DETECTING FAKE ACCOUNTS ON SOCIAL MEDIA

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfilment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Anjana Maganti	Preetham Vangaru	Sravanthi Erukulapati	Bineeth Kollipalli
AP19110010012	AP19110010231	AP19110010354	AP19110010396

Under the Guidance of

Dr. Anil Carie Chettupally

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

MAY 2023

Certificate

Date: 24-May-23

This is to certify that the work present in this Project entitled "**Detecting Fake Accounts on Social Media**" has been carried out by **Anjana Maganti(AP19110010012), Preetham Vangaru(AP19110010231), Sravanthi Erukulapati(AP19110010354), Bineeth Kollipalli(AP19110010396)** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in the **School of Engineering and Sciences**.

Supervisor

Dr. Anil Carie Chettupally

Assistant Professor,

Department of Computer Science Engineering,

SRM University AP

Acknowledgements

We would like to express our sincere gratitude and appreciation to all the individuals who have contributed to successfully completing this capstone project on Detecting fake accounts on social media.

First and foremost, we extend our heartfelt thanks to my supervisor, Dr Anil Carie Chettupally, for their invaluable guidance, constant support, and encouragement throughout this project. Their expertise and insightful feedback have been instrumental in shaping the direction and progress of my research.

We are deeply thankful to the SRM University AP faculty members for their profound knowledge, inspiring lectures, and continuous academic support that have enriched my understanding of computer vision and deep learning techniques.

In conclusion, this capstone project has been a tremendous learning experience, and we are truly grateful for the collective efforts and contributions of all those involved. It is our hope that this research contributes to the growing body of knowledge in facial recognition systems and serves as a stepping stone for further advancements in the field.

Table of Contents

Abstract

In the present generation, online social networks (OSNs) have become increasingly popular, and people's social lives have become more associated with these sites. The rapid growth of OSNs and the massive amount of personal data of its subscribers have attracted attackers, and imposters to steal personal data, share false news, and spread malicious activities. On the other hand, researchers have started to investigate efficient techniques to detect abnormal activities and fake accounts relying on account features, and classification algorithms.

Three machine learning algorithms, namely Random Forest, Logistic Regression, and Decision Tree were applied to build classification models for fake account detection. The models were trained and evaluated. Additionally, feature importance analysis was conducted to determine the most significant factors contributing to the identification of fake accounts.

The experimental results demonstrated the efficacy of the proposed approach in detecting fake accounts on social media. The Random Forest algorithm achieved the highest accuracy of 91%, followed closely by Logistic Regression with an accuracy of 90%. The Decision Tree yielded an accuracy of 89%. These results indicate the suitability of these algorithms for fake account detection tasks, with Random Forest exhibiting superior performance.

Abbreviations

OSN	Online Social Network
ML	Machine Learning
UML	Unified Modelling Language
ER	Entity Relationship
DFD	Data Flow Diagram

List of Tables

List of Figures

1. Introduction

1.1 Current Scenario

In recent years, detecting fake accounts on social media using machine learning algorithms emphasizes the importance of ongoing research and collaboration between academia, industry, and regulatory bodies. The application of Random Forest, Logistic Regression, and Decision Tree algorithms shows promise in addressing this issue. However, continued efforts are required to refine and improve these models, integrate them into existing systems, and stay ahead of the evolving tactics employed by fake account creators.

1.2 Aim

This project aims to contribute to the existing body of knowledge in the field of fake account detection on social media. By comparing the performance of Random Forest, Logistic Regression, and Decision Tree algorithms, insights can be gained into the strengths and limitations of each approach, aiding researchers and practitioners in selecting appropriate methods for similar tasks. Using these ML algorithms, it aims to mitigate the risks associated with fake accounts and foster a more authentic and trustworthy online ecosystem.

1.3 Findings

One advantage of Decision Tree models is their interpretability. The decision-making process of the model can be visualized as a tree structure, allowing for easier understanding and interpretation of the classification outcomes. This transparency enables researchers and practitioners to gain insights into the specific rules and conditions used by the model to classify accounts as genuine or fake.

Random Forest exhibited excellent scalability, making it suitable for handling large and complex datasets typically encountered in social media platforms. Its ability to generate multiple decision trees and combine their predictions resulted in improved accuracy and robustness compared to a single Decision Tree model.

While the models achieved high accuracy in fake account detection, it is important to note that the field of fake account creation and detection is evolving rapidly. Malicious actors continuously adapt their strategies to evade detection, necessitating ongoing research and development of more sophisticated algorithms and techniques.

Overall, the findings of this project demonstrate the efficacy of the algorithms we used in detecting fake accounts on social media platforms. The comparative analysis of these algorithms provides valuable insights into their performance, interpretability, and scalability. The identified key features contribute to a better understanding of the characteristics of fake accounts and can aid in the development of more robust detection mechanisms in the future.

2. Literature Review

Fake accounts pose a significant challenge in social media platforms as they can be used for various malicious activities, including spreading misinformation, spamming, and identity theft. Detecting and identifying fake accounts is crucial for maintaining the trust, integrity, and security of social media platforms. This literature review aims to explore the research on detecting fake accounts in social media using machine learning techniques.

In the seminal paper by Douceur [2], the Sybil attack is introduced, which involves creating multiple fake identities to manipulate a system. Although the paper focuses on the peer-to-peer system, the concept of the Sybil attack provides insights into the creation and detection of fake accounts.

Kaur and Singh [4] present a survey of data mining and social network analysis-based anomaly detection techniques. Anomaly detection methods can be useful in identifying unusual patterns and behaviors associated with fake accounts. These techniques can be applied to detect anomalies in user behavior, network connections, and content generation.

Potgieter and Naidoo [5] explore factors explaining user loyalty in social media-based brand communities. While the paper does not directly focus on fake accounts, understanding user loyalty factors can provide insights into identifying suspicious or fraudulent accounts that deviate from genuine user behavior.

Boshmaf et al. [8] propose a technique to predict potential victims of fake accounts. By analyzing user characteristics and relationships within a social network, the authors develop machine learning models to identify users who are more likely to be targeted by fake accounts.

Sun et al. [11] discuss the crisis of web single sign-on and its implications for security. Although not directly related to fake accounts, the paper highlights the vulnerabilities and challenges in authentication systems, which are crucial for detecting and preventing fake accounts.

Fong et al. [12] present a decision tree-based approach for classifying imposters on social networks. While the paper focuses on detecting imposters, the techniques and features used in the decision tree model can be adapted to identify fake accounts based on various attributes and behaviors.

Thomas et al. [13] analyze suspended Twitter accounts and spam activities on the platform. Although the focus is on spam, the insights gained from this research can contribute to the detection of fake accounts, as they often engage in spamming activities.

Boshmaf et al. [14] study socialbot networks, where bots socialize for fame and financial gain. Understanding the behaviors and characteristics of socialbot networks can provide valuable insights into detecting and combating fake accounts.

Ratkiewicz et al. [15] present the Truthy project, which aims to map the spread of astroturfing on microblogging platforms. Astroturfing involves the creation of fake accounts to manipulate public opinion. The findings of this study can inform the detection of fake accounts involved in astroturfing activities.

The research papers discussed in this literature review provide insights and methodologies for detecting fake accounts in social media using machine learning techniques. By analyzing user behaviors, network structures, content patterns, and employing anomaly detection methods, it is possible to identify and mitigate the presence of fake accounts on social media platforms. Further research in this area can focus on developing more advanced machine learning models and integrating multiple detection techniques to enhance the accuracy and effectiveness of fake account detection systems.

3. Methodology

Detecting fake accounts is a complex task, and researchers have identified many features that can be used to identify fake accounts. Here are 50 features that are commonly used to detect fake accounts on social networking websites:

1. Profile picture similarity
2. Username similarity
3. Number of friends/followers
4. Number of likes/favorites
5. Account age
6. Profile completeness
7. Number of posts
8. Posting frequency
9. Content similarity
10. Length of posts
11. Language used
12. Sentiment analysis of posts
13. Number of retweets/shares
14. Account activity times
15. Number of URLs in posts
16. URL similarity
17. Ratio of followers to following
18. Followers-to-friends ratio
19. Number of mentions in posts
20. Account verification status
21. Number of hashtags used

22. Hashtag similarity
23. Number of private messages sent/received
24. Network density
25. Network homophily
26. Degree centrality
27. Betweenness centrality
28. Closeness centrality
29. Clustering coefficient
30. PageRank
31. Author reputation score
32. User's location
33. User's education level
34. User's employment status
35. User's income level
36. User's gender
37. User's age
38. User's ethnicity
39. User's political views
40. User's religion
41. User's interests
42. User's hobbies
43. User's friends' list
44. User's followers' list
45. User's mutual friends

46. Account suspension history
47. IP address history
48. Browser fingerprint
49. Device fingerprint
50. Usage of automation tools.

