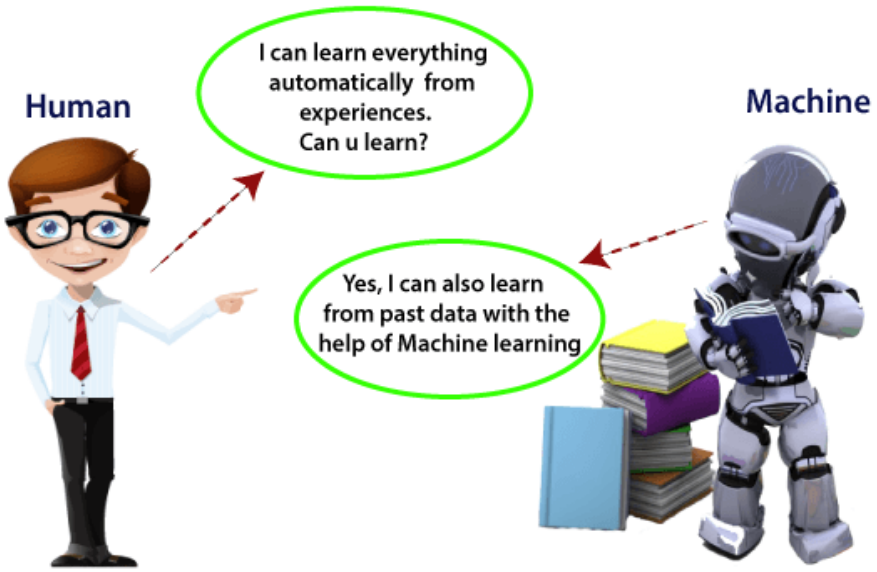


Fig:Machine Learning

1.2 Information on Fake Account

In today's digital age, the rise of social media and online communities has brought about immense opportunities for communication, networking, and collaboration. However, alongside these benefits comes the challenge of combating fraudulent activities, including the creation and proliferation of fake accounts. Fake accounts, also known as bots, trolls, or sock puppets, are often created with malicious intent to deceive, manipulate, or exploit unsuspecting users and the platform itself.

Fake account detection has emerged as a critical component in safeguarding the integrity, security, and trustworthiness of online platforms. By employing advanced technologies such as machine learning (ML) and data science (DS), platforms can proactively identify and remove fake accounts, thereby mitigating the risks associated with spamming, phishing, misinformation, and other forms of online abuse.

1.3 Features

Features play a crucial role in the effectiveness of fake account detection models. Here are some common features used in machine learning models for detecting fake accounts:

1.3.1 Profile Information:

Profile completeness: Percentage of required fields filled out.

Profile uniqueness: Frequency of duplicate profiles.

Profile consistency: Consistency of information across different sections (e.g., mismatch between profile picture and listed age).

1.3.2 Activity Patterns:

Account age: Duration since the account creation.

Posting frequency: Number of posts, comments, or interactions per unit time.

Time of activity: Timing and frequency of activity during different times of the day or week.

Burstiness: Sudden spikes in activity followed by periods of inactivity.

1.3.3 Network Characteristics:

Number of connections: Number of friends, followers, or connections.

Network density: Ratio of actual connections to potential connections.

Network centrality measures: Degree centrality, betweenness centrality, or eigenvector centrality.

Community structure: Detection of communities within the network and the role of the account within those communities.

1.3.4 Content Analysis:

Linguistic features: Language used, grammar, spelling mistakes, and vocabulary richness.

Sentiment analysis: Positive, negative, or neutral sentiment expressed in posts or comments.

Topic modeling: Identification of topics discussed by the account and their relevance to the account's purported interests.

Spam indicators: Presence of spammy keywords, URLs, or patterns commonly associated with spam.

1.3.5 Interaction Patterns:

Engagement ratio: Ratio of received interactions (likes, comments, shares) to the number of posts.

Reciprocity: Mutual interactions between accounts.

Bot-like behavior: Automated or scripted interactions, such as liking or following a large number of accounts within a short period.

1.3.6 Device and Location Information:

IP address: Location and frequency of logins from different IP addresses.

Device information: Type of device used for accessing the account (e.g., desktop, mobile, bot).

Geolocation data: Location information associated with the account or IP address.

1.3.7 Temporal Features:

Changes over time: Evolution of profile information, activity patterns, or network connections over time.

Trend analysis: Identification of sudden changes or anomalies in behavior over time.

These features can be used individually or in combination to build comprehensive models for detecting fake accounts. Additionally, feature engineering techniques such as normalization, scaling, and transformation may be applied to enhance the effectiveness of these features in machine learning models.

1.4 Objective

The primary objective is to develop a robust fake account detection system that can accurately distinguish between genuine and fake accounts within online platforms. The system should be capable of identifying various types of fake accounts, including bots, trolls, spammers, and impersonators, while minimizing false positives and false negatives.

1.5 Problem Statement

The proliferation of social media and online platforms has led to a surge in fake accounts, which are often created with malicious intent. These fake accounts can engage in a variety of harmful activities such as spreading misinformation, conducting scams, and manipulating public opinion. Detecting and removing fake accounts is essential for maintaining the integrity and security of online communities. Therefore, the problem statement for fake account detection using machine learning (ML) and data science (DS). Develop an effective and scalable system for detecting fake accounts on online platforms using machine learning and data science techniques.

1.6 Scope

The scope of the project includes:

- Data collection: Gathering relevant data from the online platform, including user profiles, activity logs, network connections, and content.
- Feature engineering: Extracting meaningful features from the collected data to characterize the behavior and characteristics of both genuine and fake accounts.
- Model development: Building ML models that can classify accounts as genuine or fake based on the extracted features. This may involve supervised, unsupervised, or semi-supervised learning techniques.
- Model evaluation: Assessing the performance of the models using appropriate metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC).
- Deployment: Integrating the trained models into the platform's infrastructure for real-time or batch processing of fake account detection.
- Monitoring and refinement: Continuously monitoring the performance of the detection system and refining the models to adapt to evolving tactics used by malicious actors.

CHAPTER-2
LITERATURE SURVEY

2.LITERATURE SURVEY

Fake account detection is a critical task in the domain of cyber security and social media platforms. The rise of social media has led to an increase in fake accounts, which can lead to several malicious activities, including spamming, phishing, and social engineering attacks. Machine learning and data science techniques have been applied in the past to address this issue, and several research studies have been conducted in this area. Here are a few literature surveys on fake account detection using machine learning and data science.1."A Survey on Detection Techniques for Fake Accounts on Social Networks" by Z. Ji et al. (2018): This survey provides a comprehensive overview of the state-of-the-art fake account detection techniques for social networks. The paper discusses the limitations of traditional detection techniques and presents a range of machine learning-based approaches for detecting fake accounts. “A Survey on Fake Account Detection in Online Social Networks" by S. Agarwal et al. (2018): This survey explores the various techniques used for detecting fake accounts in online social networks. The authors discuss several features that can be used to identify fake accounts, such as network features, user behaviour, and content analysis. The paper also highlights the limitations of existing approaches and identifies future research directions in this area. “Fake Account Detection in Social Media using Machine Learning: A Systematic Literature Review" by H. Lin et al. (2020): This systematic literature review examines the various machine learning techniques used for fake account detection in social media. The authors provide an overview of the most commonly used features and classification algorithms, as well as the datasets used for training and testing. The paper also highlights the challenges of fake account detection and suggests possible solutions.4."Fake User Detection in Social Networks: A Review and Future Directions" by M. Islam et al. (2021): This paper reviews the various techniques used for fake user detection in social networks, including machine learning-based approaches. The authors present a comprehensive survey of the most commonly used features and classification algorithms, as well as the limitations and future research directions in this area. Overall, these literature surveys highlight the importance of fake account detection and the role of machine learning and data science in addressing this issue. They also provide an overview of the most commonly used techniques, the limitations of existing approaches, and future research directions in this area.

