**A Project Report on**

### Use of Machine Learning to identify Fake Profiles

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

**Bachelor of Technology**

**In**

**Computer Science And Engineering**

Submitted by

K.HARSHITHA

(20H51A0537)

T.TANUJHA

(20H51A05F9)

Under the esteemed guidance of MR. M. SHIVA KUMAR

(Assistant Professor)



**Department of Computer Science And Engineering**

### CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

\*Approved by AICTE \*Affiliated to JNTUH \*NAAC Accredited with A+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

### 2023- 2024

**CMR COLLEGE OF ENGINEERING & TECHNOLOGY**

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



### CERTIFICATE

This is to certify that the Major Project report entitled **"Use of Machine Learning to Identify Fake Profiles"** being submitted by **K.Harshitha (20H51A0537), T.Tanujha (20H51A05F9)** in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out under my guidance and supervision**.**

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

|  |  |  |
| --- | --- | --- |
| **Dr. G. Ravi Kumar**  **Associate Professor** | **Dr. Siva Skandha Sanagala**  **Associate Professor and HOD** | **EXTERNAL EXAMINER** |
| **Dept. Of CSE** | **Dept. of CSE** |  |

### ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Dr. G. Ravi Kumar, Associate Professor,** Department of Branch Name for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**, Dr. Siva Skandha Sanagala,** Head of the Department of Dept Name, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr.** **Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Dept Name for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and Shri Ch Abhinav Reddy, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

K.Harshitha (20H51A0537)

T.Tanujha (20H51A05F9)

**CHAPTER**

**TABLE OF CONTENTS**

###### NO. TITLE PAGE NO.

|  |  |
| --- | --- |
| LIST OF FIGURES ABSTRACT  **1 INTRODUCTION** | i ii  1 |
| 1.1 Problem Statement | 2 |
| 1.2 Research Objective | 4 |
| 1.3 Project Scope and Limitations | 4 |
| **2 BACKGROUND WORK** | 6 |

* 1. Fake profile detection using XGB and GBM
     1. Introduction 7
     2. Merits, Demerits and Challenges 8
     3. Implementation of fake profile detection using XGB 9

and GBM

* 1. Deep neural network for detecting fake profiles in social media
     1. Introduction 11
     2. Merits, Demerits and Challenges 12
     3. Implementation of Deep neural network for detecting 13 fake profiles in social media

* 1. Fake profiles identification in online social networks using 16

machine learning and NLP

* + 1. Introduction 16
    2. Merits, Demerits and Challenges 17
    3. Implementation of Fake profiles identification in online 19

social networks using machine learning and NLP

1. PROPOSED SYSTEM 23
   1. Objective of Proposed Model 24
   2. [Algorithms Used for Proposed Model 25](#_TOC_250009)
   3. [Designing 26](#_TOC_250008)
      1. Dataset 26
      2. Training and Test Data 27
      3. Fake profiles Prediction Tool 28
      4. Decision Tree Accuracy 28
      5. [Prediction on Decision Tree Accuracy](#_TOC_250007) 28
      6. Architecture 29
      7. Data Flow 30
      8. UML Diagram 31
      9. [Sequence Diagram 32](#_TOC_250006)
      10. [Activity Diagram](#_TOC_250005) 33
   4. Stepwise Implementation and Code 34

4 RESULTS AND DISCUSSION 40

4.1 Performance metrics 44

5 [CONCLUSION 45](#_TOC_250004)

[5.1.Conclusion and Future Enhancement](#_TOC_250003) 46

[REFERENCES](#_TOC_250002) 49

[GITHUB LINK 50](#_TOC_250001)

[JOURNAL PUBLICATION 51](#_TOC_250000)

###### List of Figures

**FIGURE**

**NO. TITLE PAGE NO.**

|  |  |  |
| --- | --- | --- |
| Fig 2.1.4 | Implementation of XGB Model | 9 |
| Fig 2.2.4 | Implementation of Neural Network Model | 13 |
| Fig 2.3.4 | Implementation of NLP Model | 17 |
| Fig 3.2.1 | Decision tree Classifier | 20 |
| Fig 3.3.2 | Dataset | 21 |
| Fig 3.3.6 | Architecture | 24 |
| Fig 3.3.7 | Data flow | 25 |
| Fig 3.3.8 | UML Diagram | 26 |
| Fig 3.3.9 | Sequence Diagram | 27 |
| Fig 3.3.10 | Activity Diagram | 28 |
| Fig 4.0.1 | Result | 36 |
| Fig 4.0.2 | Web page | 37 |
| Fig 4.2.1 | Accuracy Formula | 39 |
| Fig 4.2.2 | Accuracy Comparison | 40 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**ABSTRACT**

Social media is presently a significant piece of our daily life. Currently more than the half of the world is an active user of the social media platforms. The ever - increasing popularity of these platforms has also given rise to a major issue which is the presence of fake accounts on them. These fake accounts serve the purpose of impersonating or cat-fishing other people. They have become an easy way to sell fake products and services to the customers. Also, the personal data of billions of people are at stake. These threats have made it essential to detect and deactivate the dummy accounts before any harm gets done. By the virtue of Machine Learning it has become easy to automatically detect millions of such accounts in a matter of seconds. In this project, we explore a deep learning model that can be used to classify a given account as real or fake. In the proposed work accuracy of the model is 93.63 percent.

People often use these dummy accounts to spread fake news which in the worst case can cause riot like conditions. Some people make use of fake accounts to spread hate which can be directed at certain race, religion, country or often at a particular person. This has increased the cases of cyber bullying leading to rise in the cases of depression and anxiety in teenagers. The growing threats of these fake accounts has made it necessary to take them down. With the number of fake accounts being in millions it has become impossible to manually detect them. Luckily the advancement in digital technology can benefit a lot in this situation. Methods like Machine Learning can help in making the stratification process a lot easier and accurate. This project involves use of deep learning model to classify social media accounts as genuine or fake. A Decision tree Classifier model is used to support the stratification.

# CHAPTER 1

## INTRODUCTION

**CHAPTER 1**

**INTRODUCTION**

**1.1.Problem Statement**

Social networking site is a website where each user has a profile and is able to keep up with friends, share updates and meet new stakeholders. The social networks online use the technology web2.0, which enables users to communicate. These social networking websites grow quickly and change the contacts between individuals. The online community brings together individuals with the same interests, facilitating user friendships. Social impact Everybody's social life has been linked to internet social networks in the current generation. These sites have dramatically altered our way of living in society. New friends and updates have become simpler to keep in touch with. Online social networks influence science, education, grassroots organization, work, company, etc.

Social networking sites have extensively used as a media of communication between people in day to day life. Users using this sites always share their information and daily activities which attract a number of people towards these sites. Increasing popularity of Facebook or Twitter or Twitter from the year 2006 to 2016 They allow the users to add friends and share various kind of information such as personal, social, economic, educational, political, business etc. . Moreover, they can also share photos, videos, and another day to day interaction. However, some people don’t use these sites with good objective. Therefore, they create fake accounts on social sites. Fake accounts do not have any real identity so we can call them as an Attacker. These attacker uses incorrect information or statistics about some real world person to create a fake account. Using theses fake accounts, attacker spread fake information which affects other users. To protect such sensitive data of users is one of the major challenges of social sites. There is a number of techniques in the field of machine learning that have been developed to detect fake accounts in social networking sites such as Neural Network (NN), Naive Bayes, Markov Model and Bayesian Network. In recent researches, it has been found that these techniques make available enhanced results to detect fake accounts.

Neural Network consists of many interconnected processing elements. It takes decisions just like a human brain. Support vector machines (SVM) is supervised machine learning techniques used for classification. It finds the hyper plane to classify the data. Neural network and SVM are able to accept a large amount of random data and suitable to detect the fake accounts on social networking sites based on various characteristics of accounts. Naive Bayes classifier is based on Bayes’ theorem. It predicts the probability that a given variable belongs to particular class.

**1.2.Research Objective**

The research objective for the project "Use of Machine Learning to Identify the Fake Profile " is to develop and evaluate deep learning models that can accurately identify and classify fake or fraudulent profiles on social media platforms.

These internet social networks have been studied by researchers to see their effect on the individuals. Teachers can readily reach their learners in a pleasant setting, educators now familiarize themselves with these websites that bring online classroom pages, do homework, talk, etc. which greatly enhances their schooling. In spite of all the advantages such social sites have their own disadvantages as well, in a certain way they pose threat to unvigilant individuals. Attacks such as phishing, spoofing, spamming, etc. have become really common. Measures should be taken to either control or detect such attacks. The individuals of a platform should be prudent enough to understand which people can be added to their social media accounts for this purpose the social media sites should provide certain filtering criteria which will in turn weed out the fake or suspicious accounts.

**1.3.Project Scope and Limitations**

##### Scope:

**1.Detection of various types of fake profiles:** The project aims to detect a wide range of fake profiles on digital platforms, including social media, dating sites, and other online communities.

**2.Text and Image Analysis:** The scope includes analyzing both textual content (profile descriptions, comments, posts, etc.) and images associated with profiles.

**3.Real-time and Batch Processing:** The project addresses real-time or batch processing to identify fake profiles as they are created or encountered on the platform

**4.Evaluation Metrics:** The project intends to use standard evaluation metrics like accuracy, precision, recall, and F1-score to assess the effectiveness of the deep neural network models

**1.4.Limitations:**

1. **Data Availability:** The effectiveness of the models depends on the availability and quality of training data. Limited or biased data can lead to model biases.
2. **False Positives and Negatives:** No model is perfect, and there will be false positives (genuine profiles misclassified as fake) and false negatives (fake profiles not detected). The aim is to minimize these but not eliminate them entirely.
3. **Evolving Fake profiles techniques:** As fake profile creation techniques evolve, the model's performance may degrade over time. Continuous monitoring and updates are necessary.
4. **Privacy Concerns:** The project should consider privacy concerns related to profile analysis. Striking the right balance between detection and privacy is crucial.
5. **Computational Resources:** Despite project should consider the computational resources required for training and real-time processing.