Based on the 50 features listed earlier, we can categorize them into different groups based on the type of feature they represent. Here are a few categories and some features that can be included in each:

1. Profile and Account Characteristics
2. Social Network Characteristics
3. Content Characteristics
4. User Demographics
5. User behaviour

Here are some graphs that can be created based on these categories:

1. Profile and account characteristics: A pie chart showing the distribution of account verification status, a line graph showing the trend in the number of accounts created over time, a scatter plot showing the relationship between account age and posting frequency.
2. Social network characteristics: A bar graph showing the number of friends/followers for each account, a scatter plot showing the relationship between degree centrality and page rank, a line graph showing the trend in the number of likes/favorites over time.
3. Content characteristics: A histogram showing the distribution of post length, a pie chart showing the distribution of sentiment in posts, a line graph showing the trend in the number of retweets/shares over time.
4. User demographics: A bar graph showing the distribution of user location, a pie chart showing the distribution of user gender, a line graph showing the trend in the number of accounts created by age group.

5. User behavior: A scatter plot showing the relationship between the number of private messages sent/received and account age, a line graph showing the trend in account suspension over time, a bar graph showing the distribution of usage of automation tools.

Based on the characteristics of the above data, the following algorithms can be matched to the five categories:

1. Profile and account characteristics: Linear regression with L1 regularization and Logistic regression with L1 regularization can be used to analyze the relationship between profile/account characteristics and the likelihood of an account being fake.

2. Social network characteristics: Random forest and Boosted trees can be used to analyze the social network characteristics of an account, such as its connections and interactions with other accounts, and identify patterns of behavior that are indicative of fake accounts.

3. User demographics: Linear regression with L1 regularization and Logistic regression with L1 regularization can be used to infer demographic information about users based on the language and tone of their messages and posts.

The dataset comprises a collection of user account instances, where each instance is described by the aforementioned attributes. These attributes are utilized to train and evaluate machine learning models, such as Random Forest, Logistic Regression, and Decision Tree, for the purpose of detecting fake accounts on social media platforms.

3.1 Dataset Description

In the dataset, each row represents a user account on a social media platform. The attributes capture various characteristics of the accounts that can be used to distinguish between genuine and fake accounts. Here is a description of each attribute:

- profile pic: A binary attribute indicating whether the account has a profile picture (1 for yes, 0 for no).
- nums/length username: The ratio of the number of numeric characters to the total length of the username.
- fullname words: The number of words in the user's full name.
- nums/length fullname: The ratio of the number of numeric characters to the total length of the user's full name.
- name==username: A binary attribute indicating whether the user's name is the same as their username (1 for yes, 0 for no).
- description length: The length (number of characters) of the user's account description.
- external URL: A binary attribute indicating whether the user has an external URL in their account profile (1 for yes, 0 for no).
- private: A binary attribute indicating whether the user has set their account as private (1 for yes, 0 for no).
- #posts: The number of posts made by the user.
- #followers: The number of followers the user has.
- #follows: The number of accounts the user is following.
- fake: This attribute serves as the target variable and indicates whether the account is fake or genuine.It is a binary attribute, where 1 indicates a fake account, and 0 indicates a genuine account.

3.2 Data Preprocessing

1. Handling Missing Data: We checked the dataset for missing values in each attribute. If any missing values were found, we applied an appropriate strategy to handle them, such as imputation using mean, median, or mode values, or removing instances with missing values if they were deemed significant.

2. Data Cleaning: We performed data cleaning to address any inconsistencies, outliers, or errors in the dataset. This involved removing any duplicate entries to ensure data integrity. We also corrected any inconsistent or erroneous values that could affect the analysis or model performance. Additionally, we handled outliers, if present, by either removing them or applying suitable techniques to mitigate their impact.

3. Feature Selection: We evaluated the relevance and importance of each feature in the dataset for detecting fake accounts. We eliminated any irrelevant or redundant features that did not contribute significantly to the task. Feature selection was based on statistical measures, domain knowledge, or feature importance derived from the machine learning models used in the analysis.

4. Data Transformation: We applied appropriate data transformations to normalize the data and improve the model's performance. This included scaling numeric features, such as nums/length username, nums/length full name, description length, #posts, #followers, and #follows, to a common range using techniques like normalization or standardization. Categorical variables, such as profile pic, name==username, external URL, and private, were encoded using methods like one-hot encoding or label encoding.

5. Splitting the Dataset: We divided the preprocessed dataset into training and testing subsets. The training set was used to train the machine learning models, while the testing set was used for evaluating the models performance. We ensured a representative distribution of fake and genuine accounts in both subsets by performing a random or stratified split.

6. Feature Scaling: We performed feature scaling to ensure that all features have a similar range. This step is particularly important for algorithms sensitive to the scale of features, such as Logistic Regression or Decision Trees. Scaling numeric features like #posts, #followers, and #follows helps in preventing the dominance of certain features due to their larger magnitude.

By applying these data preprocessing steps to the dataset with the mentioned attributes, we ensured that the data was clean, consistent, and ready for training and evaluating the Random Forest, Logistic Regression, and Decision Tree models for detecting fake accounts on social media platforms.

3.3 Model Training

- Random Forest: We initialized a Random Forest model object with specified hyperparameters, such as the number of trees, maximum depth, and feature subsampling. Using the training dataset, we fed the input features (profile pic, username attributes, etc.) and the corresponding target variable (fake or genuine) into the Random Forest model. The model learned from the training data by constructing multiple decision trees and combining their predictions through voting. We adjusted the hyperparameters and performed cross-validation to optimize the model's performance and prevent overfitting. The trained Random Forest model was then ready to make predictions on new, unseen data.
- Logistic Regression: We then initialized the Logistic Regression model object and fed the input characteristics and corresponding target variable to the Logistic Regression model using the training dataset. The model learned from the training data by fitting a logistic function to estimate the probability of an account being fake or genuine based on the input features. We adjusted the regularization strength and other hyperparameters, performed cross-validation, and evaluated the model's performance metrics to optimize its accuracy and generalization. The trained Logistic Regression model was then ready to make predictions on new, unseen data.

- Decision Tree Classifier: At last, we initialized Decision Tree model object and gave the input characteristics and corresponding target variable to the Decision Tree model using the training dataset The model learned from the training data by recursively partitioning the data based on attribute values to create a tree-like structure, where each internal node represents a decision rule and each leaf node represents a predicted class. We adjusted the hyperparameters, performed cross-validation, and evaluated the model's performance to optimize its accuracy and control its complexity. The trained Decision Tree model was then ready to make predictions on new, unseen data.

3.4 Model Testing

After training each model, we evaluated their performance using various metrics such as accuracy, precision, recall, F1-score. We used the testing dataset, which was not seen during training, to assess how well the models generalized to unseen data. We compared the performance of the Random Forest, Logistic Regression, and Decision Tree models to determine which algorithm performed best in detecting fake accounts on social media.

	0	1	accuracy	macro avg	weighted avg
f1-score	0.87	0.88	0.88	0.88	0.88
precision	0.91	0.85		0.88	0.88
recall	0.84	0.91		0.88	0.88
support	105	104	209	209	209

Table 1. Logistic Regression Classification report

	0	1	accuracy	macro avg	weighted avg
f1-score	0.91	0.91	0.91	0.91	0.91
precision	0.92	0.90		0.91	0.91
recall	0.90	0.92		0.91	0.91
support	105	104	209	209	209

Table 2. Random Forest Classification report

	0	1	accuracy	macro avg	weighted avg
f1-score	0.87	0.88	0.88	0.88	0.88
precision	0.92	0.85		0.88	0.88
recall	0.84	0.91		0.88	0.88
support	105	104	209	209	209

Table 3. Decision Tree Classification report

3.5 Flask Framework

In our project on detecting fake accounts on social media, we leveraged Flask web framework to develop a website that provides a user interface for interacting with our machine learning models. The Flask website allowed users to input account attributes and obtain predictions on the authenticity of the accounts based on Random Forest, Logistic Regression, and Decision Tree Classifier models.