CHAPTER-3
SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

3.1 EXISISTING SYSTEM:

The existing system use very fewer factors to decide whether an account is fake or not. The factors largely affect the way decision making occurs. When the number of factors is low, the accuracy of the decision making is reduced significantly. There is an exceptional improvement is fake account creation, which is unmatched by the software or application used to detect the fake account. Due to the advancement in creation of fake account, existing methods have turned obsolete. The most common algorithm used by the fake account detection application is the Random Forest algorithm. The algorithm has few down sides such as inefficiency to handle the categorical variables which has different number of levels Also, when there is the number of trees, the algorithm's time efficiency takes a hit.

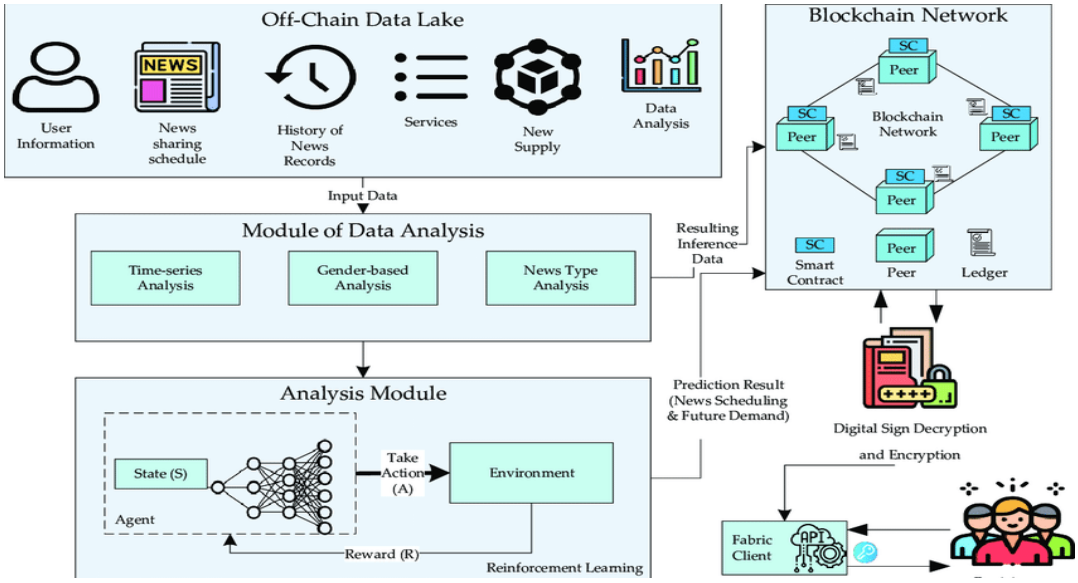
DISADVANTAGES OF EXISTING SYSTEM:

1. In the last decade, Fake Account phenomenon has experience very significant spread, favored by social networks.
2. Some are made only to increase the number of clicks and visitors on a Site.
3. Others, to influence public option on political decision or on financial Markets.
4. It is not possible to find whether the given data is real or fake.
5. Fake data will be increases.

Overall, the existing systems for fake account detection using machine learning and data science are a significant step forward in addressing this critical cyber security issue, but more research is needed to enhance their effectiveness and applicability.

3.2 PROPOSED SYSTEM:

The existing system uses random forest algorithm identify the fake account. It is efficient when it has the correct inputs and when it has the all the inputs. When some of the inputs are missing it becomes difficult for the algorithm to produce the output. To overcome such difficulties in the proposed system we used a gradient boosting algorithm. Graomponents. dent boosting algorithm is the random forest algorithm which uses decision tree as its main



We also changed the way we find the fake account i.e.; we introduced new method to find the account. The methods used are spam commenting, engagement rate and artificial activity. These inputs are used to form decision trees that are used in the gradient boosting algorithm. This algorithm gives us an output even if some inputs are missing. This is the major reason for choosing this algorithm. Due to the use of this algorithm, we are were able to get highly accurate results.

The proposed work comprises of four steps, (i) data classification, (ii) data pre-processing, (iii) data reduction or transformation, and (iv) Algorithm selection. These steps are described below.

Data Classification

Data classification is characterized as the process for deciding the appropriate type, origin of data and appropriate resources for collecting data. In the data classification step the data is selected from various Instagram accounts. We collected Instagram datasets for analysis and to test our model, dataset consists of different attributes such as Username, Profile-Age, Followers, Following, Posts, Profile-picture, Verified, Profile-Description, Inactive-Days, spam-reports and Label.

Data Pre-processing

We used machine learning algorithms in this process to convert and analyse the available raw data into feasible data. It is mainly used for better and accurate results. For instance, some algorithms like Random Forest do not support null values or they need particular format. In such a case data pre-processing is a necessary step. Then we extracted two Comma-Separated Value (CSV) files fake and genuine users, we combined both files by sampling noise in the data then feature labels were added as 0/1 to distinguish fake or real.

We selected particular columns as feature attributes Username, Profile-Age, Followers, Following, Posts, Profile-picture, Verified, Profile-Description, Inactive-Days, spam-reports and Label and then we have removed columns having more null values Profile-Description, Username, and normalized the data for better accuracy. In this method, we distributed the information for training and testing purpose at 80:20. To represent the values in confusion matrix, the model is trained with x- train and y-train data, and then it is trained with test data for precision and recall values.

Graphs are represented as Area under ROC Curve (AUC) and Receiver operating characteristic curve (ROC).

Data transformation

It is a method of converting information from one format or structure into another format. For tasks such as data integration and management, data transformation is a crucial step to improve the accuracy. In our model we used two normalization techniques for data transformation.

Normalization in Data Mining We are using two data normalization techniques such as Z-Score, Min-Max, it is mainly used when dealing with multiple attributes on different scale and to scale the information into smaller range, it is commonly applied for classification algorithms to improve the performance rate, so the attributes are normalized to bring on the same scale.

Z-Score: Z-Scores are mainly based on mean value and standardized score these scores are linearly transformed data value with a mean of 0 and the scores have been given a common standard. It helps to understand the rate of a score as per the normal distribution of the data.

AA-Min-Max: In this method linear transformation is performed on original data, according to these minimum values of that feature is transformed as 0 whereas maximum values are transformed into 1 and remaining values have been changed into decimals between 0 and 1.

3.3 Algorithm Selection:

- Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning**,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.As the name suggests**,** "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the

random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

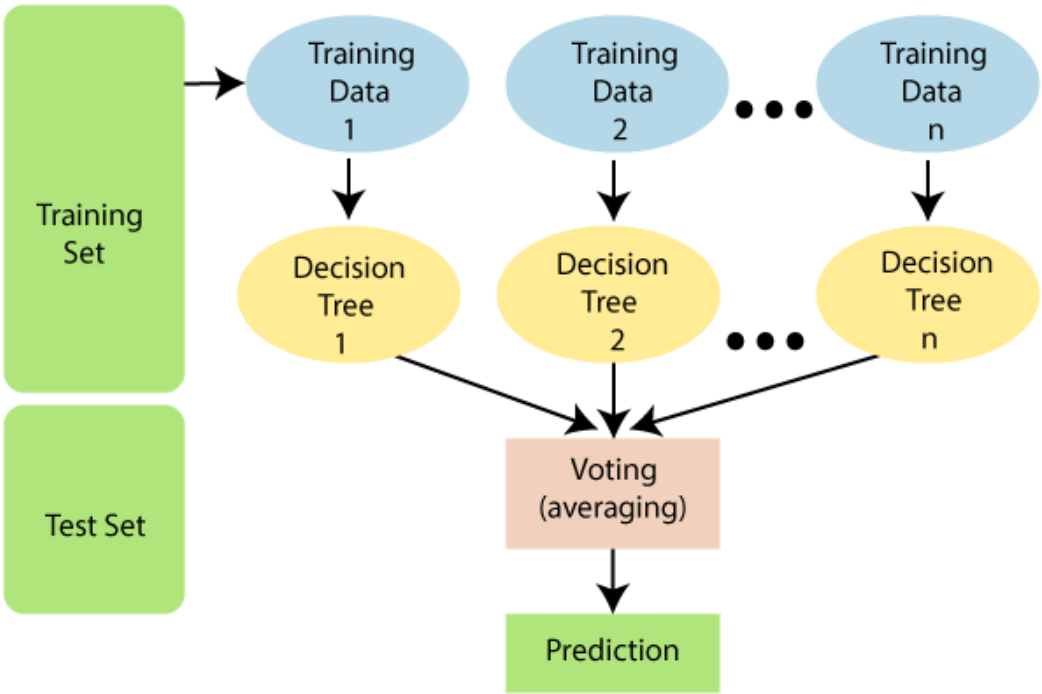
The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