# CHAPTER 2

## BACKGROUND WORK

**CHAPTER 2**

**BACKGROUND WORK**

**2.1.Fake profile detection using XGB and GBM**

**2.1.1.Introduction**

Nowadays, Online Social Media is dominating the world in several ways. Day by day the number of users using social media is increasing drastically. The main advantage of online social media is that we can connect to people easily and communicate with them in a better way. This provided a new way of a potential attack, such as fake identity, false information, etc. A recent survey suggest that the number of accounts present in the social media is much greater than the users using it. This suggest that fake accounts have been increased in the recent years. Online social media providers face difficulty in identifying these fake accounts. The need for identifying these fake accounts is that social media is flooded with false information, advertisements, etc.

The gradient boosting algorithm is similar to the random forest algorithm in that it relies heavily on decision trees. We have modified the way we find fake accounts, using new approaches to locate them. Spam commenting, interaction rate, and artificial behavior are some of the techniques used. The gradient boosting algorithm uses these inputs to build decision trees, which are then used in the gradient boosting algorithm. Even if some inputs are missing, this algorithm produces a result. This is the primary reason for using this algorithm.

##### 2.1.2.Merits, Demerits and Challenges:

**Merits:**

* High Accuracy: XGBoost and GBM are known for their high predictive accuracy. They can effectively distinguish between genuine and fake profiles, reducing false positives and false negatives.
* Feature Importance: These algorithms provide insights into feature importance, which helps in understanding which attributes contribute most to fake profile detection.
* Scalability: XGBoost and GBM can handle large datasets and perform well in real time, making them suitable for platforms with high user volumes.

**Demerits:**

* Complexity: XGBoost and GBM models are more complex than simpler algorithms, which can make them harder to set up and tune correctly.
* Resource Intensive: Training XGBoost and GBM models can be computationally intensive, requiring substantial resources in terms of memory and processing power.
* Data Preparation: High-quality, well-structured data is essential for these models to perform well. Data preprocessing can be time-consuming

**Challenges**:

* There was a lack of a gold standard public dataset for analysis, thus they had to use active learning. The use of Extreme Gradient Boosting to detect fraudulent accounts is still relatively new and on the rise.
* There are numerous branches to investigate. As previously stated, they did not perform deep hyperparameter tuning in suggested strategy or trials.
* Tuning hyperparameters is both expensive and time-consuming. Finding the optimal collection of parameters might be difficult.

**2.1.3.Implementation:**

To implement the fake profile detection using XGB and GBM it uses gradient boost and extreme gradient boosting algorithms to detect fake accounts. The technologies we have used are Python and Python’s Standard libraries like Numpy, Pandas, Matplotlib, Scipy and Sk learn.

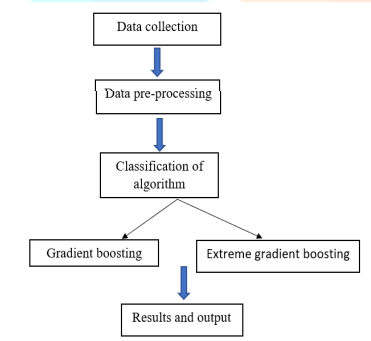


Fig 2.1.4 Implementation of XGB Model

1. Uploading the data: A collection of instances is a dataset and when working with machine learning methods we typically need a few datasets for different purposes.

• Training Dataset: A dataset that we feed into our machine learning algorithm to train our model.

• Testing Dataset: A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset.

1. Dataset pre-processing: It is an important step to detect fake account. In this step data is processed in an appropriate form which can be inputted for detection process. the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.
2. Testing with Gradient boosting and Extreme gradient boosting methods: To produce final predictions, a Gradient Boosting Machine (GBM) combines predictions from multiple decision

trees.

1. Another widely used boosting algorithm is XGBoost (Extreme Gradient Boosting). In reality, XGBoost is just a tweaked GBM algorithm! XGBoost follows the same steps as GBM in terms of operation. XGBoost builds trees in a sequential fashion, attempting to fix previous trees errors.
2. Input :

Train Data = The labeled training set (70%)

Validation Data = The validation dataset (10%)

Test Data = Unlabeled dataset (20%)

1. Output:

Predictions = prediction from classifiers used.

Validation Data is used to validate the classifier predictions

1. Load Train Data

2. for all instances in Train Data

3. for each feature matrix fed to the CLASSIFIER [ LR, RANDOM FOREST XGB, ADB, GBM]

4. train classifier

5. Accuracy, precision = PEDICTION metrices

6. RESULT COMPARISON

#### 2.2. Deep neural network for detecting fake profiles in social media:

**2.2.1ntroduction**

Social media’s growth can potentially raise people’s social evaluation and popularity. In particular, social network users may gain popularity by amassing many likes, follows, and remarks. On the other hand, establishing fake profiles is much too simple, and such accounts can be purchased online at little cost. For instance, purchasing followers and comments on social media platforms such as Facebook and Twitter may be done more easily on the internet [[4](https://www.techscience.com/csse/v47n1/53039/html#ref-4)]. Analysis of activity changes is one of the most common techniques open social networking methods use to spot strange accounts. The activities that people engage in throughout time tend to shift and evolve. Therefore, the server can identify a potential scam account by monitoring for sudden changes in access patterns to the content and activity it requires. In case of unsuccessful identification, the deviant might fill the systems with fake information

Adding multiple hidden layers in a deep neural network allows it to learn more complex features and relationships in the data, leading to better performance in tasks such as image and speech recognition, natural language processing, and many others. However, deep neural networks can be more challenging to train and suffer from overfitting or vanishing gradients.

To overcome these challenges, researchers have developed various techniques such as regularization, normalization, dropout, and gradient clipping to improve the training of deep neural networks. Overall, deep neural networks have revolutionized the field of machine learning and are widely used in various applications, from computer vision to natural language processing to autonomous vehicles.

The following is a summary of the most important contributions that our study has made, notably in addressing the fake profile classification task:

1.A deep learning approach provided a unique way of identifying fraudulent accounts inside social networks.

2. Sixteen profile-based features to train models for fake account detection problems were determined.

3.They put the proposed deep model through its paces by conducting an exhaustive examination to acquire cutting-edge findings, particularly regarding detecting fraudulent profiles.

**2.2.2.Merits, Demerits and Challenges:**

**Merits**

1. High Accuracy: Deep neural networks (DNNs) have shown impressive accuracy in detecting fake profiles due to their ability to learn complex patterns.

2. Scalability: Once trained, DNN models can be scaled to handle large volumes of social media data efficiently.

3. Automation: DNN-based systems can automate the detection process, reducing the need for manual intervention.

4. Adaptability: DNNs can adapt to evolving tactics used by fake profiles, making them robust against new types of fraud.

**Demerits:**

1. Data Quality: DNNs heavily rely on the quality of training data, and if the data is biased or incomplete, it can lead to inaccurate results.

2. Interpretability: DNNs are often considered black-box models, meaning it can be challenging to interpret how they make decisions, which may be a concern for transparency and accountability.

3. Computational Resources: Training and running DNN models can require significant computational resources, which may be a limitation for organizations with limited resources.

4. False Positives: There's a risk of false positives where legitimate profiles are incorrectly flagged as fake, leading to potential user dissatisfaction.

**Challenges:**

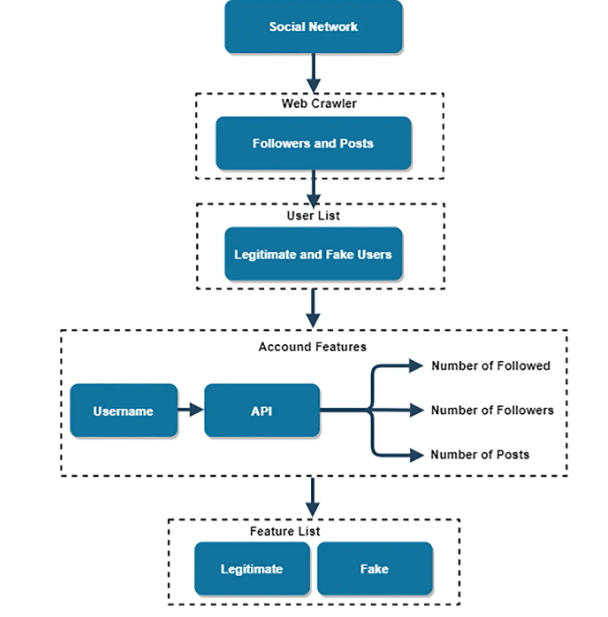
1. Data Collection: Gathering labeled data for training DNNs can be challenging, especially for detecting sophisticated fake profiles.

2. Adversarial Attacks: Fake profiles may employ adversarial techniques to evade detection, posing a challenge for the effectiveness of DNN-based detection systems.

3. Privacy Concerns: Analyzing social media data raises privacy concerns, and ensuring compliance with regulations such as GDPR can be challenging.

4. Continuous Learning: DNN models need to be continuously updated and retrained to adapt to new tactics employed by fake profiles, requiring ongoing maintenance and monitoring.

**2.2.3.Implementation**



**Fig 2.2.4 Implementation of Neural Networks Model**

**1.Data collection:** they have developed more sophisticated types of record crawlers, one for reaching typical clients and another for finding unusual ones. The daily user crawler used the find feature on Facebook to locate ordinary users to be included in the list of everyday users in the dataset. The Explore section of Facebook displays recently published photos and videos that capture the attention of other users, indicating that the content shared on Facebook is, for the most part, genuine and authentic. In addition, to find and harvest fake customers on Facebook, an advanced crawler was initially utilized to acquire fake customer identifications (IDs) through the follower listings of customers who considered a wide variety of fake customers of their follower listings. Secondly, another system that allows the manual test of all false archived customers included in the dataset was developed.

**2. Feature preparation:** The selected essential functions may be indexed in the dataset in the following manner:

1.The whole range of activities by the account.

2. Remember that the account is vital to the followers.

3.After taking into consideration the history.

4.The number of digits that may be found in the account username.

Regardless of whether or not the account is private, none of the functions are connected to the user’s media; hence, the set of restrictions does not violate the user’s account privacy. In this day and age of fake debts, some debts are manufactured by adding various numbers to the same name, which is why account usernames must have a diverse range of digits. It’s possible to make out the several ways the digits are distributed.