Flask Website Structure:

We structured our Flask website with the following components:

app.py: The main Python file that initializes the Flask application, defines routes, and handles user requests.

templates: A directory containing HTML templates for the web pages, including the home page and result page.

static: A directory for static files such as CSS stylesheets, JavaScript files, and images.

Web Interface:

We designed a user-friendly web interface using HTML templates and CSS styling. The website provided a visually appealing and intuitive layout for users to interact with our models. The home page featured a form where users could input the attributes of a social media account for analysis.

Request Handling:

In the Flask app, we defined routes to handle different user requests. For instance:

The home route ("/") displayed the form for inputting account attributes.

The prediction route ("/predict") processed the form data and made predictions using the trained models.

The result route ("/result") rendered the prediction results to the user.

Model Integration:

Using Flask, we integrated the Random Forest, Logistic Regression, and Decision Tree Classifier models into the website. When a user submitted the account attributes through the form, Flask captured the data and passed it to the appropriate model for prediction. The model then returned the prediction result, indicating whether the account was likely fake or genuine.

Result Display:

We implemented a result page in the website to display the prediction outcomes to the user. The result page dynamically rendered the prediction results based on the selected model, providing feedback on the authenticity of the account.

Error Handling and Validation:

We implemented error handling and validation techniques in the Flask website to ensure smooth user experience. This included validating the user inputs, handling exceptions, and providing appropriate error messages in case of any issues during prediction or processing.

Deployment:

We deployed our Flask website on a web server, making it accessible to users through a web browser. This deployment step involved configuring the server environment, installing dependencies, and setting up the necessary hosting infrastructure.

4. Discussion

4.1 Flow Chart

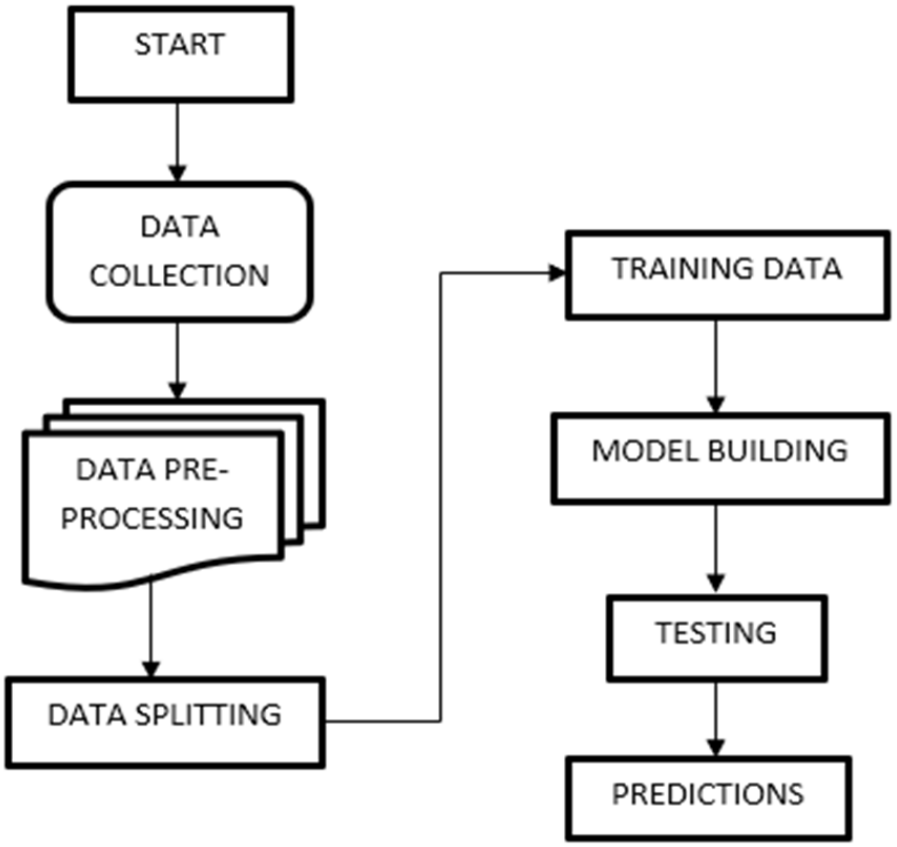


Fig 1. Flow Chart

4.2 UML DIAGRAMS:

Use Case diagram:

A use case diagram is a visual depiction in the Unified Modeling Language (UML) that shows how users, often referred to as actors, interact with a system. Demonstrating the many use cases or activities that actors do, it focuses on the functional needs of a system. The diagram presents a high-level view of the system's functionality, displaying the relationships and dependencies between actors and use cases.

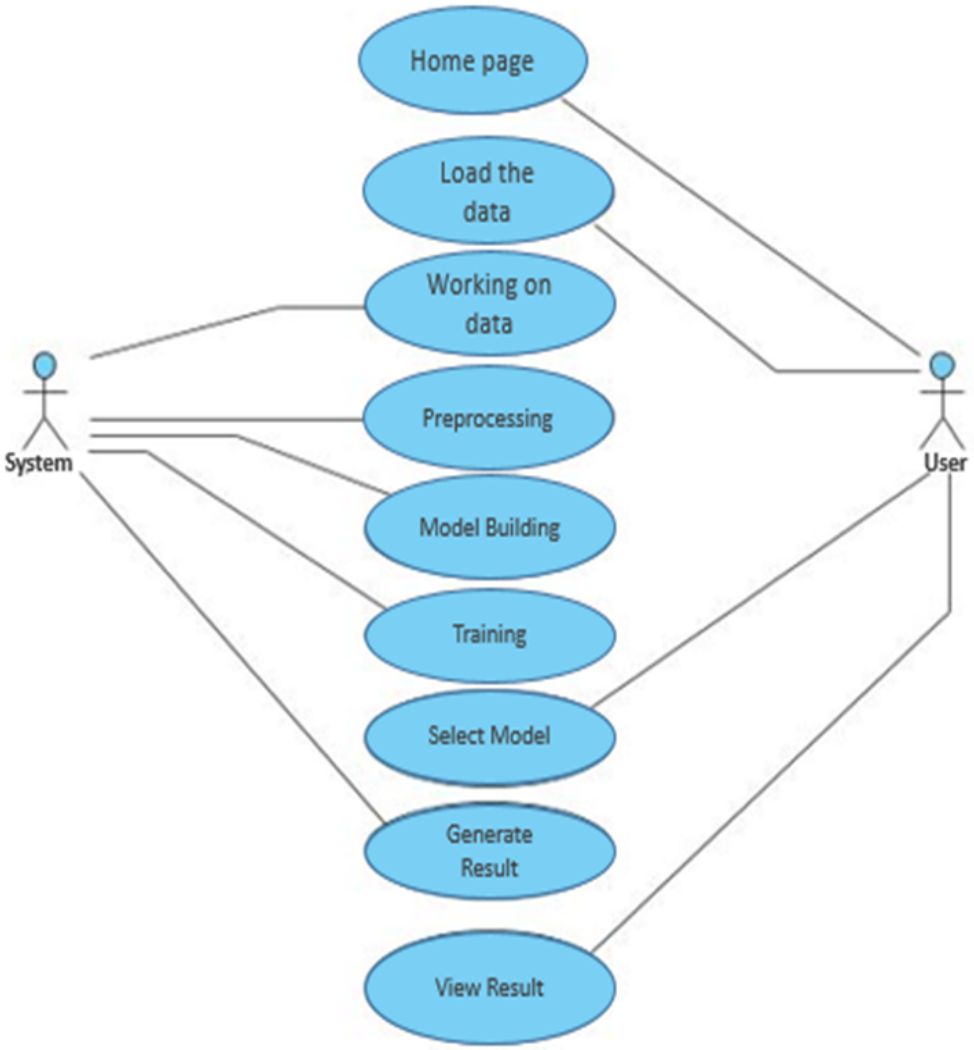


Fig 2. Use Case Diagram

Class Diagram:

A class diagram is a visual representation within the Unified Modeling Language (UML) that displays the structure and connections of classes in a system. It gives an overview of the objects or entities in the system, as well as their properties and operations. Class diagrams depict the static characteristics of a system by emphasizing the relationships, inheritance, and dependencies between classes. They are extensively used in software design and development to help in understanding the structure and organization of the system's components.