The below diagram explains the working of the Random Forest algorithm:



Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

<=”” li=””>

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

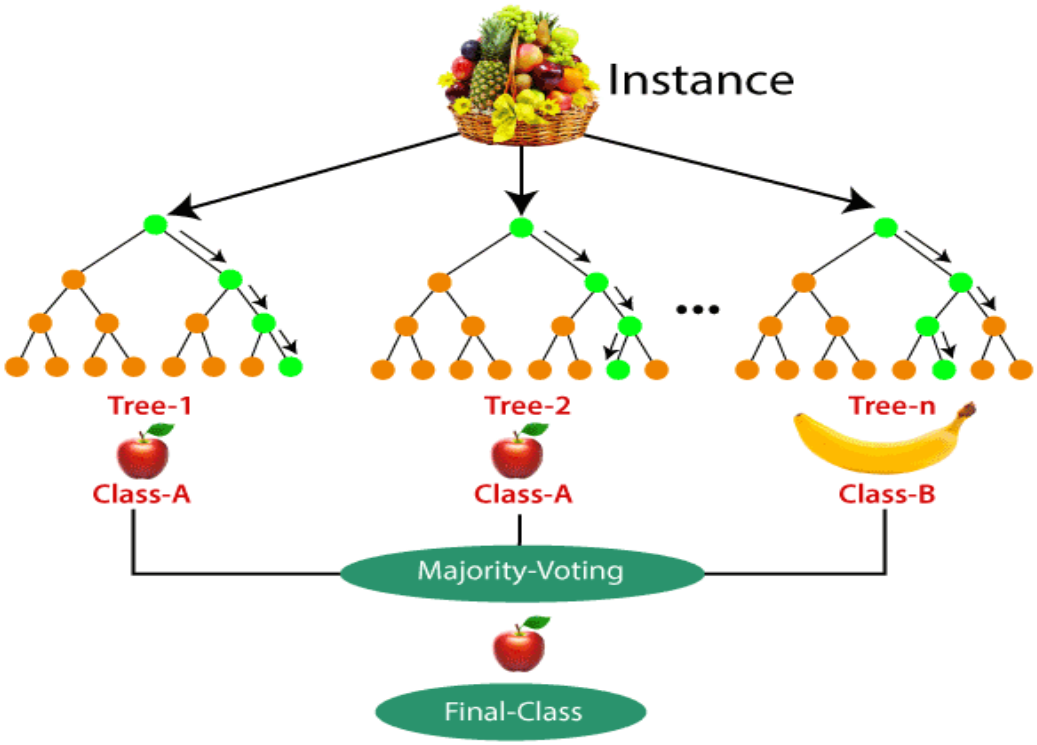
Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



Results and Experimentation

The hardware used to implement this program is a laptop with i5 processor, 500 GB Memory and 4 GB Ram. The Software used for this implementation is Anaconda, and the language used is Python. The dataset used in this program is taken from Instagram online repository. The result of this technique is obtained and measured using the following metrics.

- Confusion Matrix

In the area of machine learning, confusion matrix solves the statistical classification problem. Confusion Matrix is often used to discuss about the functioning of the classification system of an algorithm on a collection of testing data that is defined for the true positive values. It is also called as an error matrix. This allows ambiguity between groups to identify easily, e.g. one class is often mis labelled as another. Maximum performance measurements are obtained from the confusion matrix.

Class 1 Predicted Class 2 Predicted
Class 1
Actual TP FN
Class 2 Actual FP TN

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2Actual	FP	TN

Here,

• Class 1: Positive

• Class 2: Negative

Description of Terms

• Positive (P): positive values (for instance: is a ball).

• Negative (N): If the values are not positive (for example: is not a ball). • True Positive (TP): If the prediction is positive, but the observed values are even positive.

• False Negative (FN): Examined as positive, and the predicted value is negative.

• True Negative (TN): If the examined values are negative, but predicted as negative.

• False Positive (FP): Examined values are negative, and predicted as positive. Accuracy for detecting fake accounts can be obtained by using TP, TN, FP and FN from Eqn. (3).

Accuracy Prediction = TP+TN/TP+TN+FP+FN (3)

AUC, ROC curve is an efficiency calculation at various threshold rates for the classification issues. ROC is a curve of likelihood and AUC is the degree of separability factor. This shows the design value that can differentiate between classes. The model predicts accurately as 0's and 1's if the AUC is higher and it distinguishes better between accounts as fake or real. Based on the techniques described before, we assessed our strategies with ML classifiers, which includes Random Forest, Logistic Regression. In this experiment for testing the model we selected the datasets from Instagram, these datasets are pre-processed and split into 80:20 cross-validation process this split data is used for testing and training the algorithms. We applied our technique to the testing data as per the analysis Random Forest performed well with high accuracy (66%). By applying Z-Score Normalization SVM and Logistic regression, accuracy rate had been increased.

To calculate the performance of the classifier, we generated a ROC curve, based on true positive and false Positive Rate. The rank allocated to each account is based on the results of a particular classifier, so we tested the reliability by concentrating on a different range of ranks to see if our method works better by the rank created by our classification system. We used the ROC curve which shows the cut-off for a test, whereas the best cut off has the highest true positive rate with a low false positive rate, as shown in Figure 3. ROC curve based on 80:20 split and account level classification worked well with RF by giving a 93% True Positive rate with both type of the normalization techniques, however, SVM and Logistic performance are very low in Min-Max normalization when compared to Z-Score. On the other hand, we even focused on AUC. It can provide an integrated probability of performance with comparison to all possible classification thresholds. AUC is the rate at which the model is ranked random positive more than a random negative.

3.4. Feasibility Study:

A feasibility study on fake account detection using machine learning and data science would involve evaluating the technical and practical aspects of implementing such a system. The following are some of the key factors to consider:

Data availability: The effectiveness of a machine learning model for detecting fake accounts depends on the quality and quantity of data available. The feasibility study should explore the sources and types of data that can be used for training the model.

Feature selection: The model should be trained on relevant features that can differentiate between real and fake accounts. The feasibility study should identify the most important features and evaluate the availability and quality of these features in the data.

Machine learning algorithms: There are several machine learning algorithms that can be used for fake account detection, such as decision trees, support vector machines, and neural networks. The feasibility study should evaluate the performance of these algorithms on the available data and select the most suitable one.

Data processing and storage: The data used for training the machine learning model should be processed and stored appropriately. The feasibility study should consider the computational resources required for data processing and the storage requirements for the model.

Accuracy and performance: The feasibility study should evaluate the accuracy and performance of the machine learning model for detecting fake accounts. This can be done through cross-validation or by comparing the model's predictions to a labelled dataset of real and fake accounts.

Integration with existing systems: The feasibility study should evaluate the ease of integrating the fake account detection system with existing systems, such as social media platforms or online marketplaces.

Privacy and security: The privacy and security implications of implementing a fake account detection system. The system should be designed to protect user data and prevent unauthorized access. In conclusion, this would include considerations of data availability, feature selection, machine learning algorithms, data processing and storage, accuracy and performance, integration with existing systems, and privacy and security.