**3.Classification:** A deep learning model in binary classification has been applied to distinguish between dummy and valid profiles. Converting the array into binary tensors is a crucial step before supplying it with input, the matrix, into the network to facilitate adaptation. The suggested CNN begins by collecting input data in the first layer and crossing it onto hidden layers as its first operational phase. It employs social network characteristics as its input matrix at the beginning of the process.

To down sample the input feature map, reducing its size while retaining the essential information, MaxPooling was applied. This operation effectively reduces the spatial resolution of the feature map but preserves the essential features by selecting the maximum value. Max pooling has several benefits, including reducing the computational complexity of the network, making it more robust to small translations in the input, and helping to prevent overfitting by enforcing a form of regularization.

**4.Result:** By performing all the above steps the final detection of fake profiles is done.

**2.3.Fake profiles identification in online social networks using machine learning and NLP**

**2.3.1.Introduction:**

Social networking has end up a well-known recreation within the web at present, attracting hundreds of thousands of users, spending billions of minutes on such services. Online Social network (OSN) services variety from social interactions-based platforms similar to Facebook or MySpace, to understanding dissemination-centric platforms reminiscent of twitter or Google Buzz, to Social interaction characteristic brought to present systems such as Flicker. The opposite hand, enhancing security concerns and protecting the OSN privateness still signify a most important bottleneck and viewed mission. When making use of Social network’s (SN’s), one of a kind men and women share one-of-a-kind quantities of their private understanding. Having our individual know-how entirely or in part uncovered to the general public, makes us excellent targets for unique types of assaults, the worst of which could be identification theft.

Identity theft happens when any individual uses character’s expertise for a private attain or purpose. During the earlier years, online identification theft has been a primary problem considering it affected millions of people’s worldwide. Victims of identification theft may suffer unique types of penalties; for illustration, they would lose time/cash, get dispatched to reformatory, get their public image ruined, or have their relationships with associates and loved ones damaged. At present, the vast majority of SN’s does no longer verifies ordinary users‟ debts and has very susceptible privateness and safety policies. In fact, most SN’s applications default their settings to minimal privateness; and consequently, SN’s became a best platform for fraud and abuse. Social Networking offerings have facilitated identity theft and Impersonation attacks for serious as good as naive attackers. To make things worse, users are required to furnish correct understanding to set up an account in Social Networking web sites. Easy monitoring of what customers share on-line would lead to catastrophic losses, let alone, if such bills had been hacked.

**2.3.2. Merits, Demerits and challenges**

**Merits:**

1. Improved Detection Accuracy: Machine learning and NLP techniques can enhance the accuracy of identifying fake profiles by analyzing linguistic patterns and behavioral traits.

2. Scalability: Once trained, the model can scale to process large volumes of social network data efficiently.

3. Real-time Monitoring: With automation, the system can continuously monitor social networks for fake profiles, providing timely detection and response.

4. Reduced Human Effort: Automation reduces the need for manual inspection, saving time and resources for social network moderators.

5. Adaptability: The system can adapt to evolving tactics used by malicious actors to create fake profiles, improving detection capabilities over time.

**Demerits:**

1.False Positives/Negatives: Machine learning models may produce false positives (legitimate profiles misclassified as fake) or false negatives (fake profiles not detected), impacting the system's reliability.

2.Data Privacy Concerns: Collecting and analyzing user data for profile identification raises privacy concerns and may face regulatory scrutiny.

3.Bias in Training Data: Biases present in the training data can lead to biased predictions, potentially discriminating against certain demographics.

4.Adversarial Attacks: Malicious actors may attempt to manipulate the system's algorithms to evade detection, requiring constant updates and countermeasures.

5.Resource Intensive: Training and maintaining machine learning models for fake profile detection can require significant computational resources and expertise.

**Challenges:**

1.Data Quality: Ensuring the quality and reliability of the training data is crucial for the effectiveness of the model.

2.Feature Engineering: Identifying relevant features from text and user behavior data and engineering them effectively for model input can be challenging.

3.Model Interpretability: Understanding how the model makes decisions is essential for trust and accountability but can be difficult with complex machine learning algorithms.

4.Generalization: Ensuring the model performs well across different social networks and user demographics requires robust generalization techniques.

5.Ethical Considerations: Balancing the benefits of fake profile detection with potential privacy violations and discrimination concerns requires careful ethical considerations throughout the project.

**2.3.3. Implementation**

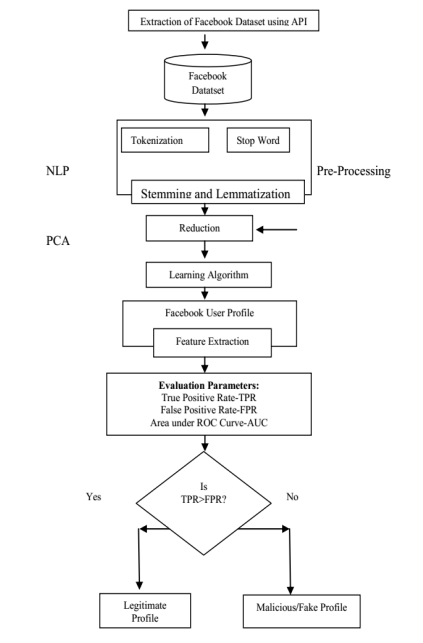


Fig 2.3.4 Implementation of NLP Model

1.The presented process used Facebook profile to notice false profiles. The working method of the proposed procedure includes three principal phases

1. NLP Pre-processing

2. Principal Component Analysis(PCA)

3. Learning Algorithms

1. NLP Pre-Processing Text pre-processing is an essential a part of any NLP method and the significance of the NLP pre-processing are

1. To minimize indexing (or knowledge) records dimension of the textual content records

i. Stop words bills 20-30% of total phrase counts in a special textual content records

ii. Stemming may just diminish indexing size as much as forty- 50%

2. To make stronger the efficiency and effectiveness of the IR method

i. Stop words aren't valuable for shopping or textual content mining and so they may just confuse the retrieval system

ii. Stemming used for matching the similar words in a text record

**2. Tokenization**: Tokenization is the process of breaking a circulate of textual content into phrases, phrases, symbols, or different significant factors called tokens .The aim of the tokenization is the exploration of the phrases in a sentence. The list of tokens turns into input for further processing akin to parsing or textual content mining. Tokenization is valuable both in linguistics (where it's a form of textual content segmentation), and in laptop science, the place it forms a part of lexical analysis. Textual knowledge is simplest a block of characters at the starting. All strategies in know-how retrieval require the words of the data set. For that reason, the requirement for a parser is a tokenization of records. This might be sound trivial because the text is already saved in computing devicereadable codecs. However, some problems are nonetheless left, like the removing of punctuation marks. Different characters like brackets, hyphens, and so on require processing as well. **3.Stop word Removal:** Stop phrases are very more often than not used fashioned phrases like ‘and’, ‘are’, ‘this’ etc. They don't seem to be useful in classification of records. So they must be removed. However, the development of such stop phrases record is problematic and inconsistent between textual sources. This process also reduces the text knowledge and improves the approach performance. Each textual content report offers with these phrases which are not vital for text mining applications.

**4.Stemming and Lemmatization:** The aim of both stemming as well as lemmatization is to scale down inflectional types & mostly derivationally associated varieties of a phrase to a fashioned base kind. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of accomplishing this goal accurately more often than not, and quite often involves the removal of derivational affixes. Lemmatization often refers to doing matters competently with the usage of a vocabulary and morphological analysis of phrases, in most cases aiming to eliminate inflectional endings only and to come back the base or dictionary type of a word, which is often called the lemma.

**5.Principal Component Analysis** (PCA) Principal Component Analysis purpose is to extract the fundamental understanding from the table, to symbolize it as a suite of new orthogonal variables known as major accessories, and to show the sample of similarity of the observations and of the variables as elements in maps.

**6.Learning Algorithms** In this proposed system we are using two machine learning algorithms named as Support Vector Machine (SVM) and naïve Bayes algorithms.

**7.Support Vector Machine (SVM)** An SVM classifies information by means of finding the exceptional hyperplane that separates all information facets of 1 type from those of the other classification. The best hyperplane for an SVM method that the one with the biggest line between the two classes. An SVM classifies data through discovering the exceptional hyperplane that separates all knowledge facets of one category from those of the other class. The help vectors are the info aspects which are closest to the keeping apart hyperplane.

**8. Naïve Bayes** Naive Bayes algorithm is the algorithm that learns the chance of an object with designated features belonging to a unique crew/category. In brief, it's a probabilistic classifier. The Naive Bayes algorithm is called "naive" on account that it makes the belief that the occurrence of a distinct feature is independent of the prevalence of other aspects. For illustration, if we're looking to determine false profiles based on its time, date of publication or posts, language and geoposition. Even if these points depend upon each and every different or on the presence of the other facets, all of these properties in my view contribute to the probability that the false profile

# CHAPTER 3

## PROPOSED SYSTEM

**CHAPTER 3**

**PROPOSED SYSTEM**

##### Objective of Proposed Model

The primary objective of this project is to develop and evaluate machine learning models that can accurately predict the presence of fake profiles on Instagram . The specific goals include:

* **Dataset Exploration:** Analyze a dataset containing information of the existing social media Profiles which were verified whether they are fake or not. The attributes are profile pic, username, nums/length of username, URL, followers, following, etc.
* **Decision tree Classifier Development**: Build a model to predict Fake profiles. Assess the model's accuracy using appropriate evaluation metrics.
* **Model Predictions:** Giving the input to the model about the account we want to know whether it is fake or not. Metrics of the model are obtained and comparison with rest of the models is done.

##### Algorithms Used for Proposed Model

###### Decision Tree Classifier

* A decision tree classifier is a type of machine learning algorithm used for classification tasks.
* It works by recursively partitioning the feature space into smaller regions based on the values of input features, ultimately creating a tree-like structure of decision rules. At each node of the tree, the algorithm selects the feature that best splits the data into classes, based on criteria like Gini impurity or information gain.
* This process continues until a stopping criterion is met, such as reaching a maximum tree depth or having nodes with only one class. Once trained, the decision tree can classify new instances by following the decision rules learned during training.
* Decision trees are easy to interpret and visualize, making them useful for understanding the decision-making process of the model
* In this project, the DTC is used to classify whether a given Profile is likely to be an fake or not.