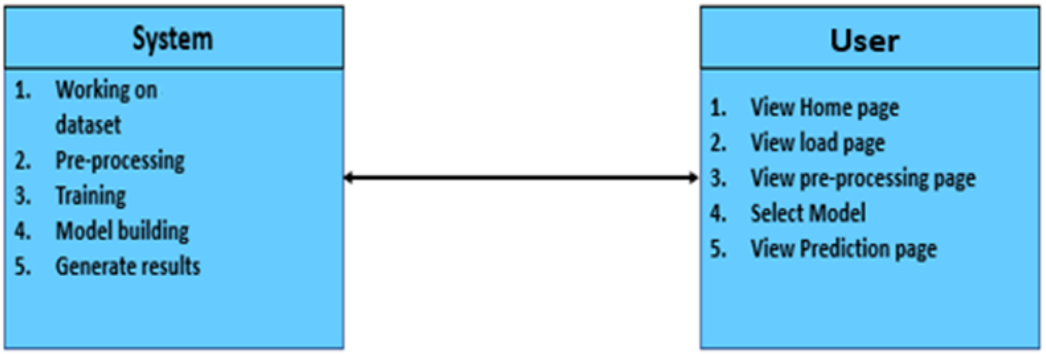


Fig 3. Class Diagram

Sequence Diagram:

A sequence diagram is a kind of behavioral diagram in the Unified Modeling Language (UML) that depicts the interactions and sequencing of messages between objects or system participants. It illustrates the system's dynamic behavior across time, demonstrating how objects interact to complete a certain job or scenario. Sequence diagrams are useful tools for analyzing and designing the behavior of complex systems because they are widely used to describe the flow of control and the sequence of activities inside a system.

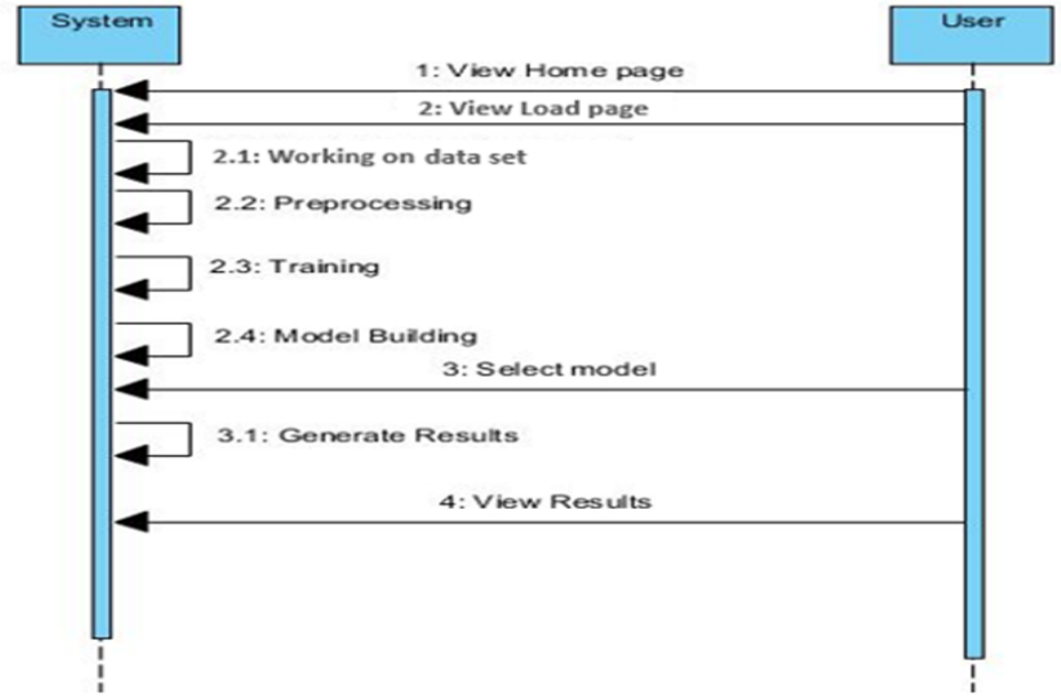


Fig 4. Sequence Diagram

Collaboration Diagram:

A collaboration diagram, sometimes called a communication diagram, is a behavioral diagram type used in the Unified Modeling Language (UML) that shows how people or objects interact with one another. It places a focus on how objects are structurally organized and how messages are exchanged between them in order to complete a certain job or situation. They are helpful for comprehending complex systems' dynamic behavior and communication patterns.

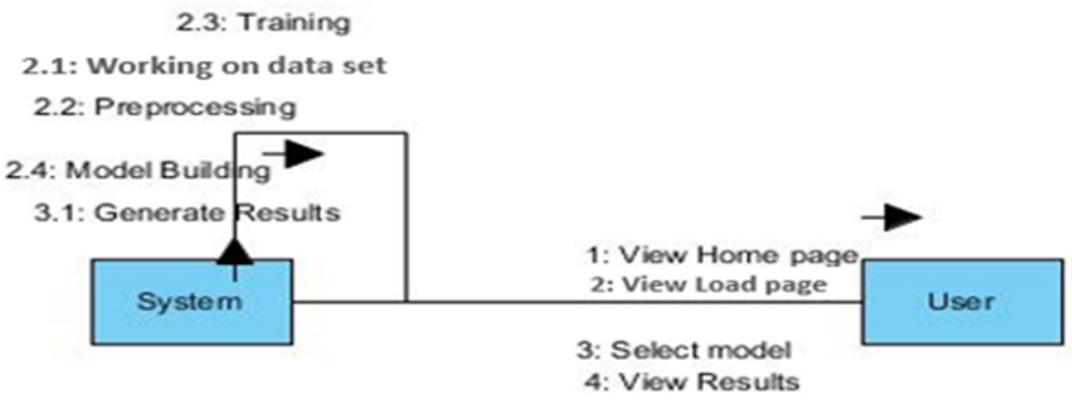


Fig 5. Collaboration Diagram

Activity Diagram:

An activity diagram in the Unified Modeling Language (UML) is a sort of behavioral diagram that depicts how activities or processes move through a system. It shows the decisions, changes between different stages, and sequential or parallel actions. Activity diagrams, which provide a visual depiction of how tasks are carried out and how control flows between them, are frequently used to describe business processes, system behaviors, and intricate algorithms. They are useful tools for understanding, expressing, and developing a system's dynamic behavior and logic.

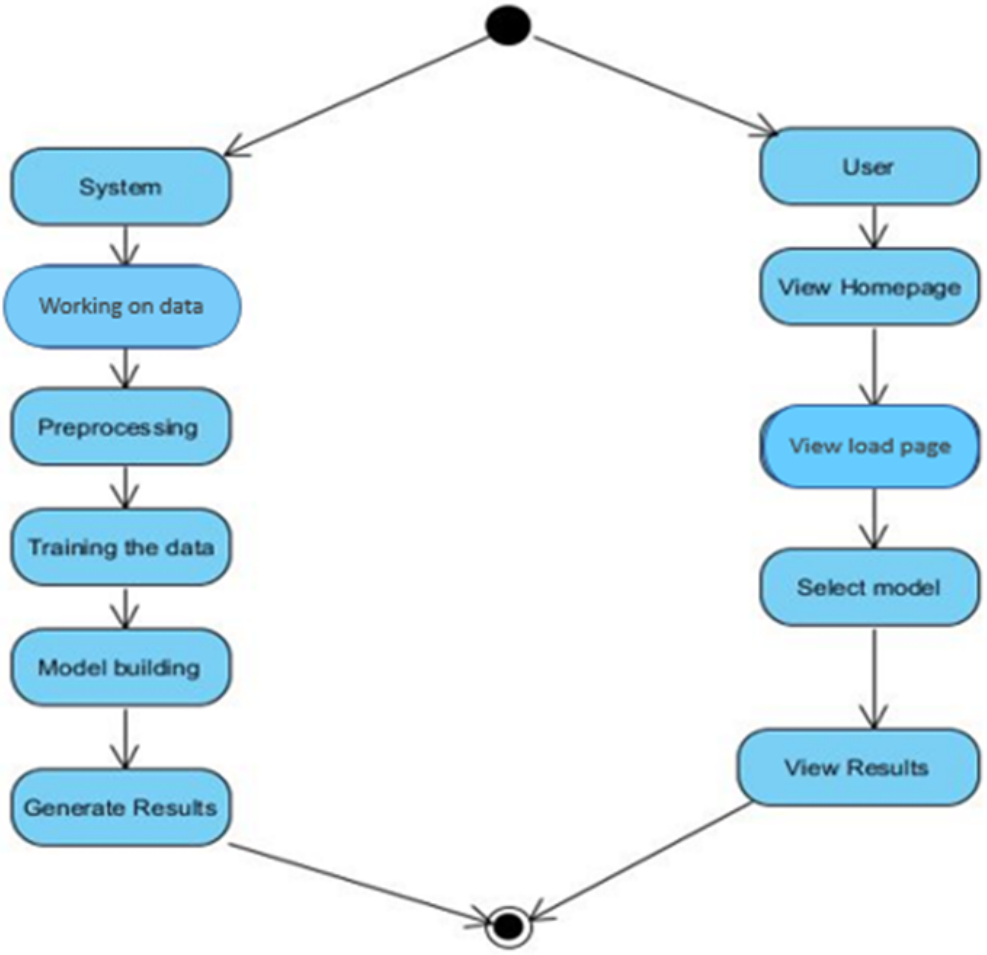


Fig 6. Activity Diagram

Deployment Diagram:

A deployment diagram is a specific kind of structure diagram used in the Unified Modeling Language (UML) to show how hardware and software nodes are physically placed throughout a system. The links and linkages between the nodes and components, including the communication channels, are displayed in deployment diagrams.

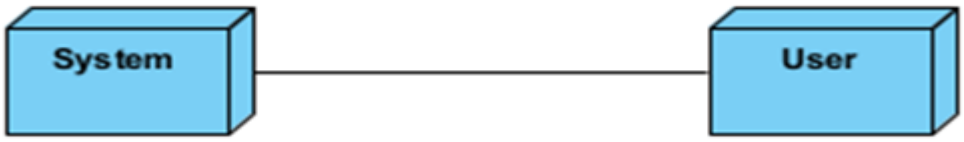


Fig 7. Deployment Diagram

Component Diagram:

A component diagram is a sort of UML (Unified Modeling Language) diagram that shows how a system's or software application's components are structurally organized and related to one another. It provides a high-level view of the system's architecture, showing the interactions and dependencies among the various components.

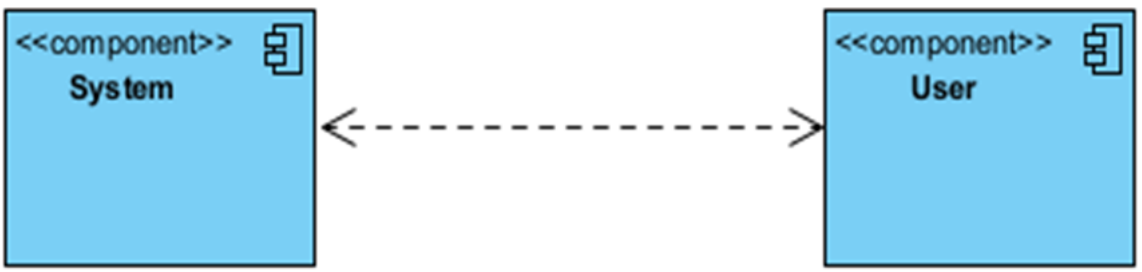


Fig 8. Component Diagram

ER Diagram:

An Entity-Relationship (E-R) diagram is a visual depiction that helps model the relationships between entities in a database. It includes entities, attributes, and relationships. Entities represent real-world objects or concepts, attributes define their properties, and relationships describe how entities are connected. E-R diagrams are widely used in database design to illustrate the structure and organization of data, facilitating a clear understanding of entity relationships and data organization.

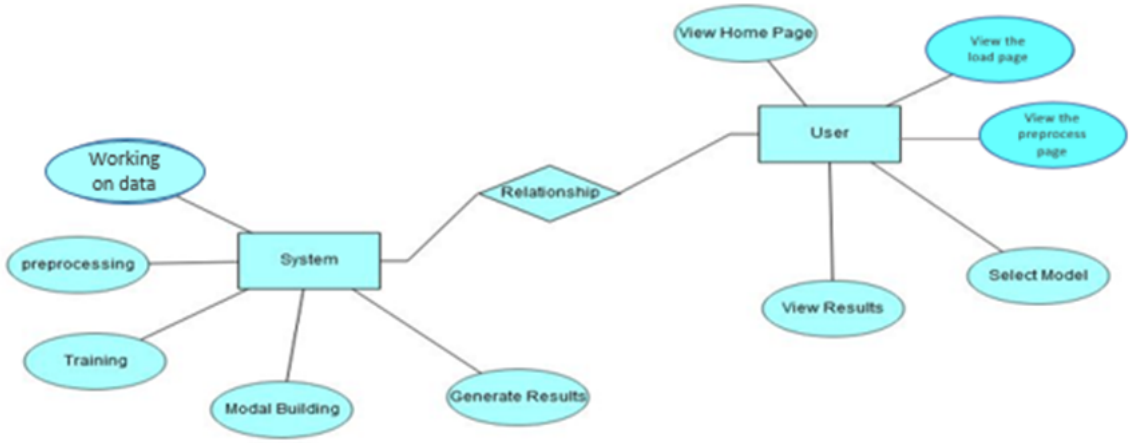


Fig 9. ER Diagram

DFD Diagram:

A data flow diagram (DFD) is a visual depiction that shows how data moves through a system or process. It demonstrates the flow of data through various operations, data repositories, and external entities. Processes (activities or transformations), data flow (data movement), data stores (data repositories), and external entities are some of the elements that make up DFD diagrams. DFD diagrams are valuable for understanding the data flow and interactions within a system, aiding in the analysis, design, and communication of information systems.