CHAPTER-4

SYSTEM REQUIREMENT SPECIFICATION

4.SYSTEM REQUIREMENTS SPECIFICATION

4.1. Purpose, Scope, Definition:

Purpose: The purpose of fake account detection using machine learning and data science is to identify and remove fake accounts that are created for fraudulent purposes, such as spreading misinformation, conducting spamming activities, or engaging in other types of malicious behaviour.

Fake accounts can be used to manipulate public opinion, impersonate real people, engage in financial fraud, or conduct other types of cybercrime. Detecting fake accounts is therefore essential for maintaining the security, trustworthiness, and reliability of online platforms, including social media networks, online marketplaces, and other types of online services.

Machine learning and data science can be used to analyse large amounts of data and identify patterns that are indicative of fake accounts. These patterns can include abnormal account activity, network behaviour, content analysis, and other indicators of suspicious behaviour.

By identifying and removing fake accounts, machine learning and data science can help to prevent the spread of misinformation, protect users from cybercrime, and improve the overall quality and safety of online platforms. In addition, fake account detection can help to maintain the integrity of online communities and ensure that online interactions are based on trust, authenticity, and transparency.

Scope: The scope of fake account detection using machine learning and data science is broad and can be applied to various online platforms, including social media networks, e-commerce websites, online gaming communities, and other online services. The following are some examples of the scope of fake account detection:

1. Social media networks: Social media networks such as Facebook, Twitter, and Instagram are prime targets for fake account creation. Machine learning and data science can be used to analyse user activity, network behaviour, and content to detect fake accounts that are used for spamming, phishing, or spreading misinformation.

2. E-commerce websites: E-commerce websites such as Amazon, eBay, and Alibaba are vulnerable to fraudulent activities such as fake reviews, fake transactions, and money laundering. Machine learning and data science can be used to detect fake accounts that are used for these purposes by analysing user behaviour, transaction history, and product reviews.

3. Online gaming communities: Online gaming communities are also vulnerable to fake accounts that are used for cheating, hacking, or other malicious activities. Machine learning and data science can be used to analyse player behaviour, network activity, and game data to detect fake

accounts that are used for these purposes.

4. Other online services: Machine learning and data science can be applied to various other types of online services, such as job portals, dating websites, and community forums, to detect fake accounts that are used for spamming, phishing, or other types of fraudulent activities.

The scope of fake account detection using machine learning and data science is constantly expanding as new types of online platforms and services emerge. As online platforms become increasingly important for communication, commerce, and social interaction, the need for effective fake account detection becomes more critical.

Definition:

Fake account detection using machine learning and data science refers to the process of automatically identifying and flagging accounts that are not genuine and are created for fraudulent purposes, such as spreading misinformation, conducting spamming activities, or engaging in other types of malicious behaviour.

Machine learning and data science techniques are used to analyze various features of the accounts, such as account activity, network behaviour, and content analysis, to detect patterns that are indicative of fake accounts. These techniques can help to identify and remove fake accounts quickly and accurately, reducing the risk of fraudulent activities and maintaining the security and trustworthiness of online platforms.

Fake account detection using machine learning and data science is an important task in the field of cyber security and is relevant for various online platforms, including social media networks, online marketplaces, and other types of online services. It is an on going process that requires continuous monitoring and improvement to stay ahead of evolving threats and maintain the integrity of online communities.

4.2. Requirement Analysis:

System requirement is the ability of the system to meet the condition desired by the users. System requirement analysis is performed by grouping the needs into functional requirements and non-functional requirements.

4.2.1. Functional Requirements:

The following are some functional requirements of fake account detection using machine learning and data science:

Data collection: The system should be able to collect a sufficient amount of data from the online platform to identify fake accounts. This may include data such as user activity, network behaviour, content analysis, and other relevant features.

Data pre-processing: The system should be able to pre-process the collected data by cleaning, normalizing, and transforming it to a suitable format for analysis.

Feature extraction: The system should be able to extract relevant features from the pre-processed data that are indicative of fake accounts. This may include features such as account activity, network behaviour, and content analysis.

Machine learning algorithms: The system should be able to use machine learning algorithms, such as supervised learning, unsupervised learning, and deep learning, to analyse the extracted features and identify fake accounts.

Thresholds and rules: The system should be able to set thresholds and rules for identifying fake accounts based on the results of the machine learning algorithms.

Alerting and reporting: The system should be able to generate alerts and reports when a fake account is identified, indicating the reason for the account's detection and any actions taken as a result.

Integration: The system should be able to integrate with other cyber security tools and platforms to enhance its functionality and effectiveness.

Overall, the functional requirements of fake account detection using machine learning and data science should enable the system to accurately and efficiently identify fake accounts and prevent fraudulent activities on online platforms.

4.2.2. Non-Functional requirements:

The following are some non-functional requirements of fake account detection using machine learning and data science:

1. Performance: The system should be able to analyse a large amount of data quickly and efficiently to detect fake accounts in real-time. It should also be able to handle peak loads without compromising its performance.

2. Accuracy: The system should have a high degree of accuracy in detecting fake accounts to minimize false positives and false negatives. The accuracy should be measured against a reliable ground truth dataset.

3. Scalability: The system should be able to handle a growing number of users and data without impacting its performance or accuracy. It should be able to scale horizontally or vertically as needed.

4. Security: The system should have adequate security measures to protect the data collected, processed, and analysed from unauthorized access or disclosure. It should also have measures to prevent attacks on the system itself.

5. Usability: The system should be user-friendly and easy to use, even for non-technical users. It should have a clear and intuitive interface for configuring, monitoring, and managing the system.

6. Reliability: The system should be reliable and able to operate continuously without downtime or errors. It should have measures to prevent data loss or corruption in case of system failures.

7. Maintainability: The system should be easy to maintain and upgrade to keep up with evolving threats and new features. It should have a modular architecture and well-documented code for easy maintenance. Overall, the non-functional requirements of fake account detection using machine learning and data science should ensure that the system is perform ant, accurate, secure, scalable, usable, reliable, and maintainable, providing a reliable and effective defence against fake accounts on online platforms.

4.3. System Requirements:

HARDWARE REQUIREMENTS:

System	:	Intel i3 6 core.
Hard Disk	:	500 GB SSD.
Monitor	:	15'' LED
Input Devices	:	Keyboard, Mouse
Ram	:	8GB.

SOFTWARE REQUIREMENTS:

Operating system	:	Windows 10.
Coding Language	:	Python
Tool	:	Jupyter Notebook
Dataset	:	Kaggle

4.4. Software Description:

The software required for this Jupyter Notebook which is able to implement all machine learning and data science algorithms which are implemented.

Machine Learning:

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

IBM has a rich history with machine learning. One of its own, Arthur Samuel, is credited for coining the term, "machine learning" with his research (PDF, 481 KB) (link resides outside IBM) around the game of checkers. Robert Nealey, the self-proclaimed checkers master, played the game on an IBM 7094 computer in 1962, and he lost to the computer. Compared to what can be done today, this feat seems trivial, but it's considered a major milestone in the field of artificial intelligence.

Over the last couple of decades, the technological advances in storage and processing power have enabled some innovative products based on machine learning, such as Netflix's recommendation engine and self-driving cars.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them.

Machine learning algorithms are typically created using frameworks that accelerate solution development, such as Tensor Flow and PyTorch.

How machine learning works

UC Berkeley (link resides outside IBM) breaks out the learning system of a machine learning algorithm into three main parts.

A Decision Process: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabelled, your algorithm will produce an estimate about a pattern in the data.

An Error Function: An error function evaluates the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.

A Model Optimization Process: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this "evaluate and optimize" process, updating weights autonomously until a threshold of accuracy has been met.

Supervised machine learning

Supervised learning, also known as supervised machine learning, is defined by its use of labelled datasets to train algorithms to classify data or predict outcomes accurately. As input data is fed into the model, the model adjusts its weights until it has been fitted appropriately. This occurs as part of the cross-validation process to ensure that the model avoids over fitting or under fitting. Supervised learning helps organizations solve a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning include neural networks, naïve bays, linear regression, logistic regression, random forest, and support vector machine (SVM).