**Gini Index:-**When building a decision tree using the CART (Classification and Regression Tree) technique, the Gini index is a measure of purity or impurity. It is better to choose an attribute with a low Gini index over one with a high index. The CART algorithm only produces binary splits, and it does so by utilizing the Gini index. The formula follows.

Gini Index= 1- ∑jPj2

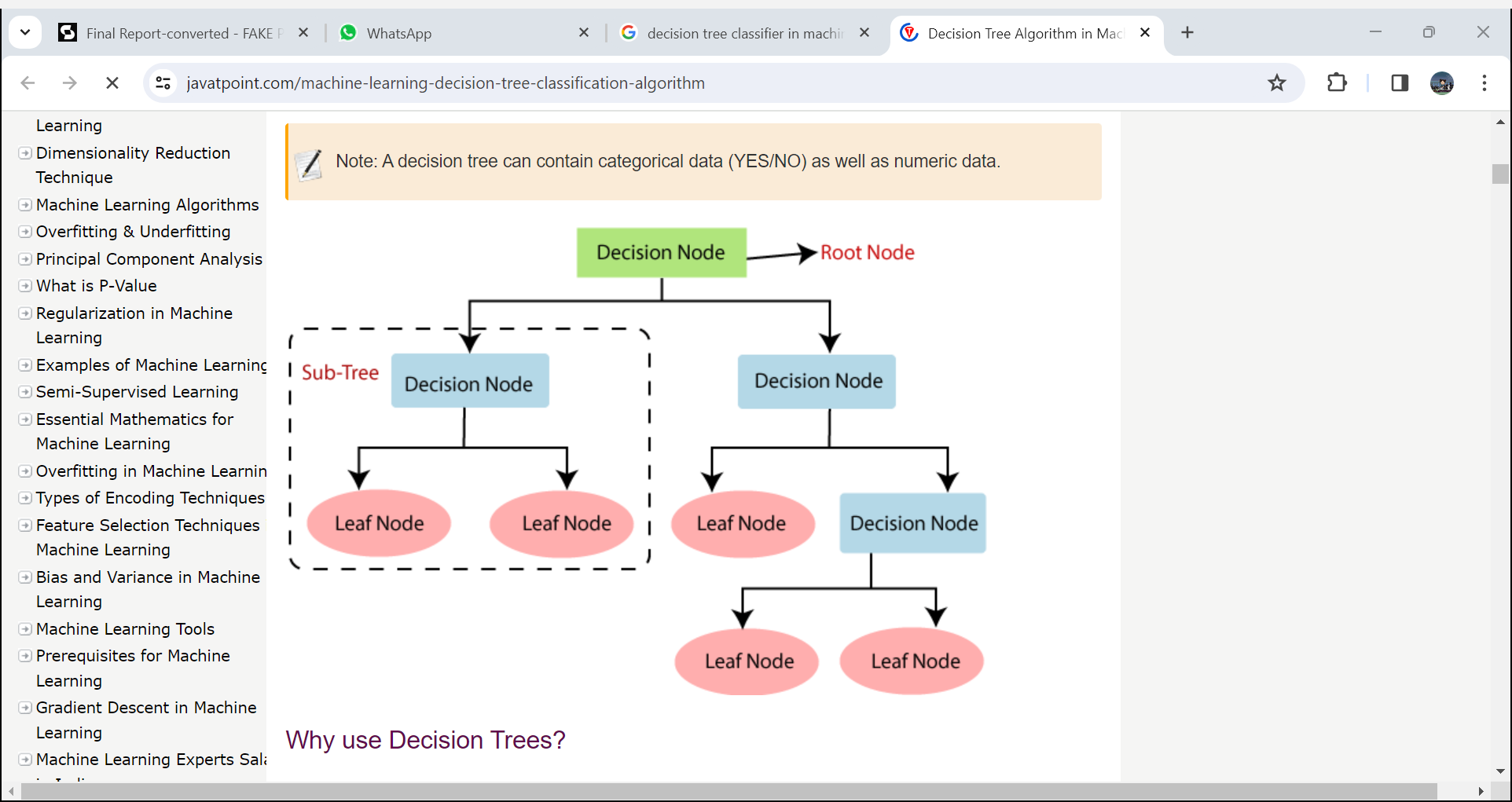


Fig 3.2.1 Decision tree Classifier

##### Designing

##### Dataset

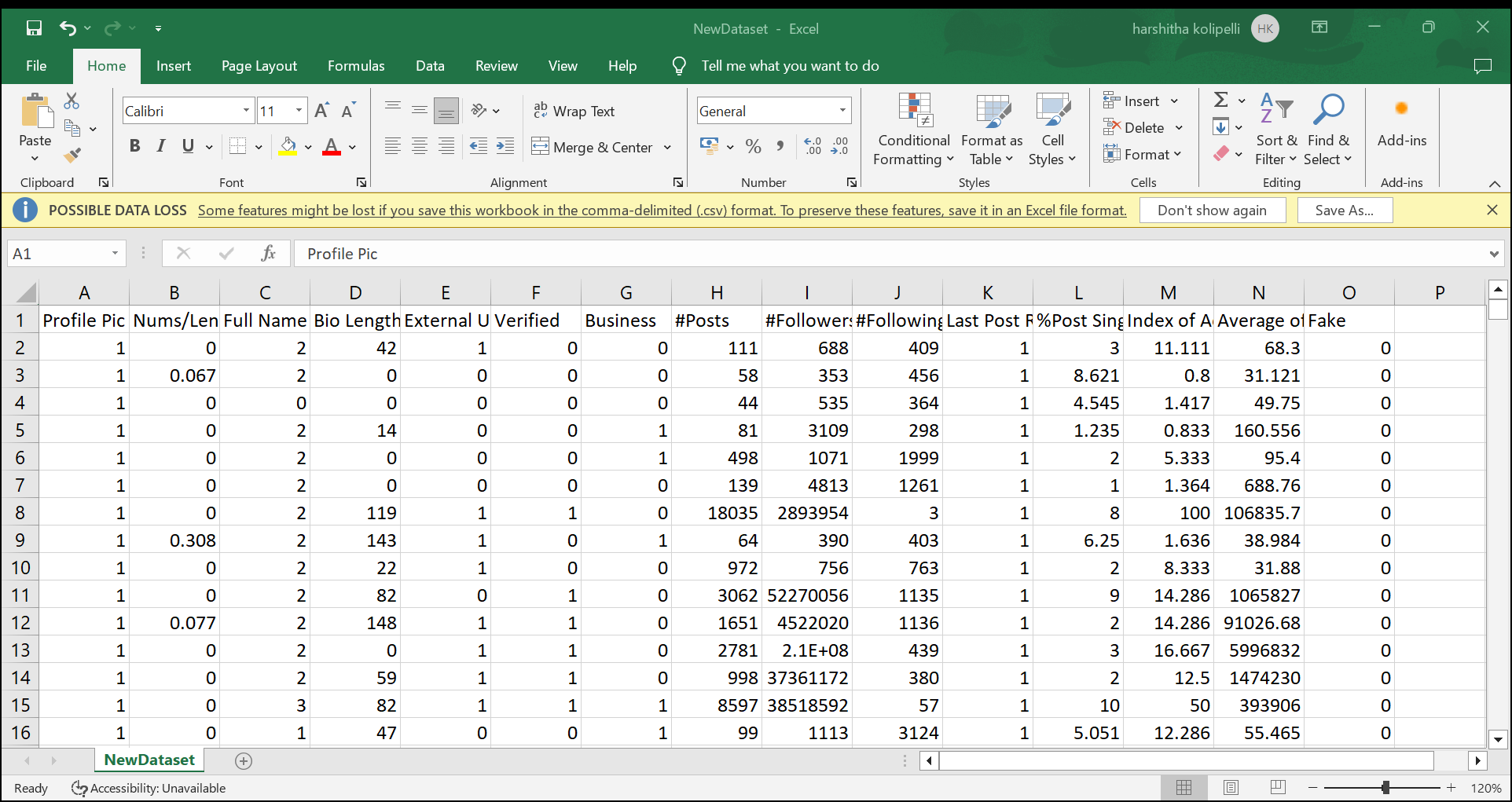
The first crucial step in developing an fake profile predictor is choosing a suitable training dataset. This data serves two key purposes: it allows us to train machine learning models and provides a benchmark for evaluating their accuracy. Our system utilizes the combination of fake and real profiles Dataset as input, accessible through user interfaces. The system then calculates outputs using decision tree classifier algorithm metrics like Accuracy & Predictions. Similar to previous methods, the training patterns are divided into training data set and testing data set. The training data set is utilized to train the machine learning algorithms (machine learning and deep learning), while the testing set is utilized to estimate the model's performance. Notably, the training data set and testing data set are distinct to avoid bias. Our specific dataset incorporates information about a profiles attributes .

Fig 3.3.2 Dataset

##### Training and Test Data

**Training Data:** For machine learning models, the training data's quality is crucial. In this project, the training data resembled a well-stocked recipe database Creating an effective training data set for fake profile detection entails several critical steps. Initially, it's crucial to ensure the data set is representative of the diverse range of both genuine and fake profiles found on social media platforms. This involves including various types of fake profiles, such as bots, spammers, and trolls, alongside genuine ones. Each profile in the data set must be accurately labeled as either genuine or fake, either through manual annotation or automated methods, to provide the necessary ground truth for training. Additionally, relevant features that distinguish between genuine and fake profiles, such as profile metadata, activity patterns, and content attributes, need to be carefully selected. Ensuring data quality by cleaning and preprocessing the data to remove noise and inconsistencies is essential to prevent biases in the training process. Addressing any imbalance between the classes of genuine and fake profiles and adhering to privacy and ethical guidelines are also crucial considerations. By meticulously addressing these factors, a high-quality training data set can be created, enabling the effective training of a decision tree classifier for fake profile detection on social media platforms.