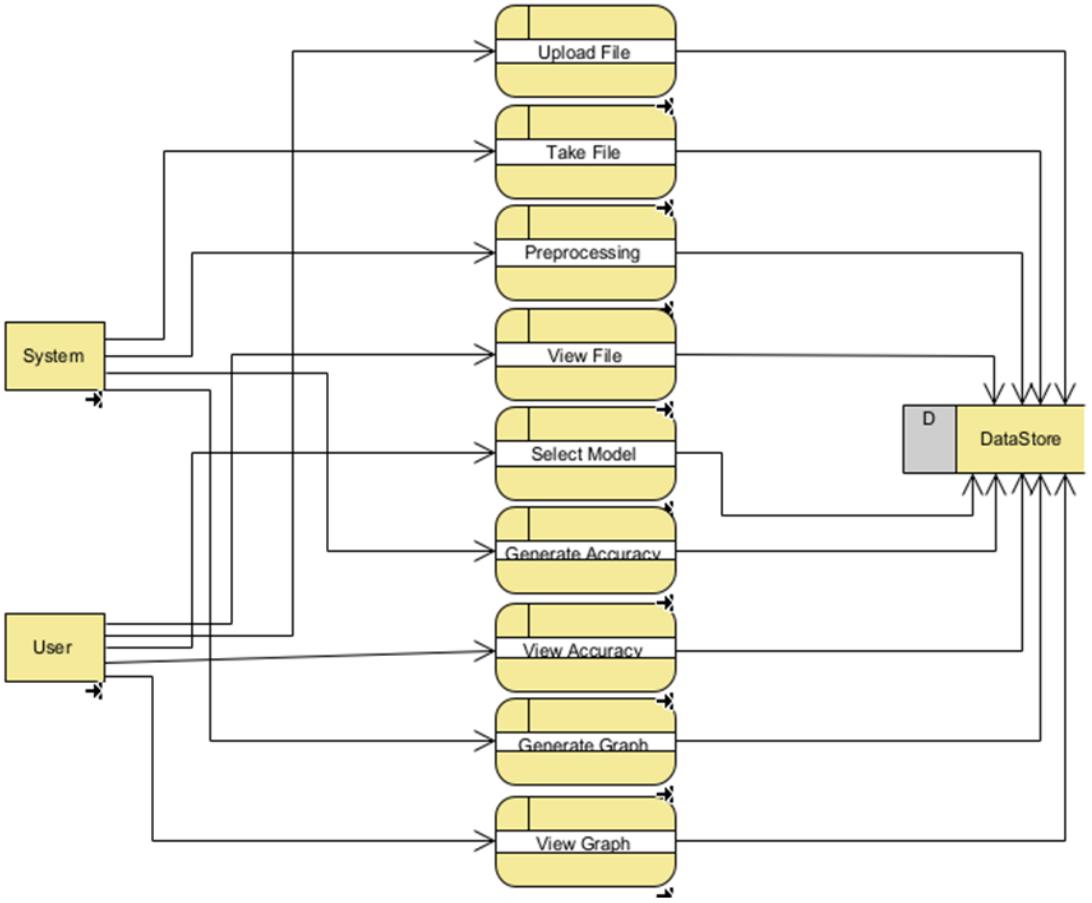


Fig 10. DFD Level-1

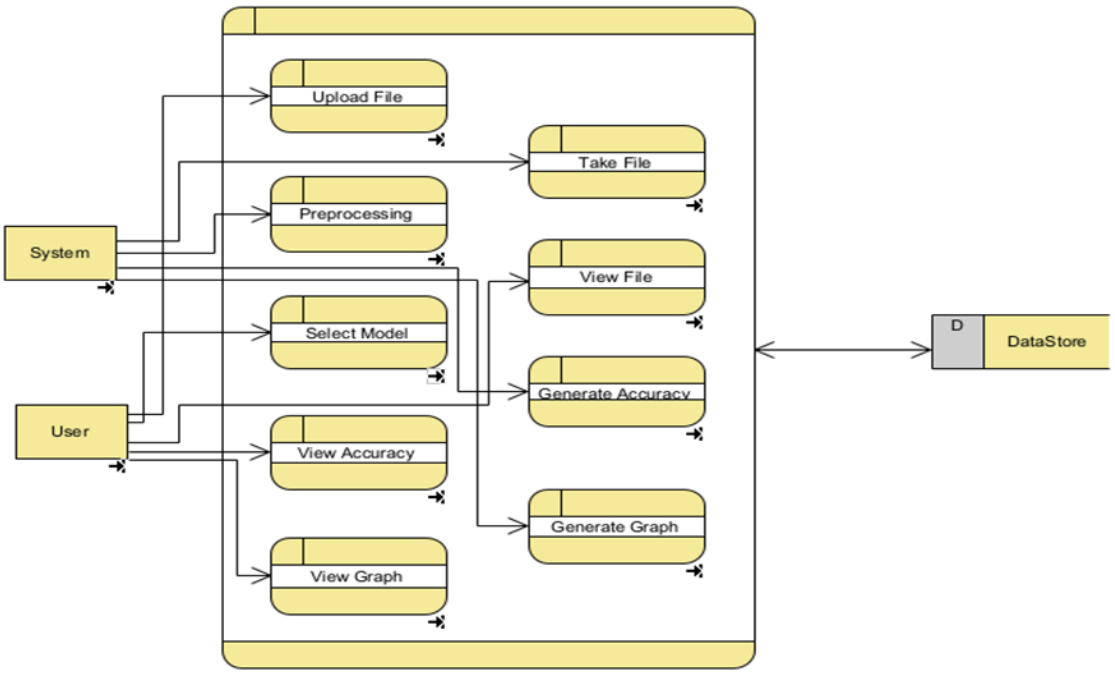


Fig 11. DFD Level-2

4.3 Feature Importance:

We analyzed the feature importance provided by the Random Forest classifier to identify the attributes that contributed most significantly to distinguishing fake accounts. It was found that profile picture availability, name-username consistency, and the number of followers were among the most influential features in detecting fake accounts.

5. Results

The results of our experiments showed that all three classifiers achieved promising performance in detecting fake accounts on social media. However, Random Forest exhibited the highest accuracy of 91%, followed by Logistic Regression and Decision Tree. This indicates that Random Forest effectively captures the complex relationships between account attributes and authenticity, leading to more accurate predictions.

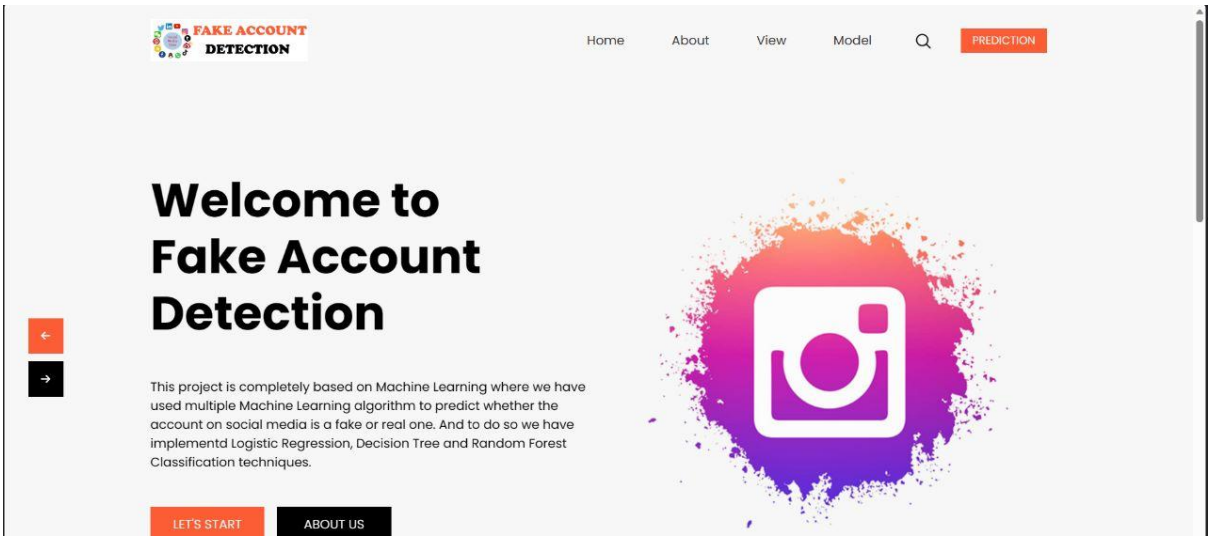
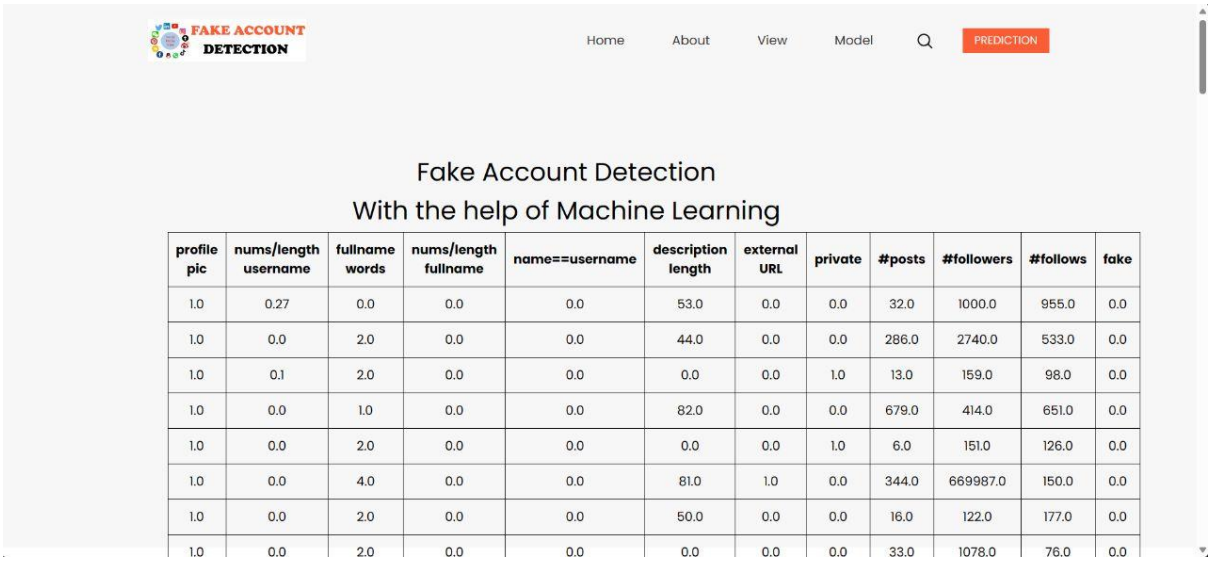