Unsupervised machine learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyse and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. This method's ability to discover similarities and differences in information make it ideal for exploratory data analysis, cross-selling strategies, customer segmentation, and image and pattern recognition. It's also used to reduce the number of features in a model through the process of dimensionality reduction. Principal component analysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include neural networks, k-means clustering, and probabilistic clustering methods.

Reinforcement machine learning

Reinforcement machine learning is a machine learning model that is similar to supervised learning, but the algorithm isn't trained using sample data. This model learns as it goes by using trial and error. A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem.

The IBM Watson® system that won the Jeopardy! challenge in 2011 is a good example. The system used reinforcement learning to learn when to attempt an answer (or question, as it were), which square to select on the board, and how much to wager—especially on daily doubles.

Learn more about reinforcement learning.

Common machine learning algorithms

A number of machine learning algorithms are commonly used. These include:

Neural networks: Neural networks simulate the way the human brain works, with a huge number of linked processing nodes. Neural networks are good at recognizing patterns and play an important role in applications including natural language translation, image recognition, speech recognition, and image creation.

Linear regression: This algorithm is used to predict numerical values, based on a linear relationship between different values. For example, the technique could be used to predict house prices based on historical data for the area.

Logistic regression: This supervised learning algorithm makes predictions for categorical response variables, such as"yes/no" answers to questions. It can be used for applications such as classifying spam and quality control on a production line.

Clustering: Using unsupervised learning, clustering algorithms can identify patterns in data so that it can be grouped. Computers can help data scientists by identifying differences between data items that humans have overlooked.

Decision trees: Decision trees can be used for both predicting numerical values (regression) and classifying data into categories. Decision trees use a branching sequence of linked decisions that can be represented with a tree diagram. One of the advantages of decision trees is that they are easy to validate and audit, unlike the black box of the neural network.

Fraud detection: Banks and other financial institutions can use machine learning to spot suspicious transactions. Supervised learning can train a model using information about known fraudulent transactions. Anomaly detection can identify transactions that look atypical and deserve further investigation.

Challenges of machine learning

As machine learning technology has developed, it has certainly made our lives easier. However, implementing machine learning in businesses has also raised a number of ethical concerns about AI technologies. Some of these include:

While this topic garners a lot of public attention, many researchers are not concerned with the idea of AI surpassing human intelligence in the near future. Technological singularity is also referred to as strong AI or super intelligence. Philosopher Nick Bo strum defines super intelligence as "any intellect that vastly outperforms the best human brains in practically every field, including scientific creativity, general wisdom, and social skills." Despite the fact that super intelligence is not imminent in society, the idea of it raises some interesting questions as we consider the use of autonomous systems, like self-

driving cars. It's unrealistic to think that a driverless car would never have an accident, but The jury is still out on this, but these are the types of ethical debates that are occurring as new, innovative AI technology develops.

Privacy

Privacy tends to be discussed in the context of data privacy, data protection, and data security. These concerns have allowed policymakers to make more strides in recent years. For example, in 2016, GDPR legislation was created to protect the personal data of people in the European Union and European Economic Area, giving individuals more control of their data. As a result, investments in security have become an increasing priority for businesses as they seek to eliminate any vulnerabilities and opportunities for surveillance, hacking, and cyber attacks.

Data Science:

Data science is the study of data to extract meaningful insights for business. It is a multidisciplinary approach that combines principles and practices from the fields of mathematics, statistics, artificial intelligence, and computer engineering to analyse large amounts of data.

Data science is important because it combines tools, methods, and technology to generate meaning from data. Modern organizations are inundated with data; there is a proliferation of devices that can automatically collect and store information. Online systems and payment portals capture more data in the fields of e-commerce, medicine, finance, and every other aspect of human life. We have text, audio, video, and image data available in vast quantities.

Data science is used to study data in four main ways:

1. Descriptive analysis

Descriptive analysis examines data to gain insights into what happened or what is happening in the data environment. It is characterized by data visualizations such as pie charts, bar charts, line graphs, tables, or generated narratives. For example, a flight booking service may record data like the number of tickets booked each day. Descriptive analysis will reveal booking spikes, booking slumps, and high-performing months for this service.

2. Diagnostic analysis

Diagnostic analysis is a deep-dive or detailed data examination to understand why something happened. It is characterized by techniques such as drill-down, data discovery, data mining, and correlations. Multiple data operations and transformations may be performed on a given data set to discover unique patterns in each of these techniques For example, the flight service might drill down

on a particularly high-performing month to better understand the booking spike. This may lead to the discovery that many customers visit a particular city to attend a monthly sporting event.

3. Predictive analysis

Predictive analysis uses historical data to make accurate forecasts about data patterns that may occur in the future. It is characterized by techniques such as machine learning, forecasting, pattern matching, and predictive modelling. In each of these techniques, computers are trained to reverse engineer causality connections in the data. For example, the flight service team might use data science to predict flight booking patterns for the coming year at the start of each year. The computer program or algorithm may look at past data and predict booking spikes for certain destinations in May.

4. Prescriptive analysis

Prescriptive analytics takes predictive data to the next level. It not only predicts what is likely to happen but also suggests an optimum response to that outcome. It can analyse the potential implications of different choices and recommend the best course of action. It uses graph analysis, simulation, complex event processing, neural networks, and recommendation engines from machine learning.

A data scientist could project booking outcomes for different levels of marketing spend on various marketing channels. These data forecasts would give the flight booking company greater confidence in their marketing decisions.

CHAPTER-5

SYSTEM DESIGN

5.SYSTEM DESIGN

5.1. System architecture:

A system architecture for fake account detection using machine learning and data science typically involves the following components:

1. Data collection and pre-processing: The first step is to collect data about user behaviour on the platform, such as account creation patterns, activity patterns, and profile information. This data needs to be pre-processed, which includes cleaning, transforming, and feature engineering to extract relevant features for the machine learning models.

2.Feature engineering: Feature engineering is the process of selecting and extracting the most important features from the collected data. Features can be categorical, numerical, or text-based, and can include user metadata, content metadata, social graph features, and behavioural features.

3. Machine learning models: Once the features are engineered, machine learning models can be trained on the data. The models can be trained using a supervised or unsupervised learning approach. For example, classification algorithms such as decision trees, logistic regression, or neural networks can be used to classify users as either real or fake.

4. Model evaluation and selection: The trained models need to be evaluated using various metrics such as accuracy, precision, recall, and F1 score. The best-performing model can then be selected and deployed in production.

5. Deployment and monitoring: The selected model can be deployed in a production environment, and it should be monitored regularly to ensure it is performing as expected. The system can be integrated with the platform's registration process to identify fake accounts in real-time.

6. Continuous improvement: The system should be continuously improved by re-evaluating the performance of the model and retraining the model on new data. This will help to ensure that the system remains effective at detecting fake accounts over time**.**

Overall, the system architecture for fake account detection using machine learning and data science is an iterative process that involves data collection, feature engineering, model training, and continuous improvement to ensure that the system remains effective over time.

5.2. Modules:

There are various modules and tools required for building a fake account detection system using machine learning and data science. Here are some of the essential modules:

1.Environment Setup:

- Ensure that you have the necessary environment set up with Python installed along with required libraries like pandas, scikit-learn, matplotlib, and seaborn.

2.Data Collection and Pre-processing:

- Data Collection: Kaggle
- **Data Manipulation tool**: Pandas, NumPy, Matplotlib, Seaborn
- **Scikit-learn:** for feature extraction, feature scaling, and feature selection

3.Training:

- The script will load the dataset, preprocess it (one-hot encoding of 'Profile_description'), split it into training and testing sets, and then train a Random Forest classfier using the training data.

4. Model Evaluation:

- After training, the model's performance is evaluated using accuracy and confusion matrix metrics. These metrics will help you understand how well the model is performing on unseen data.