**Testing Data:** Within the machine learning workflow, the testing data serves as a critical benchmark for evaluating model generalizability, a cornerstone for real-world applicability. This independent dataset mirrors the structure of the training data set, providing details regarding the information of the profiles that are existing. However, the key distinction lies without labels indicating whether fake profiles are present or not. This deliberate omission fosters an unbiased evaluation scenario. By presenting the models with unseen data, we can objectively assess their capacity for transfer the knowledge gleaned from the training phase.

##### Fake profiles Prediction Tool

To build the model few environments were required such as Google Colab, anvil interface, and Google Colab with pre-installed libraries such as Seaborn, numpy, pandas, etc,... The algorithm that is implemented is a decision tree classifier and the code is written in Python language in Google Colab. The data which was given to the algorithm is given in the CSV format. The correlation of the attributes is calculated so that we can improve accuracy in identifying fake profiles in social media accounts. From the correlation values, a few attributes are chosen by applying the chi-square attribute selection method. We choose attributes with high 3 correlation values by implementing the chi-square attribute selection method.

##### 3.3.4.Decision tree Accuracy

The system puts the trained model to the test, using it to predict fake profiles in unseen data (testing data). This assesses the model's ability to generalize its knowledge. Accuracy is then computed utilizing the accuracy\_score function, reflecting the model's success in making correct forecasts regarding the new data.

##### 3.3.5. Prediction on Decision tree Classifier

Takes the input from the user and gives the output whether the profile is fake or not.

##### 3.3.6.Architecture

**Input Data:** The project utilizes a dataset of attributes which consists of profiles information

**Output:** The project delivers four main results:

a. Decision tree Model Establishment and Accuracy: This module focuses on building and assessing the accuracy of th model in predicting fake profiles.

b. GBC Model Predictions: This module analyzes the specific predictions generated by the trained GBC model.

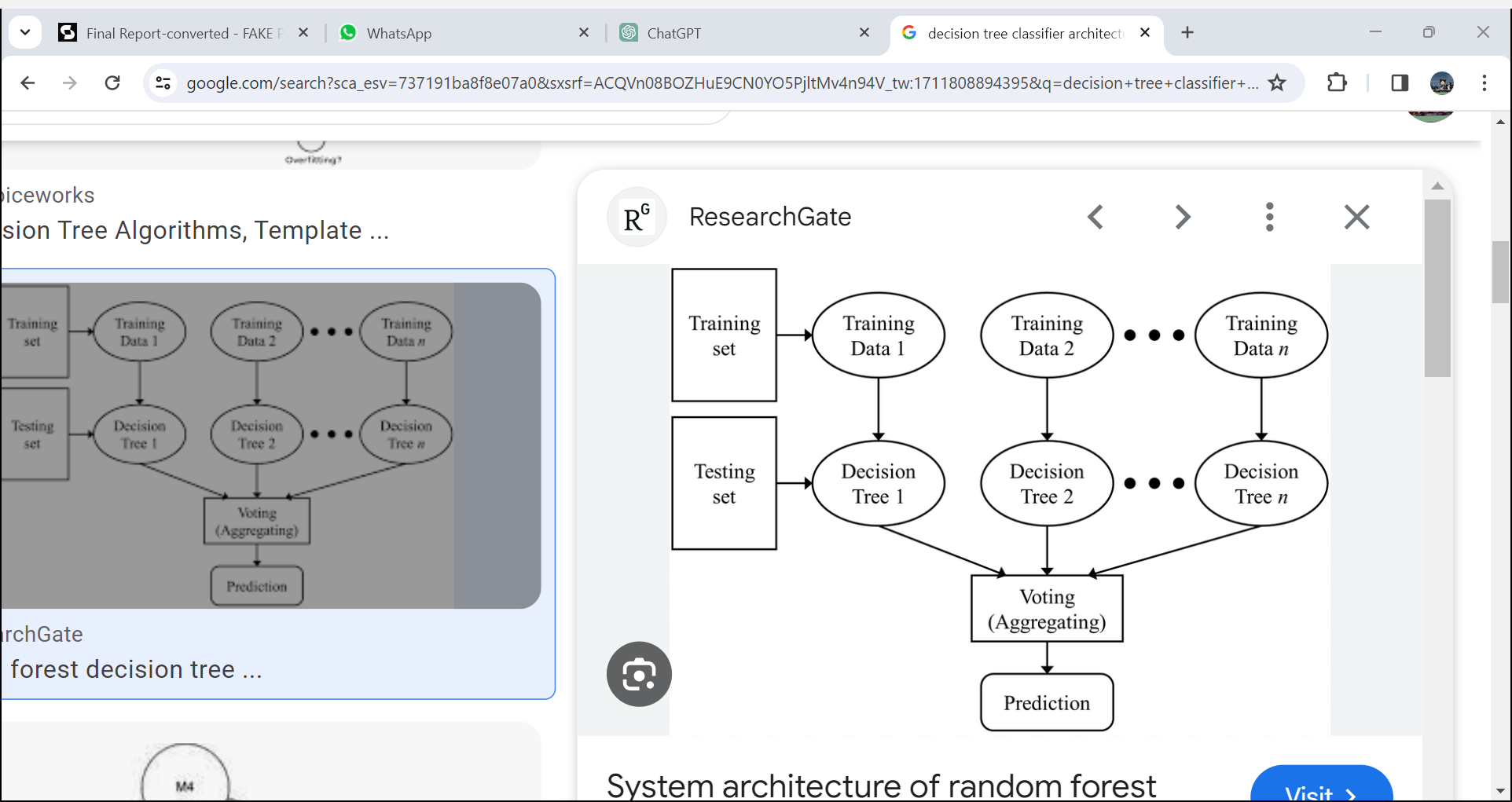


Fig 3.3.7 Architecture

##### Data Flow

The data flow in this project begins with the dataset detailing of various Profiles information in the form of attributes which helps us in getting to know the profile owners identity. This data is preprocessed, which may involve cleaning, transforming, and formatting it for model use. The prepared data is then fed into the Decision tree Classifier model for training. Once trained, these models accept new user input about Fake profiles patterns and generate predictions about the Fake profiles presence.

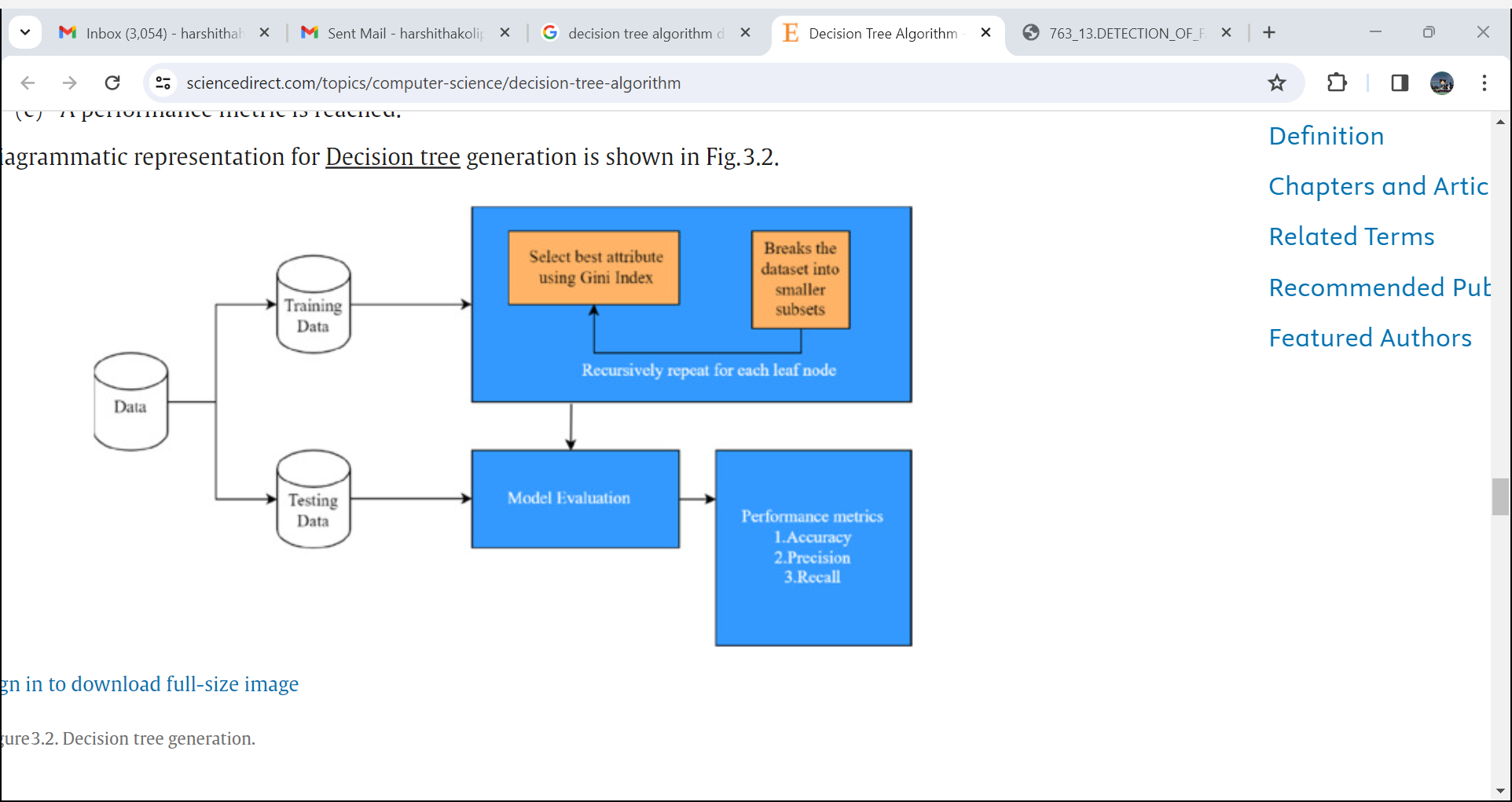
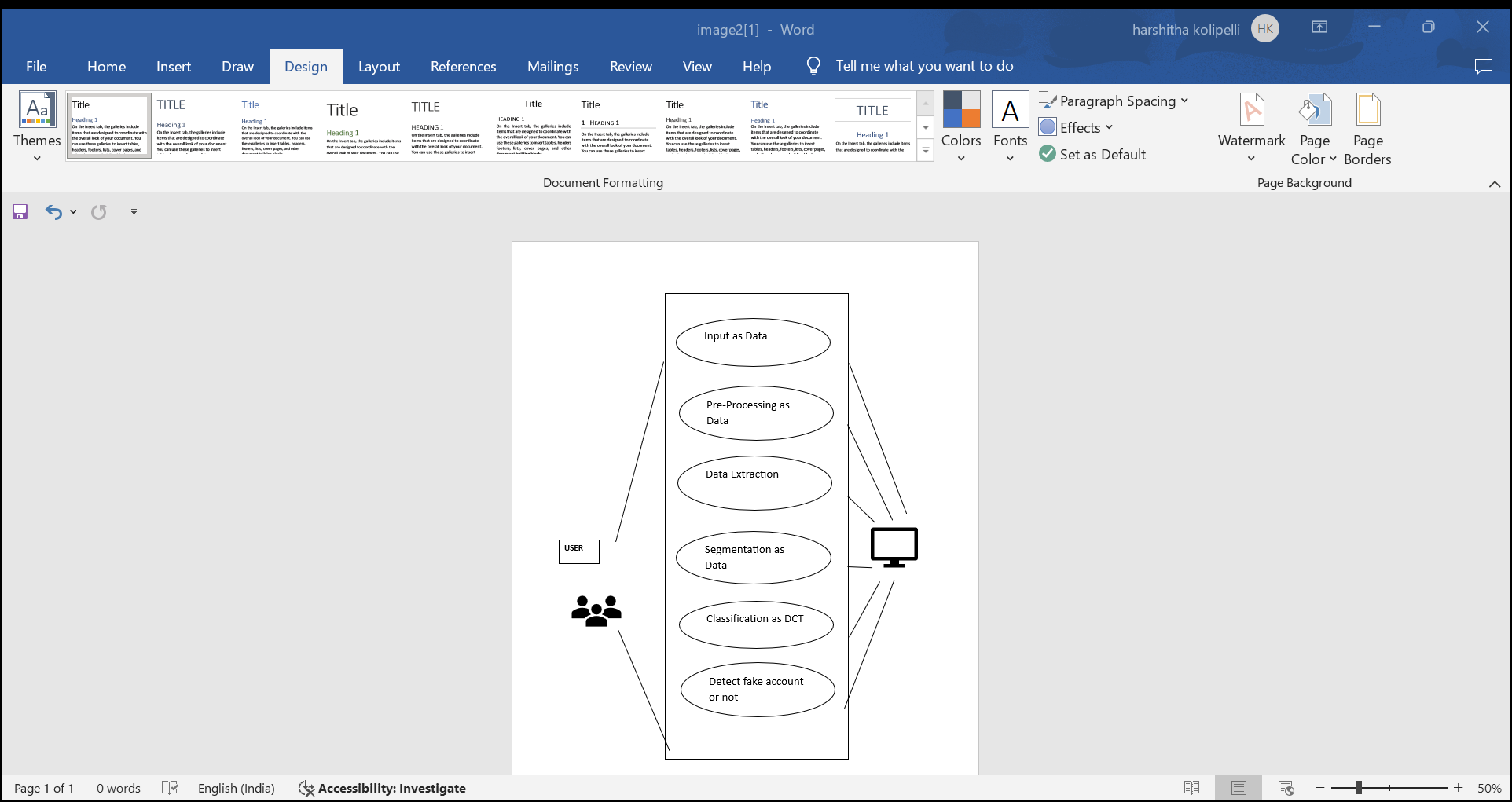


