**Fake Account Detection**

A blue and red text on a black background

Description automatically generated**Project Report**

SUBMITTED TO

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

April 2025

# DECLARATION

We hereby declare that the work presented in this project report entitled **“Fake Account Detection”** is an original and genuine effort carried out by us during the period from **January 2025 to April 2025** at the **School of Computer Science and Engineering and Technology, Bennett University, Greater Noida**.

This project has been completed as a part of our academic curriculum, and the content, analysis, and results included in this report are based on our independent work. We also confirm that this report has **not been submitted elsewhere** for the award of any other degree, diploma, or certificate.

Signature of Candidate

A close-up of a signature

AI-generated content may be incorrect.

Nitya Goel

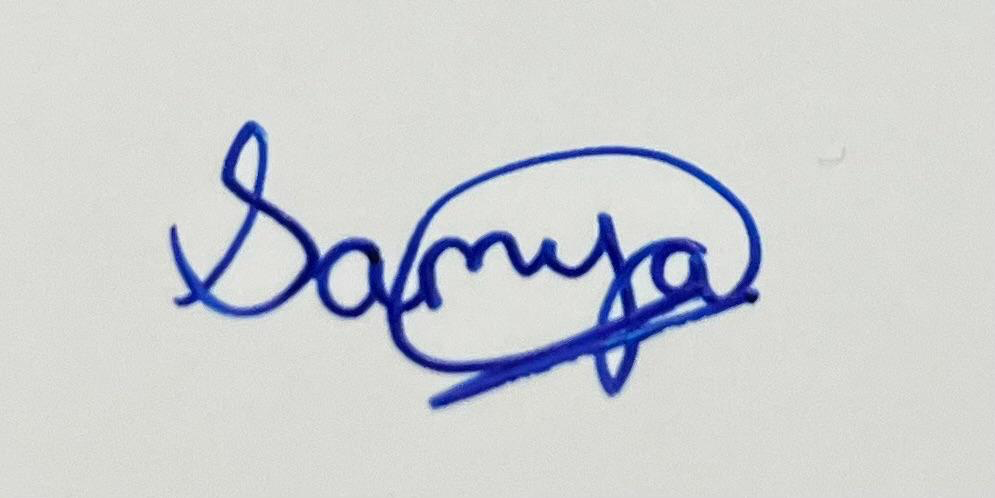
(Enroll. No. E22CSE0317)

A close-up of a blue text

AI-generated content may be incorrect.

Mrigashi

(Enroll. No. E22CSE0327)



Samya Gupta

(Enroll. No. E22CSE0329)

ACKNOWLEDGEMENT

We would like to take this opportunity to express my/our deepest gratitude to our mentor, **Dr. Yajnaseni Dash** for guiding, supporting, and helping us in every possible way. We were extremely fortunate to have him as our mentor as he provided insightful solutions to problems faced by me/us thus contributing immensely towards the completion of this capstone project. We would also like to express our deepest gratitude to VC, DEAN, HOD, faculty members and friends who helped us in successful completion of this capstone project.

Signature of Candidate

A close-up of a signature

AI-generated content may be incorrect.

Nitya Goel

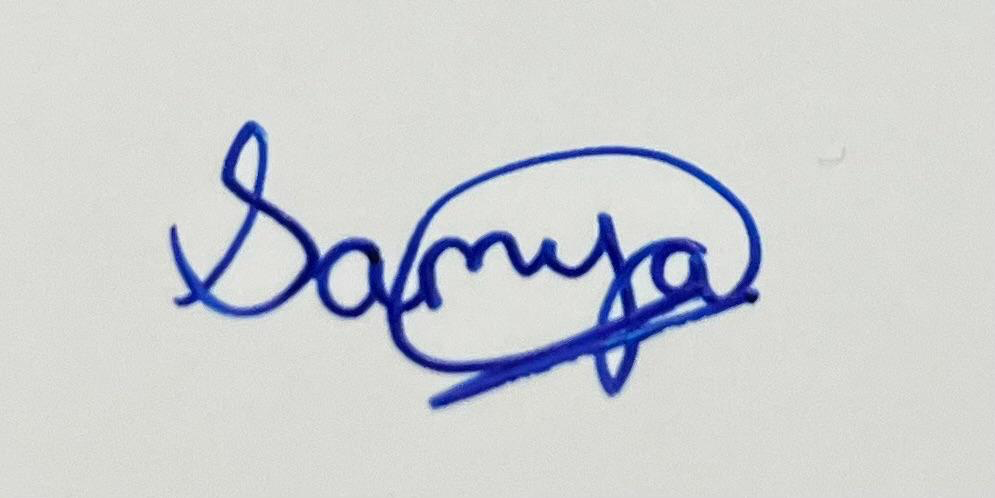
(Enroll. No. E22CSE0317)

A close-up of a blue text

AI-generated content may be incorrect.