Fig 12. Home Page



profile pic	nums/length username	fullname words	nums/length fullname	name==username	description length	external URL	private	#posts	#followers	#follows	fake
1.0	0.27	0.0	0.0	0.0	53.0	0.0	0.0	32.0	1000.0	955.0	0.0
1.0	0.0	2.0	0.0	0.0	44.0	0.0	0.0	286.0	2740.0	533.0	0.0
1.0	0.1	2.0	0.0	0.0	0.0	0.0	1.0	13.0	159.0	98.0	0.0
1.0	0.0	1.0	0.0	0.0	82.0	0.0	0.0	679.0	414.0	651.0	0.0
1.0	0.0	2.0	0.0	0.0	0.0	0.0	1.0	6.0	151.0	126.0	0.0
1.0	0.0	4.0	0.0	0.0	81.0	1.0	0.0	344.0	669987.0	150.0	0.0
1.0	0.0	2.0	0.0	0.0	50.0	0.0	0.0	16.0	122.0	177.0	0.0
1.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	33.0	1078.0	76.0	0.0

Fig 13. Dataset Overview

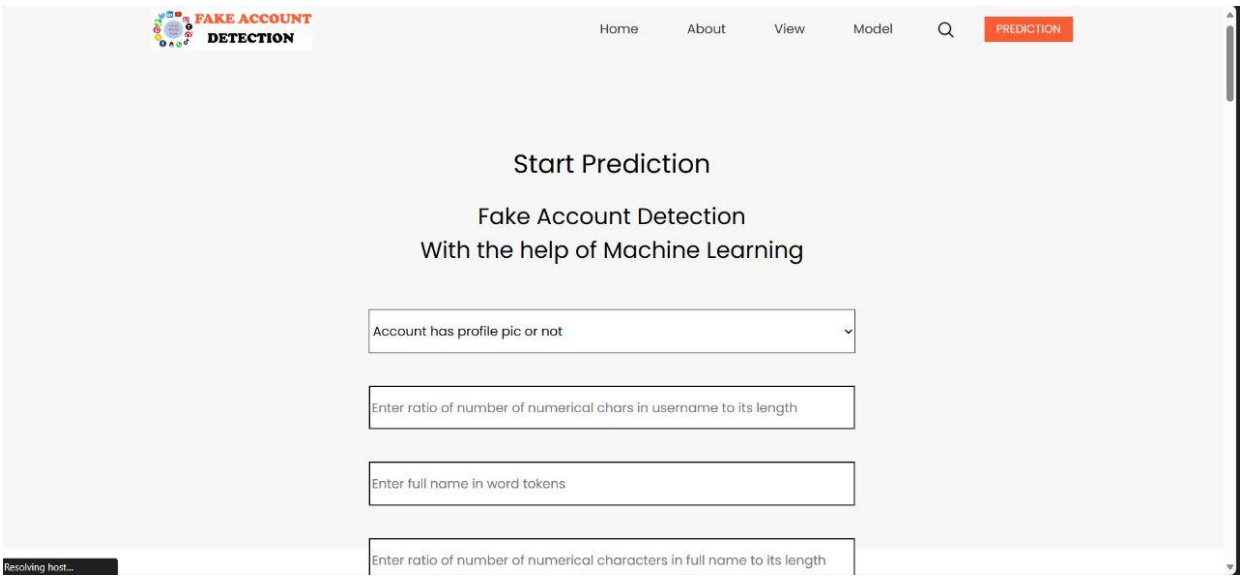


Fig 14. Prediction Page

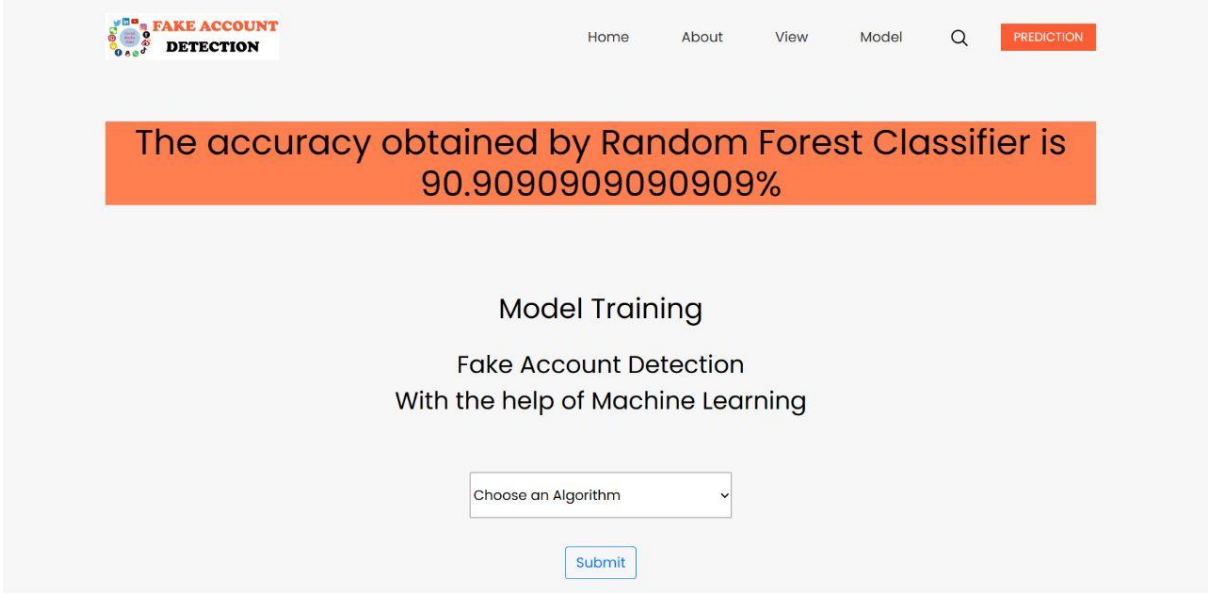


Fig 15. Random Forest Accuracy

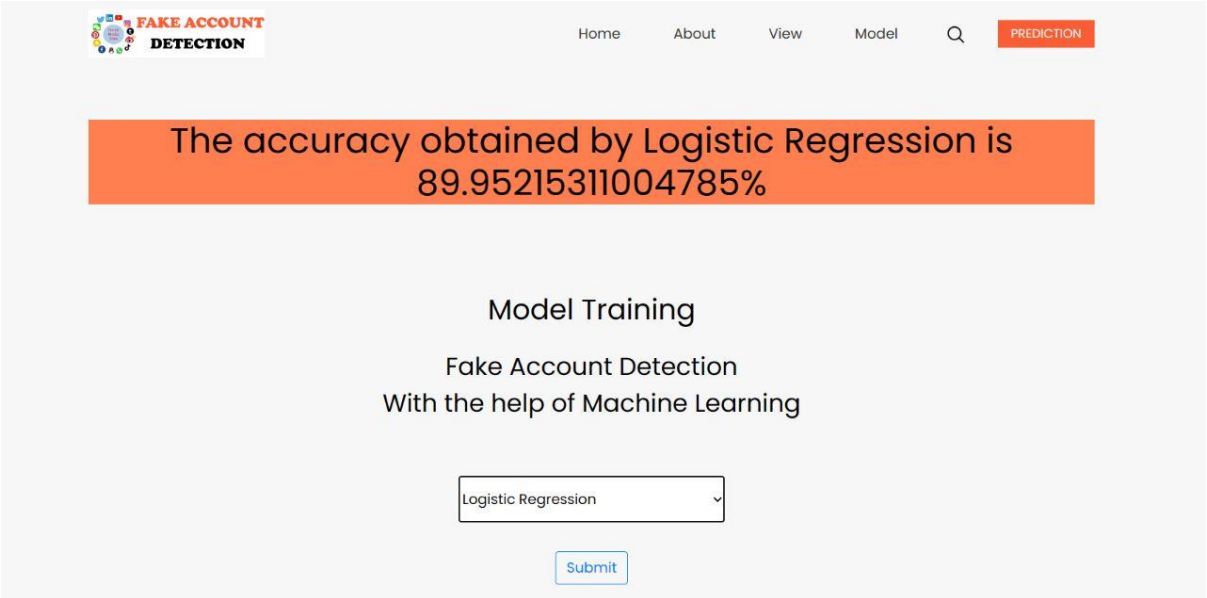


Fig 16. Logistic Regression Accuracy

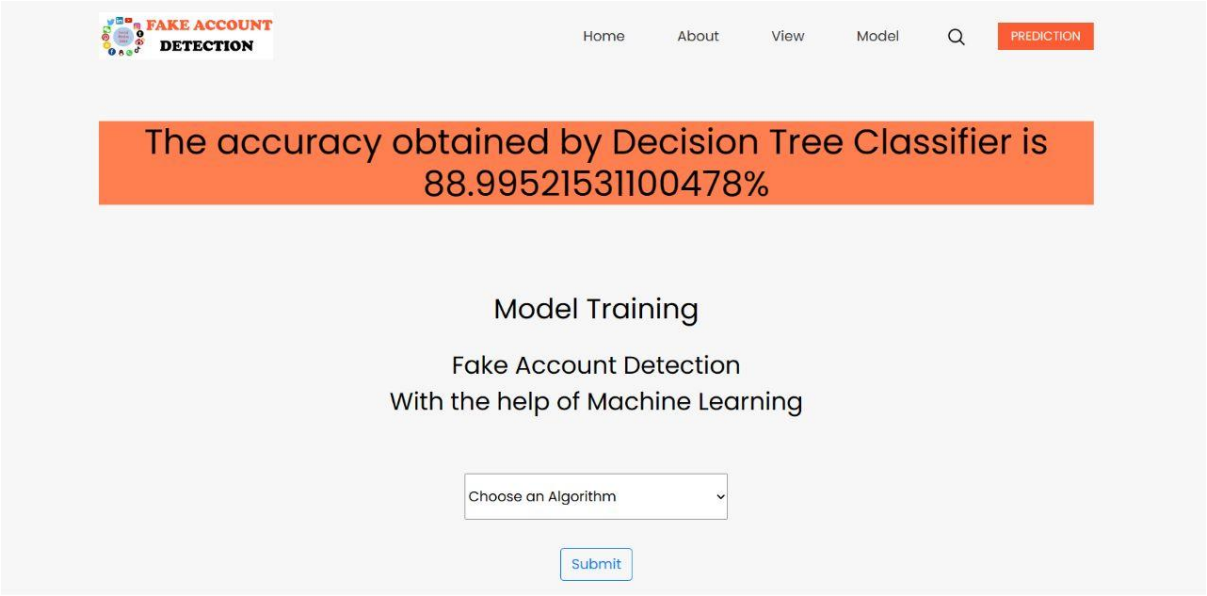


Fig 17. Decision Tree Accuracy

6. Concluding Remarks

In this project, we implemented various stages of development, starting with data preprocessing to ensure data quality and handle missing values. Feature selection techniques were applied to identify the most relevant attributes for the classification task. Training the classifiers on the selected features resulted in accurate classification of fake and genuine accounts. The evaluation metrics provided quantitative measures of the models' performance. The results demonstrated the classifiers' capability to distinguish between fake and genuine accounts with high accuracy and precision.

The integration of the developed models into a Flask web application enabled real-time detection of fake accounts. This web interface enhanced the usability and accessibility of the system, allowing users to easily identify and report suspicious accounts.

Overall, the project's outcomes highlight the potential of Random Forest, Logistic Regression, and Decision Tree classifiers in detecting fake accounts on social media. The accurate identification of fraudulent accounts contributes to maintaining a secure and trustworthy social media environment. Future enhancements, such as incorporating advanced machine learning techniques, analyzing user behavior patterns, and collaborating with social media platforms, can further improve the system's effectiveness and address emerging challenges in the detection of fake accounts.

7. Future Work

Enhanced Feature Engineering: Investigate additional account attributes that may contribute to the detection of fake accounts. Explore the inclusion of text-based features, such as post content, comments, or user interactions, to capture linguistic patterns and behavioral cues.

Advanced Machine Learning Techniques: Consider exploring more advanced machine learning techniques, such as ensemble methods (e.g., Gradient Boosting, XGBoost) or deep learning models (e.g., Recurrent Neural Networks, Convolutional Neural Networks). These models may provide higher predictive accuracy by capturing complex relationships and patterns in the data.

Handling Imbalanced Data: Address the issue of imbalanced data, where the number of genuine accounts outweighs the number of fake accounts or vice versa. Investigate techniques such as oversampling (e.g., SMOTE) or undersampling to balance the data and improve the model's ability to detect both classes effectively.

Incorporating User Behavior Analysis: Consider incorporating user behavior analysis techniques to detect anomalies or suspicious patterns in account activity. This can involve analyzing patterns such as posting frequency, engagement patterns, or sudden changes in behavior that might indicate fraudulent activity.

Real-time Detection and Deployment: Develop a real-time detection system that can monitor social media accounts in real-time and identify potential fake accounts as they are created or become active. This can involve integrating the developed models into a scalable and efficient system that can handle large volumes of data and provide instant results.