5.Visualization:

- The script produces various visualization to help understand the data and the model's performance, including a confusion matrix, a bar plot showing the distribution of fake and original accounts, and a pie chart representing the same distribution.

6.Interpretation:

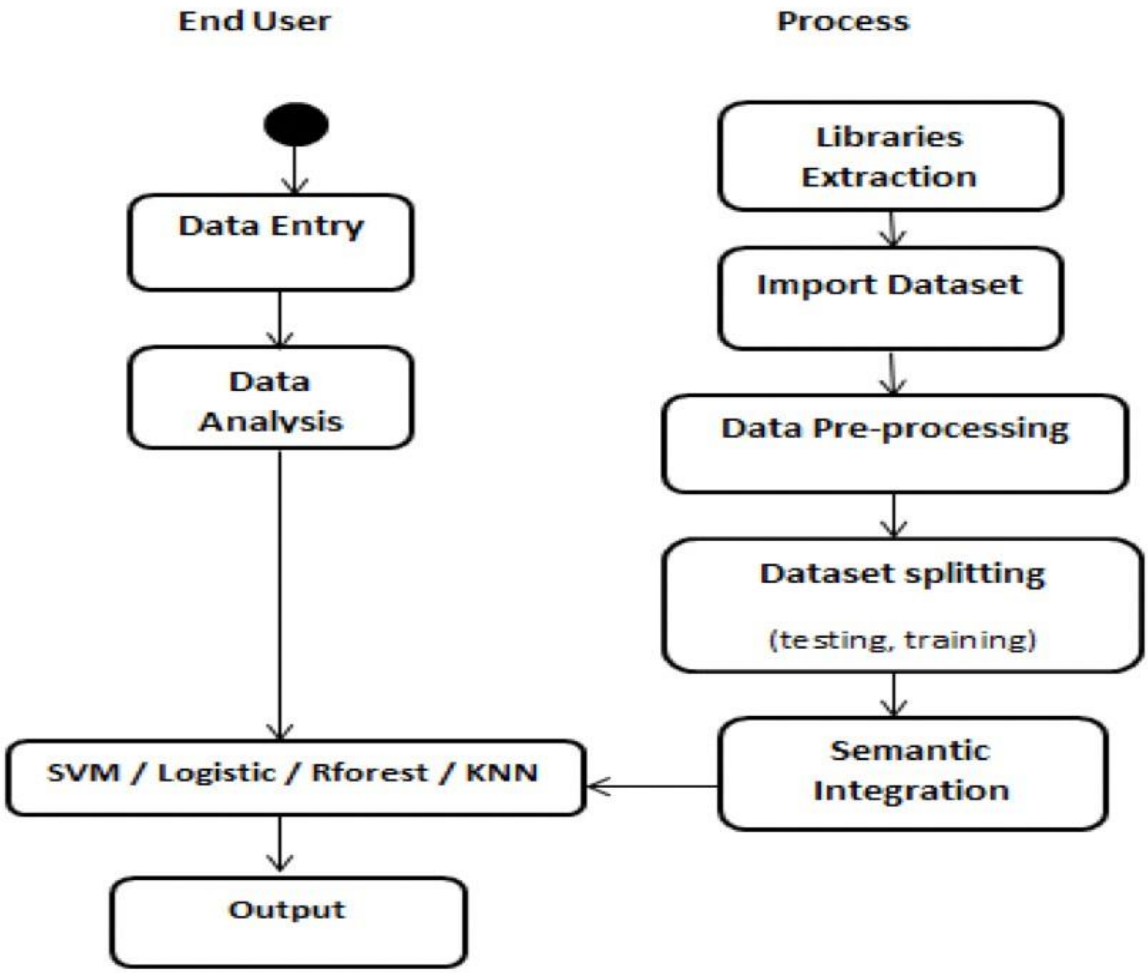
- Analyze the results to understand how the model is classifying fake and original accounts and whether there are any areas for improvement.

7.Deployment:

- Once satisfied with the model's performance, you can deploy it in various ways depending on your requirements. It could be integrated into a web application, used via APIs, or deployed as part of a larger machine learning pipeline.

In summary, these modules are essential for building a robust and scalable fake account detection system using machine learning and data science.

Architecture:



CHAPTER-6

IMPLEMENTATION

6.IMPLEMENTATION

6.1 STEPS OF IMPLEMENTATION:

The implementation of fake account detection using machine learning and data science involves the following steps:

1. Define the problem statement: Define the problem statement and the objectives of the fake account detection system. Identify the type of fake accounts that need to be detected and the types of data sources that need to be used.

2. Data collection and pre-processing: Collect data about user behaviour on the platform, such as account creation patterns, activity patterns, and profile information. Clean, transform, and engineer the collected data to extract relevant features for the machine learning models.

3. Feature engineering: Select and extract the most important features from the collected data. Features can be categorical, numerical, or text-based, and can include user metadata, content metadata, social graph features, and behavioural features.

4. Machine learning models: Train machine learning models on the data. The models can be trained using a supervised or unsupervised learning approach. For example, classification algorithms such as decision trees, logistic regression, or neural networks can be used to classify users as either real or fake.

5. Model evaluation and selection: Evaluate the trained models using various metrics such as accuracy, precision, recall, and F1 score. Select the best-performing model and tune the hyper parameters for better performance.

6. Deployment and monitoring: Deploy the selected model in a production environment. Integrate the system with the platform's registration process to identify fake accounts in real-time. Monitor the system regularly to ensure it is performing as expected.

7. Continuous improvement: Continuously improve the system by re-evaluating the performance of the model and retraining the model on new data. This will help to ensure that the system remains effective at detecting fake accounts over time.

8. Final testing and validation: Test and validate the system thoroughly before releasing it to the users. Run several experiments to ensure the accuracy of the system.

9. Maintenance and updates: Regularly maintain and update the system to keep it up-to-date with the latest technologies and to fix any bugs or issues.

Algorithm:

1.Random Forest Classifier:

- Random Forest (RF) is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is mode of the classes of the individual trees. It combines the predictions of multiple decision trees to omprove generalization and robustness over a single estimator.

2.One-Hot Encoding:

- This technique is used to convert categorical varibles into a form that can be provided to ML algorithms to do a better job in prediction. Each category is converted into a binary feature (0 or 1), making it suitable for ML algorithms to operate on.

3.Confusion Matrix:

- A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It helps in understanding the performance of the classfier in terms of true positives, true negatives, false positives, and false negatives.

4.Accuracy Score:

- Accuracy is one of the metrics used to evaluate classifications. However, accuracy alone might not be sufficient for imbalanced datasets.

5.Visualization:

- Visualizations such as bar plots, pie charts, and heatmaps are used to gain insights into the data distribution, model performance, and the class distribution, making it easier to interpret and communicate the results.

By understanding both the deployment process and the theoritical underpinnings of the code effectively utilize and improve the machine learning model for your specific application.

CHAPTER-7
TECHNOLOGY DESCRIPTION

7.TECHNOLOGY DESCRIPTION

7. 1. PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language. Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- Web development (server-side),
- Software development,
- Mathematics.
- System scripting

Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production- ready software development. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-orientated way or a functional way. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

7.2. History of Python:

Simple:

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

Easy to Learn:

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

Free and Open Source:

Python is an example of a FLOSS (Free/Libre and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

High-level Language:

When you write programs in Python, you never need to bother about the low- level details such as managing the memory used by your program, etc.

Portable:

Due to its open-source nature, Python has been ported to (i.e., changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features. You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, Vx Works, PlayStation, Sharp Taurus, Windows CE and Pocket! You can even use a platform like Ivy to create games for your computer and for iPhone, iPad, and Android.