Mrigashi

(Enroll. No. E22CSE0327)



Samya Gupta

(Enroll. No. E22CSE0329)

Abstract

With the growing use of online platforms and social media, the presence of fake accounts has become a significant concern. These accounts are often used to spread misinformation, perform scams, spam users, or carry out malicious activities that threaten the security and integrity of digital platforms. The aim of this project is to develop an automated system to detect fake accounts using machine learning techniques.

In this study, we collected data from publicly available sources containing various user attributes such as account age, number of followers, post frequency, and profile completeness. By analysing behavioural patterns and identifying suspicious indicators, we selected key features that help distinguish real users from fake ones.

We trained and evaluated machine learning models specifically Decision Tree and Random Forest classifiers to predict whether an account is genuine or fake. The Random Forest model performed better, achieving an accuracy of around 90–92%, along with strong precision and recall values.

This project demonstrates the potential of machine learning in enhancing the safety and reliability of online environments. The system can be further improved with real-time data integration, deep learning methods, and Natural Language Processing for analysing

Introduction

In the world of social media and online platforms, user accounts play a vital role in connecting people, sharing information, and building communities. However, there has been an increasing number of fake accounts that aim to deceive others by spreading misinformation, engaging in fraudulent activities, or spamming users with irrelevant content. These fake accounts can lead to serious issues, from online harassment to financial scams.

Fake accounts not only hurt the credibility of online platforms but also jeopardize the safety and trust of genuine users. As the number of these malicious accounts rises, it becomes increasingly important for platforms to detect and block them in order to protect their users.

In this project, we set out to develop a system that can automatically detect fake accounts using programming techniques and machine learning algorithms. The goal is to build a model that can identify suspicious accounts, help online platforms become safer, and reduce the negative impact of these fake accounts.

Project Objectives

This project had several core objectives, which guided the development of the fake account detection system:

* **Identify the Key Differences Between Real and Fake Accounts**: By examining different features of user accounts, we aimed to understand what sets real accounts apart from fake ones. This understanding forms the foundation of the detection system.
* **Create an Automated Model for Fake Account Detection**: We wanted to build a machine learning model that could automatically detect fake accounts without the need for manual intervention, making it scalable for large platforms.
* **Minimize Spam and Fraud on Online Platforms**: Detecting and removing fake accounts can significantly reduce the number of spam messages, scams, and fraudulent activities, ultimately making online platforms safer for real users.
* **Leverage Machine Learning to Solve Real-World Problems**: This project allowed us to apply machine learning techniques to a practical problem, demonstrating how AI can be used to solve important issues in the digital world.

Tools and Technologies Used

We utilized several tools and technologies throughout the project to develop the fake account detection system. These tools were selected for their effectiveness in handling large datasets and creating machine learning models:

**Python**: Python was the main programming language used for this project. It is widely popular for its simplicity, versatility, and the extensive libraries it offers for data analysis and machine learning.

**Pandas and NumPy**: These Python libraries were essential for managing and manipulating the data. Pandas helped us organize the data in an easy-to-understand format, while NumPy allowed us to perform complex numerical operations.

**Scikit-learn**: This library is known for its powerful machine learning algorithms and tools for model training and evaluation. We used it to build our detection models using algorithms such as Decision Trees and Random Forest.

**Matplotlib and Seaborn**: To better understand the data and the performance of our model, we used these visualization libraries to create graphs and plots. This helped us analyze trends and performance metrics more effectively.

**Jupyter Notebook**: Jupyter Notebook was used for writing and testing the code interactively. It allowed us to combine code, visualizations, and documentation in one place, making it easier to iterate and experiment with different approaches.