Fig 3.3.8 Data Flow

##### UML Diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. As we can see the user is interacting with system by a UI through which the customer can perform above mentioned operations like providing dataset containing the fake profiles indormation and then calculating the accuracies of Decision tree along with the predictions.

 Fig 3.3.9 UML Diagram

##### Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. From above mentioned sequence diagram we have to go in sequence: Enter the needed details as shown in the above figure, provide the Profiles data set containing attributes and then calculate the accuracies of Decision tree along with the predictions.

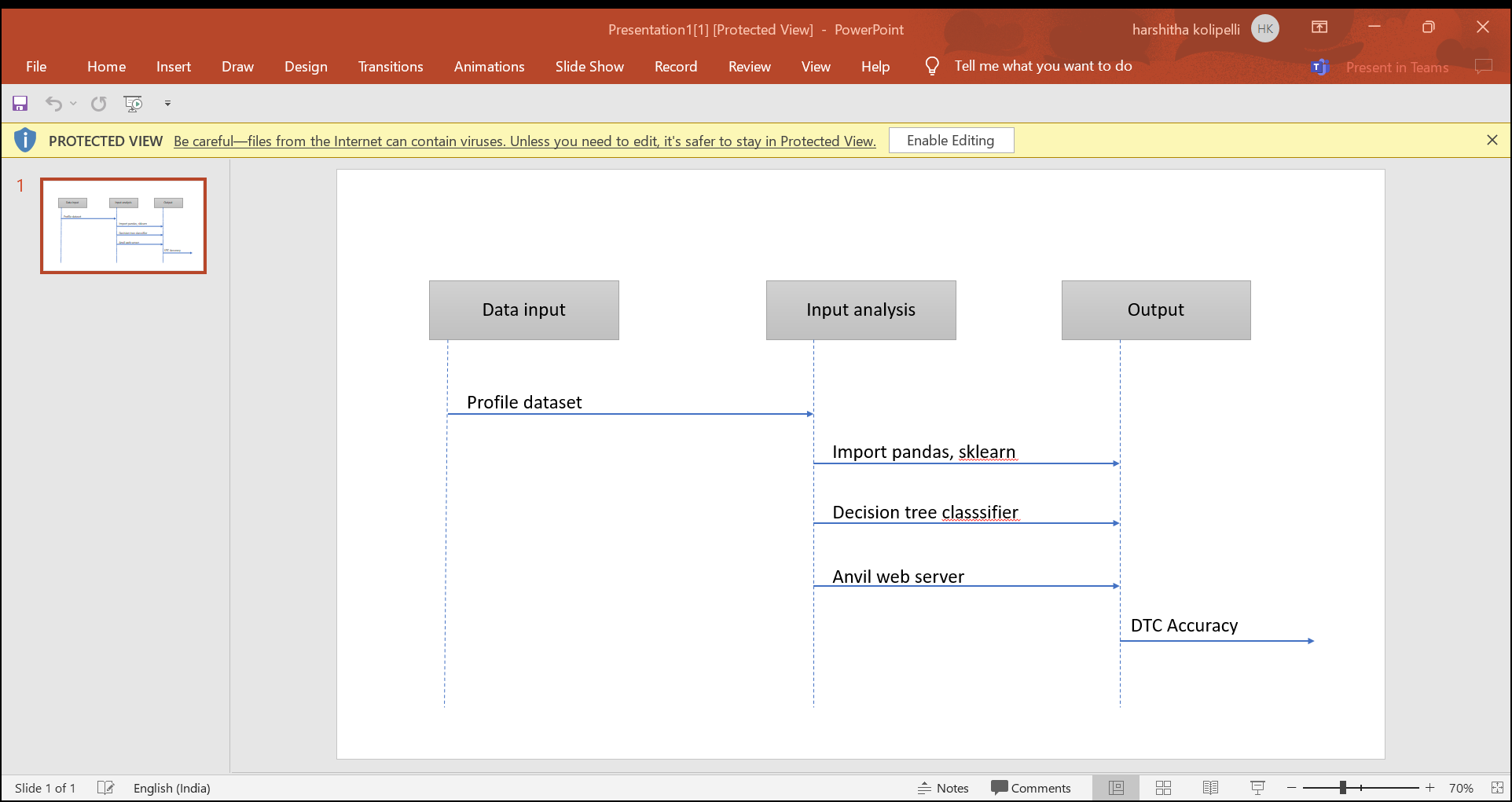


Fig 3.3.10 Sequence Diagram

* + 1. **Activity Diagram**

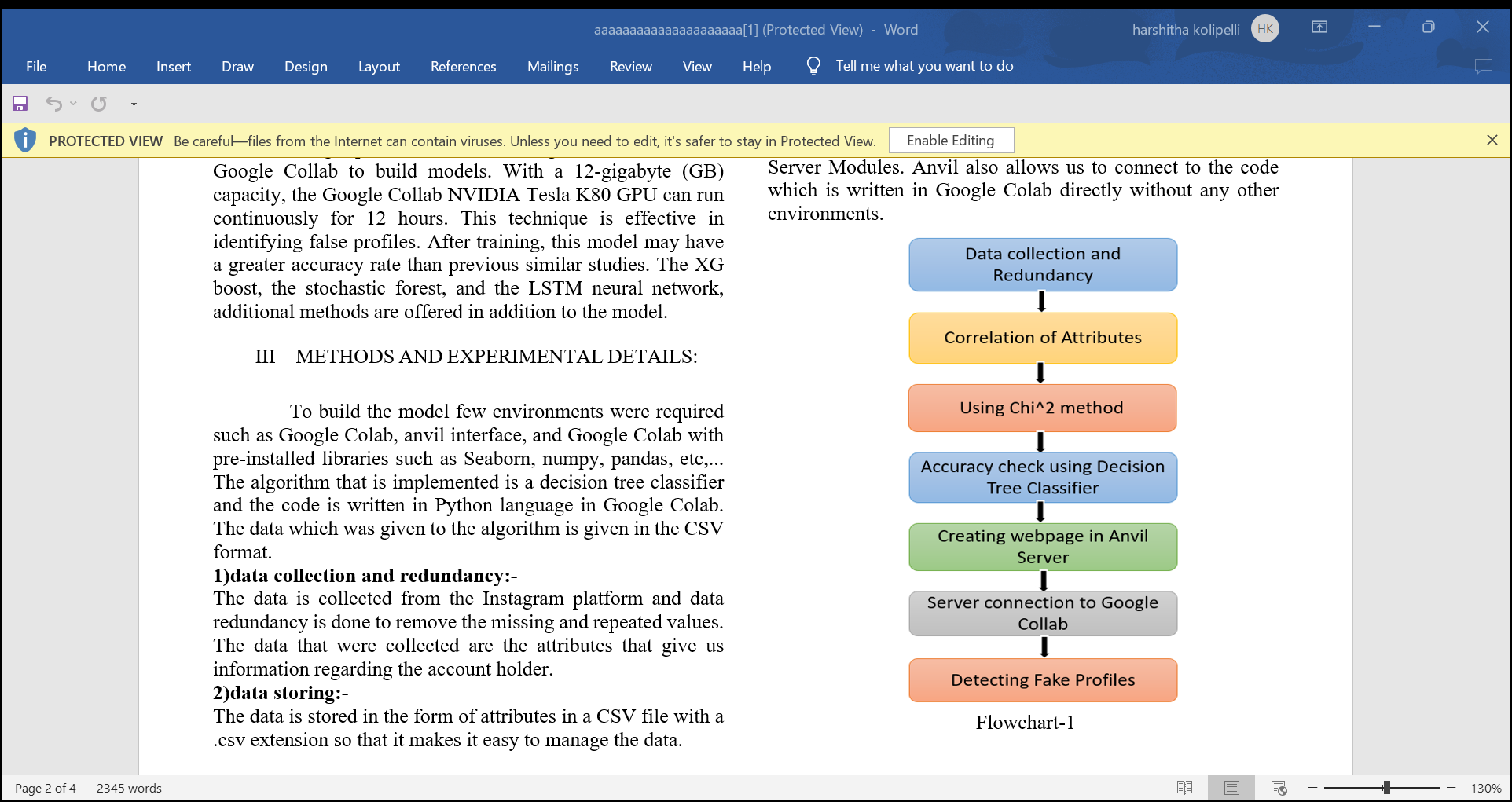
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. In activity diagram we can see that first provide the profiles dataset comprising of the information as attributes, and then train the model, test the model and calculate the accuracies along with the predictions of Decision tree classifier.

Fig 3.3.11 Activity Diagram

##### Stepwise Implementation and Code

###### Decision tree Accuracy

* Implements the Decision tree Classifier model.
* Preprocesses the data (mapping attributes to numerical values).
* Splits the data into training and testing sets.
* Trains the Decision tree model on the training data.
* Evaluates the model's accuracy on the testing data.

###### Classifier Predictions

* Loads the trained model.
* Provides a function to get user input about the details of the profile.
* Preprocesses the user input.
* Uses the trained model to make a prediction about Fake profiles.

###### Data Preprocessing (GBC & NN Accuracy)

* Import necessary libraries (NumPy, Pandas, scikit-learn, TensorFlow/Keras).
* Load the profiles.csv dataset into a Pandas DataFrame.
* Map the categorical attributes to numerical values

###### Model Building

1. **Decision tree Classifier Accuracy**

* Create a Decision tree Classifier object.
* Train the model using the fit method on the training data.
* Calculate accuracy using the .score method on the testing data.

###### Predictions (classifier)

* Load the trained models.
* Define functions to:
* Take profiles attributes as input from the user.
  + Preprocess the user input into the correct format.
  + Use the model.predict method to generate predictions.
  + Map the numerical prediction back to a label (e.g., 0 -> "not fake").

###### ANVIL WORKS

* Use PYTHON to design the interface with buttons.
* Connect button clicks to functions that execute the other code files.

##### ANVIL CODE

from .\_anvil\_designer import Form1Template

from anvil import \*

import anvil.server

class Form1(Form1Template):

def \_\_init\_\_(self, \*\*properties):

# Set Form properties and Data Bindings.

self.init\_components(\*\*properties)

# Any code you write here will run before the form opens.

pass

def button\_1\_click(self, \*\*event\_args):

"""This method is called when the button is clicked"""

# If a category is returned set our species

ls= self.L\_n.text

Fs=self.F\_ls.text

Fling=self.F\_ling.text

profile=anvil.server.call('predict\_Fake',ls,Fs,Fling)

result=profile

if result:

self.species\_label.visible = True

self.species\_label.text = "is a Fake"