Evaluation on New and Diverse Datasets: Validate the developed models on new and diverse datasets from different social media platforms. This will help assess the generalizability and robustness of the models across various contexts and populations.

Explainability and Interpretability: Investigate techniques to improve the explainability and interpretability of the models. This can involve using techniques such as SHAP values or LIME to understand the contribution of each feature towards the model's prediction, providing more transparency and insights into the decision-making process.

Collaboration with Social Media Platforms: Collaborate with social media platforms to leverage their expertise, data, and resources. This collaboration can help refine the models, access additional features or data sources, and contribute to the development of more effective fraud detection systems.

User Feedback and Iterative Improvement: Gather user feedback on the accuracy and usability of the developed system. Incorporate user feedback to refine the models, address any limitations or challenges identified, and continually improve the overall performance and user experience.

References

1. [Online], 2018. Political advertising spending on Facebook between 2014 and 2018. Internet draft. [Online]. Available: https://www.statista.com/statistics/891327/political-advertising spending-face book-by-sponsor-category/
2. [J. R. Douceur], 2002. "The Sybil attack" in the International workshop on peer-to-peer systems. Springer, pp. 251–260.
3. [Online], 2012. Cbc.facebook shares drop on news of fake accounts. Internet draft. Available: http://www.cbc.ca/news/technology/facebook-shares-drop-on news-of-fake-accounts-1.1177067
4. [R. Kaur and S. Singh], 2016. "A survey of data mining and social network analysis based on anomaly detection techniques," Egyptian informatics Journal, vol. 17, no. 2, pp. 199–216.
5. [L. M. Potgieter and R. Naidoo], 2017. "Factors explaining user loyalty in a social media-based brand community," South African Journal of Information Management, vol. 19, no. 1, pp. 1–9.
6. [Online], 2018. Quarterly earning reports. Internet draft. Available: https://investor.fb.com/home/default.aspx
7. [Online], 2018. Statista.twitter: number of monthly active users 2010-2018. Internet draft. Available: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/
8. [Y. Boshmaf, M. Ripeanu, K. Beznosov, and E. Santos-Neto], 2015. "Thwarting fake accounts by predicting their victims," in Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security. ACM, pp. 81–89.
9. [Online], 2018. Facebook publishes enforcement numbers for the first time. Available: https://newsroom.fb.com/news/2018/05/enforcement-numbers/
10. [Online], 2013. Banque populaire dis-moi combien damis tu as sur facebook, je te dirai si ta banque va taccorder un prłt. Internet draft. Available:

http://bigbrowser.blog.lemonde.fr/2013/09/19/popularitedis-moi-combien-damis-tu-as-sur-facebook-je-te-dirai-si-ta-banqueva-taccorder-un-pret/

11. [S.-T. Sun, Y. Boshmaf, K. Hawkey, and K. Beznosov], 2010. "A billion keys, but few locks: the crisis of web single sign-on," in Proceedings of the 2010 New Security Paradigms Workshop. ACM, pp. 61–72.
12. [S. Fong, Y. Zhuang, and J. He], 2012. "Not every friend on a social network can be trusted: Classifying imposters using decision trees," in Future Generation Communication Technology (FGCT), International Conference on. IEEE, 2012, pp. 58–63.
13. [K. Thomas, C. Grier, D. Song, and V. Paxson], 2011. "Suspended accounts in retrospect: an analysis of twitter spam," in Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. ACM, pp. 243–258.
14. [Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu], 2011. "The socialbot network: when bots socialize for fame and money," in Proceedings of the 27th annual computer security applications conference. ACM, pp. 93–102.
15. [J. Ratkiewicz, M. Conover, M. Meiss, B. Gonc¸alves, S. Patil, A. Flammini, and F. Menczer], 2011. "Truthy: mapping the spread of astroturf in microblog streams," in Proceedings of the 20th international conference companion on World wide web. ACM, pp. 249–252.