Interpreted:

This requires a bit of explanation. A program written in a compiled language like C or C++ is converted from the source language i.e., C or C++ into a language that is spoken by your computer (binary code i.e., 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it. Python, on

the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called byte codes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

Object Oriented:

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object- oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

Extensible:

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

Embeddable: You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

Extensive Libraries:

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the Batteries Included philosophy of Python. Besides the standard library, there are various other high-quality libraries which you can find python package index.

CHAPTER-8

CODING

8.CODING

8.1 SOURCE CODE

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data_df = pd.read_csv('social_media_accounts.csv')

# One-hot encode the 'Profile_Description' column
X_onehot = pd.get_dummies(data_df['Profile_Description'])

# Concatenate the one-hot encoded features with the original features
X = data_df.drop(columns=['Username', 'Profile_Description', 'Label'])  # Features (attributes of social media accounts)
X = pd.concat([X, X_onehot], axis=1)

# Target variable (fake or original)
y = data_df['Label']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize a Random Forest classifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the classifier
clf.fit(X_train, y_train)
```

```
# Make predictions on the testing set
y_pred = clf.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print("Accuracy:", accuracy)
print("Confusion Matrix:")
print(conf_matrix)

# Count the number of fake and original accounts
count_labels = data_df['Label'].value_counts()

# Calculate percentages
total_accounts = count_labels.sum()
percentage_fake = (count_labels['Fake'] / total_accounts) * 100
percentage_original = (count_labels['Original'] / total_accounts) * 100

print(f"Percentage of Fake Accounts: {percentage_fake:.2f}%")
print(f"Percentage of Original Accounts: {percentage_original:.2f}%")

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", cbar=True, xticklabels=['Predicted
Fake', 'Predicted True'], yticklabels=['Actual Fake', 'Actual True'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```

```
# Plot the distribution of fake and original accounts
plt.bar(count_labels.index, count_labels.values, color=('#26B660', '#B72747'))
plt.xlabel('Account Type')
plt.ylabel('Number of Accounts')
plt.title('Distribution of Fake and Original Accounts')
plt.xticks(rotation=45)
plt.show()

# Plotting pie chart
plt.pie(count_labels.values, labels=count_labels.index, autopct='%1.1f%%', startangle=140)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title('Distribution of Fake and Original Accounts')
plt.legend()
plt.show()
```

CHAPTER-9

Output Screens (FORMS & REPORTS)

9.OUTOUT SCREENS (FORMS & REPORT)