else:

self.species\_label.visible = True

self.species\_label.text = "Not Fake"

self.species\_label.foreground="green"

pass

pass

##### DECISION TREE CLASSIFIER ACCURACY CODE

!pip3 install anvil

!pip3 install anvil-uplink

# Load libraries

import pandas as pd

from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier

from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function

from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

import anvil.server

anvil.server.connect("server\_YNCRQ6ADA7HVB567QMU4YGM6-OZYCNKDAO3GKMXEF")

from google.colab import drive

drive.mount('/content/drive')

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

pdata = pd.read\_csv('/content/drive/MyDrive/NewDataset.csv')

print(pdata.describe())

corr = pdata.corr()

corr

fig = plt.figure(figsize=(8,8))

plt.matshow(corr, cmap='RdBu', fignum=fig.number)

plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertical')

plt.yticks(range(len(corr.columns)), corr.columns)

#split dataset in features and target variable

#feature\_cols = ['Profile Pic', 'Nums/Length Username', 'Full Name Words', 'Bio Length', 'External Url','Verified','Business', '#Posts', '#Followers','#Following','Last Post Recent','%Post Single Day','Index of Activity','Average of Likes']

#feature\_cols = ['Nums/Length Username', '#Posts', '#Followers','#Following','Index of Activity','Average of Likes']

#feature\_cols = ['Nums/Length Username', '#Posts', '#Followers','#Following','Average of Likes']

#feature\_cols = ['Nums/Length Username', '#Followers','#Following','Average of Likes']

feature\_cols = ['Nums/Length Username', '#Followers','#Following']

#feature\_cols = ['Nums/Length Username', '#Followers',]

#feature\_cols = ['Nums/Length Username', 'Bio Length', 'Business','#Following','Last Post Recent']

X = pdata[feature\_cols] # Features

y = pdata.Fake # Target variable

# Split dataset into training set and test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42) # 70% training and 30% test

# Create Decision Tree classifier object

clf = DecisionTreeClassifier()

# Train Decision Tree Classifier

clf = clf.fit(X\_train,y\_train)

#Predict the response for test dataset

y\_pred=clf.predict(X\_test)

conf = metrics.confusion\_matrix(y\_test, y\_pred)

conf

from sklearn import tree

tree.plot\_tree(clf,filled=True)

# Model Accuracy, how often is the classifier correct?

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print("Accuracy:",metrics.precision\_score(y\_test, y\_pred))

classification = clf.predict([['0','38','5404']])

print (classification[0])

@anvil.server.callable

def predict\_Fake(ln,Fls,Fling):

classification = clf.predict([[ln,Fls,Fling]])

return classification[0]

anvil.server.wait\_forever()

# CHAPTER 4

## RESULTS AND DISCUSSION

### CHAPTER 4 RESULTS AND DISCUSSION

This section showcases our project's functionality through a series of steps and accompanying visuals. Executing the Application:

1. Launch the Anvil works Uplink
2. Web app link is given
3. Giving the input of the profile
4. Prediction of the profiles is given



Fig 4.0.1 Results

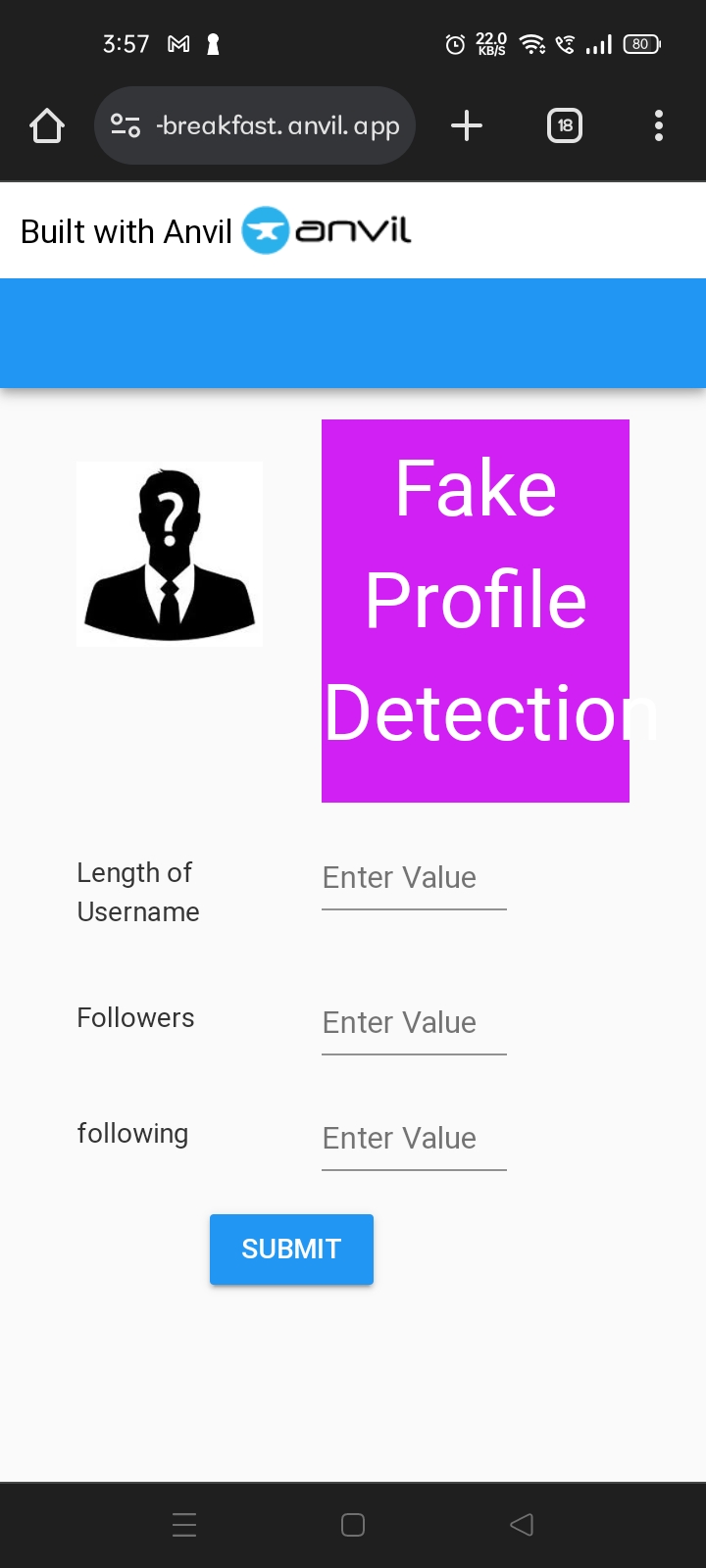


Fig 4.0.2 Web page

##### Performance Metrics

This section will evaluate the effectiveness of Decision tree Classifier (GBC) model in Identifying Fake profiles levels. The following metrics are used:

* + **Accuracy:** - This indicates that the model is approximately 96% accurate in its predictions on the testing data.
  + **Confusion Matrix:** The confusion matrix shows the counts of true positive, true negative, false positive, and false negative predictions. It breaks down correct and incorrect predictions by class .The matrix highlights specific areas where either the model may be struggling.
  + **Precision:** - Precision is the ratio of correctly predicted positive observations to the total predicted positives. It's a measure of the accuracy of the positive predictions. A precision of 0.8104 means that about 81.04% of the predictions labeled as positive are actually correct.
  + **Recall:** - Recall, also known as sensitivity or true positive rate, measures the ratio of correctly predicted positive observations to all actual positives. It indicates the model's ability to find all relevant cases within a dataset. A recall of 0.7926 means that about 79.26% of actual positives were correctly identified by the model.
  + **F1-Score:** - The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is often used as a single metric to evaluate a model's performance. A higher F1-score indicates better model performance. In this case, your model achieved an F1-score of 0.973, which is quite good.

##### Comparison of accuracies with other classifiers

Accuracy is a secondary performance metric used to assess the correctness of the model's predictions. Based on the below graph, which includes a list of classifiers with our model compared to the other existing classifiers. Accuracy is a common metric used to evaluate the performance of a classifier. It is defined as the ratio of the number of correct predictions to the total number of input samples. The accuracy score of each classifier is given for a range of scores from 0.0 to 1.0. The classifier with the highest accuracy point is the classifier considered to be most accurate that is, Decision tree classifier is highest among all the classifiers.

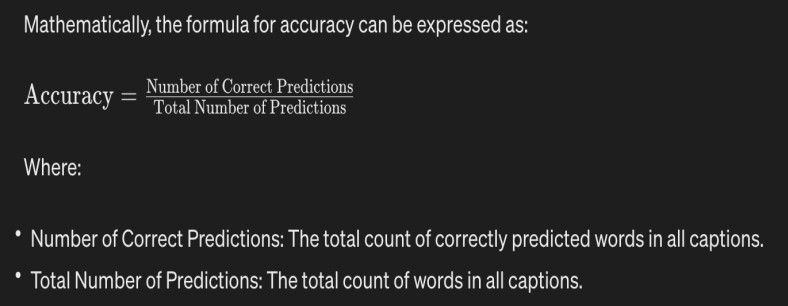


Fig 4.2.1 Accuracy Formula

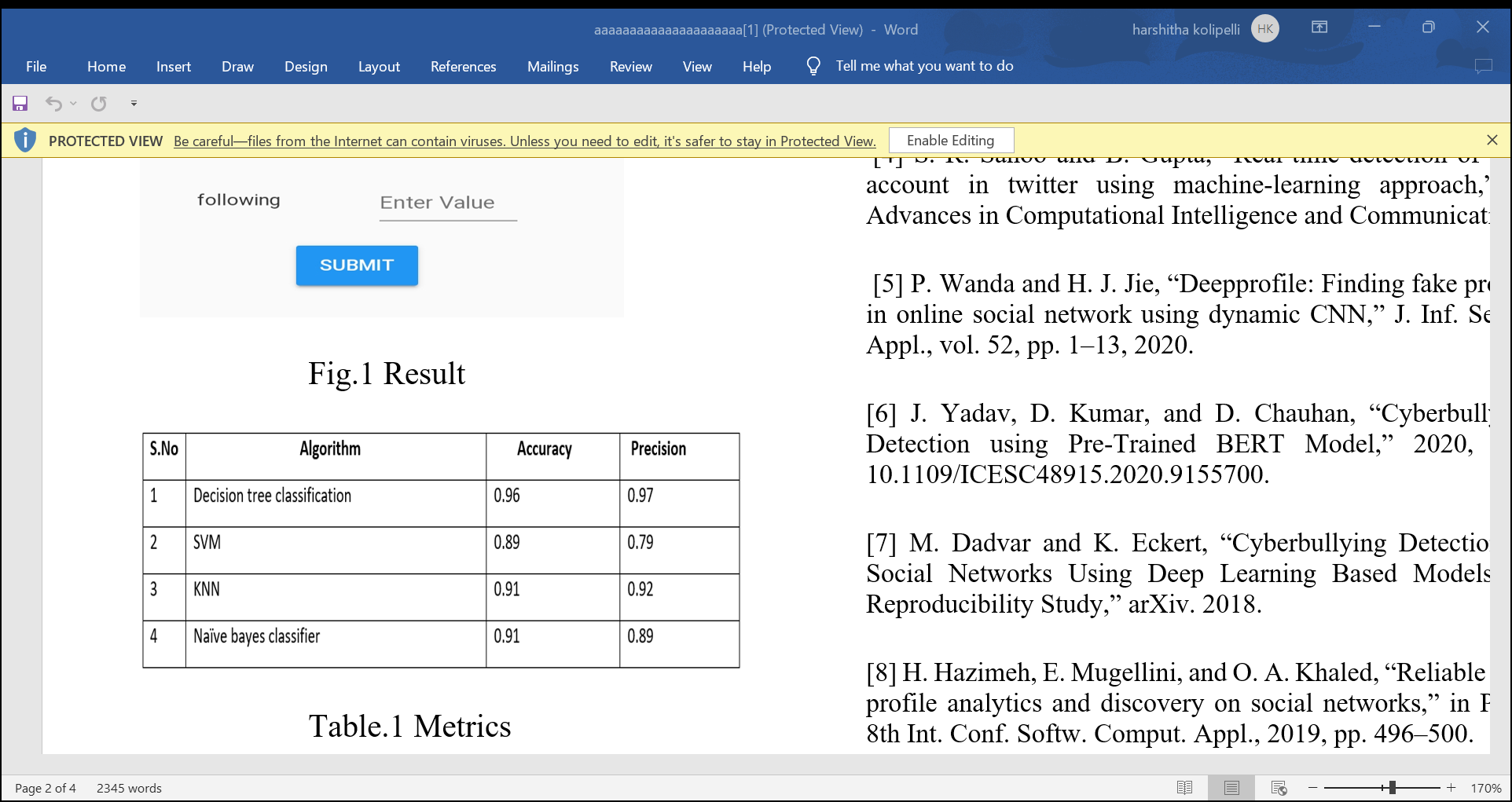


Fig 4.2.2 Accuracy comparison

CHAPTER 5

## CONCLUSION

**CHAPTER 5**

**CONCLUSION**

Based on our research, a significant problem plaguing social media platforms is the growing quantity of phony accounts on them. In order to solve this issue, we have developed a machine learning model that can quickly identify fake accounts, they can be eliminated before they endanger people seriously. In this study, a machine learning has been proposed while taking into consideration the shortcomings of the present approaches. The employed model examines the data linked to the accounts in order to establish a connection between it and the veracity of the account. We employed learning curves in addition to the model's accuracy to depict the model's performance using the decision classifier approach, which has high accuracy.

The model performed well on both the training and testing sets. Only the data supplied for Instagram profiles has been used so far for testing and training, but by providing an effective dataset for them, we may eventually train the model to recognize phony accounts on other well-known platforms like Facebook, LinkedIn, Twitter, and many more.

.

##### 5.1 Future Enhancement

* **Larger Dataset:** Expanding and diversifying your dataset will improve model robustness and ability to generalize to new situations. Consider acquiring more data points or using data augmentation techniques.

###### Feature Engineering:

* Explore additional features relevant to social media profiles
* Experiment with feature selection techniques to identify the most impactful attributes combinations.
* **Alternative Algorithms:** Investigate other advanced machine learning algorithms such as:
* Random Forest for potentially improved accuracy.
* Support Vector Machines (SVM) for handling smaller datasets.
* **Ensemble Methods:** Combine predictions from multiple models (e.g., GBC, NN, others) using voting or averaging techniques. This often leads to better overall results.
* **Interpretability:** Employ techniques like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations.
* **User Interface Improvement:** Develop a more sophisticated and user-friendly interface for your application. This could include:
* Data visualizations representing intake patterns and outcomes.
* Personalized dietary recommendations based on predictions.

## REFERENCES

### REFERENCES

1. International Journal of Creative Research Thoughts (IJCRT), Fake Account Detection Using Machine Learning, VOL.9, Issue 6 june 2021, T. Om Prathyush, N.Sai Kumar, E.Vishnu Priya.
2. Tech Science Press, Computer Systems Science and Engineering **2023**, *47*(1), 1091-1108., Deep Neural Network For Detecting Fake Profiles in Social Networks, Daniyal Amankeldin, Lyailya Kurmangaziyeva, Ayman Mailybayeva, Natalya Glazyrina, Ainur Zhumadillayeva, Nurzhamal Karasheva

**[3]** IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, VOL. 1, NO. 3, DECEMBER 2020 271 ” Fake Profile Detection on Social Networking Websites” A Comprehensive Review Pradeep Kumar Roy and Shivam Chahar

**[4]** https:/[/www.p](http://www.python.org/)y[thon.org/](http://www.python.org/)

**[5]** <https://www.numpy.org/>

**[6]** https://www.scirp.org/journal/paperinformation.aspx?paperid=120727

### GITHUB LINK

https://github.com/AdityaVardhan754/Major-Project-Phase-2

### JOURNAL PUBLICATION