10.1. Execution Screen Shorts

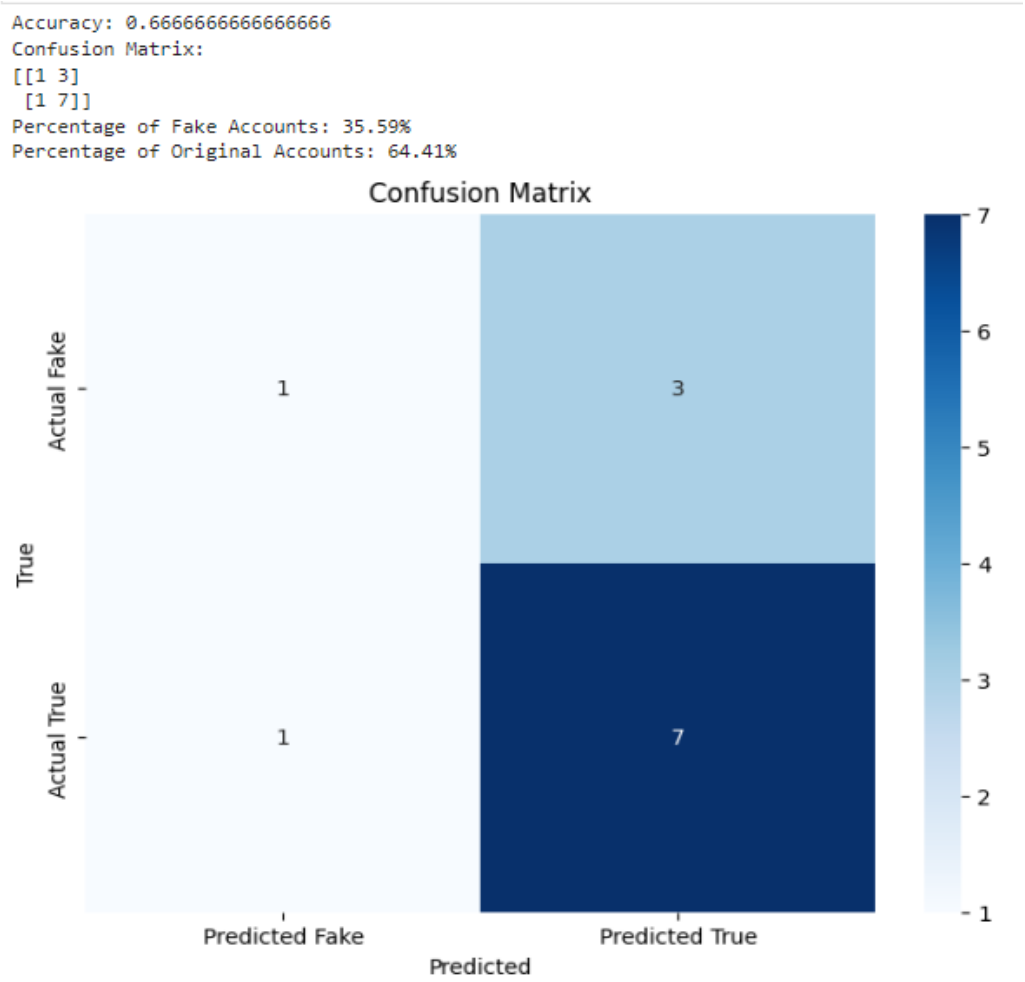


Figure 1: Accuracy Matrix

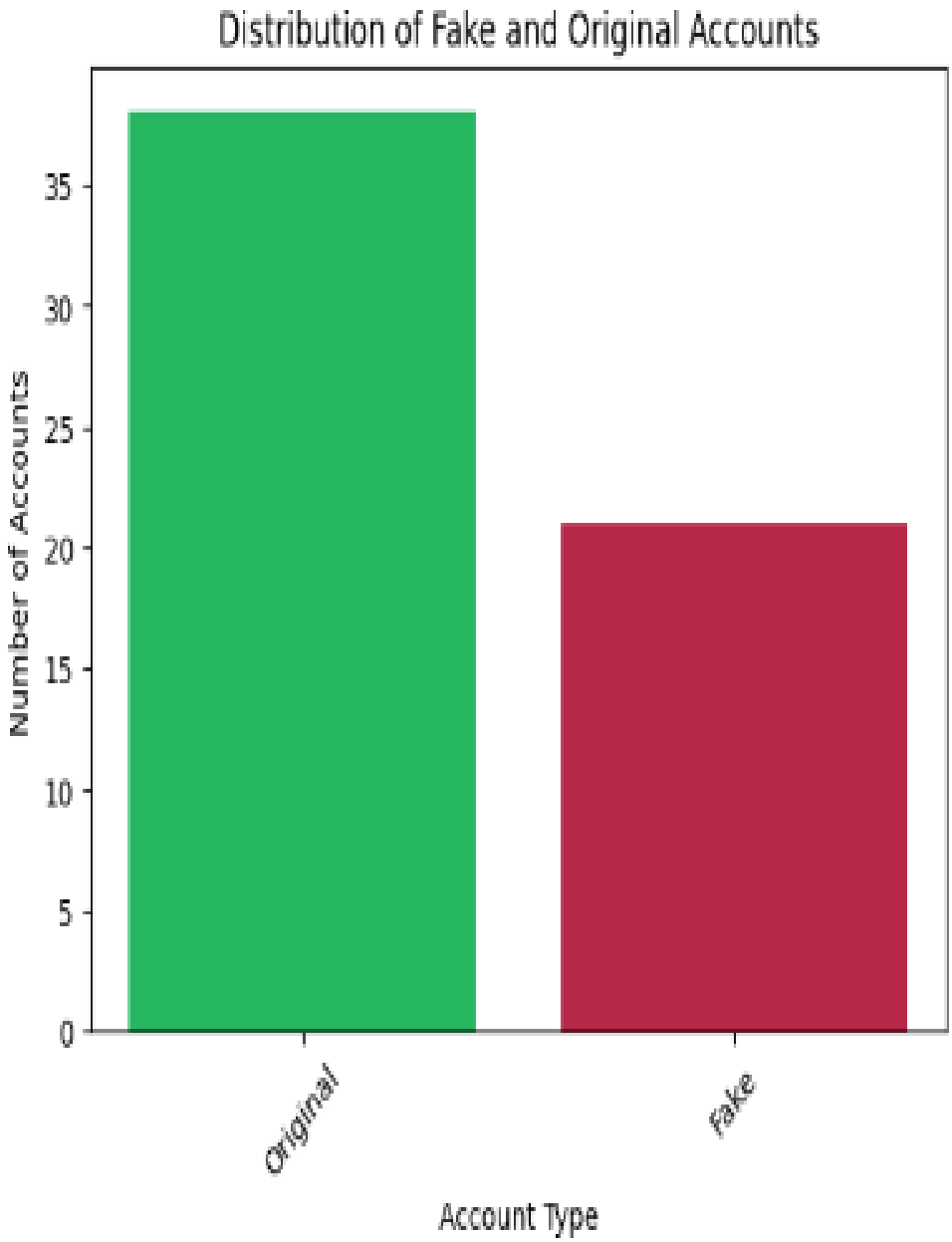


Figure 2: Bar Graph

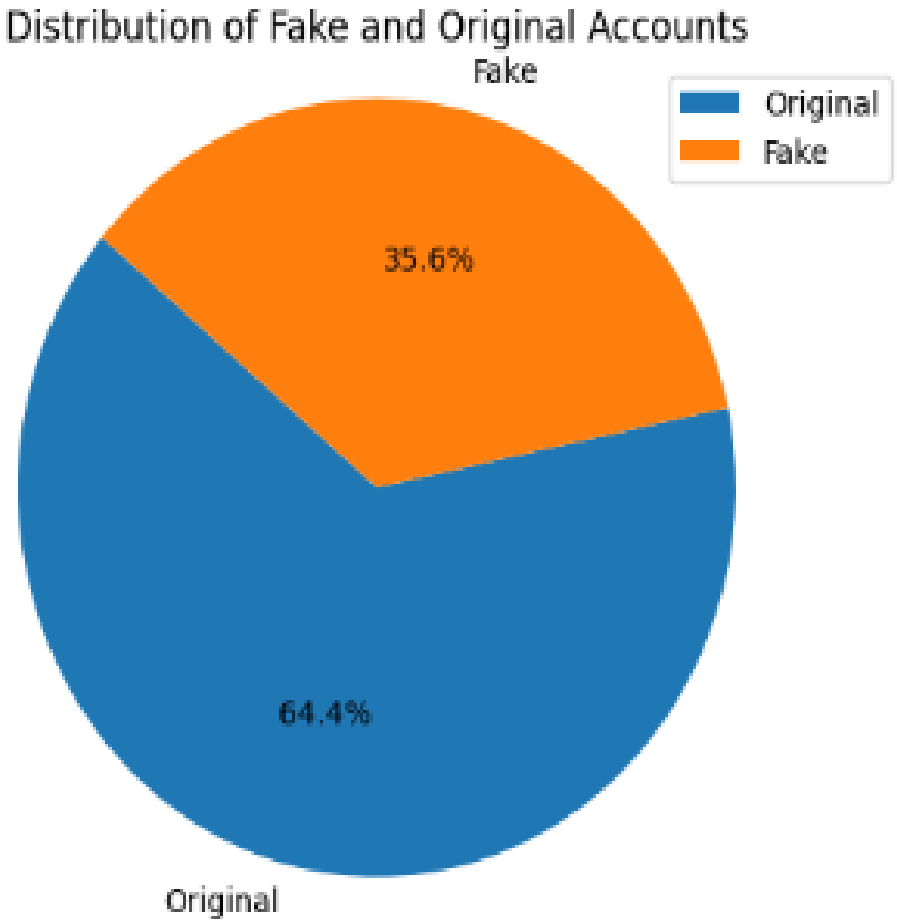


Figure 3: Pie Chart

CHAPTER 10

CONCLUSION

10.CONCLUSION

Fake account detection is an important problem for many online platforms, as fake accounts can be used for various fraudulent and malicious activities. Machine learning and data science can be used to build an effective fake account detection system, which can help online platforms to detect and remove fake accounts in real-time.

The process of building a fake account detection system using machine learning and data science involves several steps, including data collection, pre-processing, feature engineering, machine learning model training, and continuous improvement. The system can be deployed in a production environment, integrated with the platform's registration process, and monitored regularly to ensure that it is performing as expected.

The key advantage of using machine learning and data science for fake account detection is the ability to analyze large amounts of data and identify patterns that are indicative of fake accounts. This can help to identify fake accounts that may have otherwise gone undetected, and help online platforms to maintain a safe and secure environment for their users.

In conclusion, the use of machine learning and data science for fake account detection is a promising approach that can help to improve the security of online platforms and protect users from fraudulent and malicious activities**.**

CHAPTER-11

REFERENCE

11.REFERENCE

1.Cao, X., David, MF., Theodore, H.: Detecting Clusters of Fake Accounts in Online Social Networks. In: 8th ACM Workshop on Artificial Intelligence and Security, pp. 91-101 (2015).

2. Buket, E., Ozlem, A., Deniz, K., Cyhun, A.: Twitter Fake Account Detection. In: IEEE 2nd International Conference on Computer Science and Engineering, pp. 388-392 (2017).

3. Naman, S., Tushar, S., Abha, T., Tanupriya, C.: Detection of Fake Profile in Online Social Networks Using Machine Learning. In: IEEE International Conference on Advances in Computing and Communicaton Engineering. pp. 231-234 (2018).

4. Sarah, K., Neamat, E., Hoda, M. O .M.: Detecting Fake Accounts on Social Media. In: IEEE Intenational Conference on Big Data. pp. 3672-3681 (2018).

5. Yeh-Cheng, C., Shyhtsun, F .W.: FakeBuster: A Robust Fake Account Detection by Activiy Analysis. In: IEEE 9th International Symposium on Parallel Architectures, Algorithms and Programming. pp. 108-110 (2018).

6. Myo, MS., Nyein, NM.: Fake Accounts Detection on Twitter using Blacklist. In: IEEE 17th International Conference on Computer and Information and Information Science. pp. 562-566 (2018).

7. Qiang, C., Michael, S., Xiaowei, Y., Tiago P.: Aiding the Detection of Fake Accounts in Large Scale Social Online Services. In: 9th USENIX Conference on Networked Systems Design and Implementation. pp. 1-14 (2012).

8. Mauro, C., Radha, P., Macro, S.: Fakebook: Detecting Fake Profiles in Online Social Networks. In: IEEE International Conference on Advances in Social Networks Analysis and Mining. pp. 1071-1078 (2012).

9. Kaur , R., and Singh, S.: A survey of data mining and social network analysis based anomaly detection techniques. In: Egyptian informatics journal. pp.199–216 (2016).

10. Yazan, B., Dionysios, L., Georgos, S., Jorge, L., Jose, L., Matei, R., Konstatin, B., and Hassan, H.: Integro : Leveraging victim prediction for robust fake account detection in large scale osns ,Computers & Security. pp. 142–168 (2016).

11. Tsikerdekis, M., Zeadally , S.: Multiple Account Identity Deception Detection in Social Media using Non Verbal Behaviour. In: IEEE Transactions on Information Forensics and Security. 9(8), 1311-1321 (2014).

12. How fake news and hoaxes have tried to derail Jakarta's election. Internet draft(online).

13. Political advertising spending on Facebook between 2014 and 2018 Internet draft.