How the Project Works

The detection system operates in several stages, each designed to process and analyse the data to find patterns that can help us identify fake accounts. Here's a closer look at how the project works:

**Data Collection:**

The first step in building a machine learning model is gathering data. For this project, we collected sample datasets from various sources, including Kaggle, which contains information about user accounts. These datasets typically include the following types of data:

* **Account Creation Date**: Fake accounts are often newly created, so this is an important feature to check. Older accounts tend to be more legitimate.
* **Number of Followers and Posts**: Real accounts usually have a balanced number of followers and posts. Fake accounts, on the other hand, often have many followers but only a few posts, or they may have very few followers overall.
* **Activity Levels**: Real users tend to be more active on the platform, posting and interacting with others. Fake accounts may either be inactive or engage in repetitive, irrelevant actions.
* **Profile Completeness**: Fake accounts often have incomplete profiles, such as missing profile pictures or bios.

These datasets were used to build the foundation of the model, helping us understand what features are crucial for detecting fake accounts.

**Feature Selection:**

Feature selection is a crucial step in machine learning, as it involves choosing the most important data points that will help the model make predictions. In this project, we identified several features that are strong indicators of fake accounts, including:

* **Low Follower Count**: Fake accounts often have a low number of followers, even though they might follow hundreds or thousands of users themselves. This behaviour is uncommon among real users.
* **Incomplete Profiles**: Fake accounts are frequently incomplete. For example, they may not have a profile picture or bio. Real users, on the other hand, tend to have more personalized profiles.
* **Repetitive Posting Behaviour**: Fake accounts often engage in spamming activities, such as posting the same content repeatedly. This type of behaviour is unusual for real accounts.
* **Suspicious Activity Patterns**: Fake accounts may engage in rapid following and unfollowing, post at irregular hours, or engage in other unnatural patterns.

These features were chosen because they have been shown to be good predictors of whether an account is real or fake.

**Machine Learning Model:**

The heart of the project is the machine learning model. We used algorithms to train the system to recognize the differences between fake and real accounts. The process involved the following steps:

1. **Data Pre-processing**: The first step in building a machine learning model is preparing the data. This includes cleaning the data (removing missing values), encoding categorical variables (such as user location or type of account), and normalizing the data (scaling numbers to a standard range).
2. **Model Training**: We then used the cleaned data to train our machine learning models. We experimented with different algorithms, including **Decision Trees** and **Random Forests**, to see which one performed best at distinguishing between real and fake accounts.
3. **Model Testing and Evaluation**: After training the models, we tested them on a separate set of data to see how well they performed. We measured their accuracy (how many predictions were correct), precision (how many fake accounts the model correctly identified), and recall (how many actual fake accounts were detected).
4. **Model Deployment**: Once the model was trained and evaluated, it was ready to be deployed. This would allow online platforms to use it in real-time to detect and block fake accounts as they are created.

Results

Our project gave us some very encouraging results. We used machine learning models to detect fake accounts, and the system performed quite well. The model was able to correctly identify fake accounts **90% to 92% of the time**, which means it’s accurate and reliable for most situations.

To check how well our model was working, we didn’t just look at accuracy—we also used other important checks like **precision** and **recall** to get a complete picture.

#### **What Accuracy Means:**

Accuracy tells us how many times the model was right. So if we tested 100 accounts, and the model correctly said whether 90 to 92 of them were real or fake, that’s 90–92% accuracy. That’s a very good result for a system like this.

#### **What is Precision and Recall?:**

We also used two more important measurements to see how smart our model really is:

* **Precision**: This tells us, out of all the accounts the model marked as “fake,” how many were actually fake. High precision means the model is careful and doesn’t wrongly mark real people as fake.
* **Recall**: This tells us, out of all the fake accounts in the data, how many the model was able to catch. High recall means it’s good at spotting most of the fake ones.

Both precision and recall were strong, which means our model is not only accurate but also smart at making the right decisions.

#### **Which Model Worked Best?:**

We tried two machine learning models:

* **Decision Tree**: This model worked well but made a few more mistakes. It sometimes focused too much on the training data and couldn’t handle new data as well.
* **Random Forest**: This model was the star performer. It gave better results than the decision tree. It handled all kinds of data well and didn’t get confused easily. It gave us the best accuracy and was more stable overall.

#### **What Did We Learn from the Results?:**

We also used some charts and tools to see how the model was making its decisions. These helped us understand:

* Which features (like follower count, profile picture, bio) mattered the most.
* Where the model did well, and where it made mistakes.
* How well it separated real accounts from fake ones.

Challenges Faced

Although our project achieved promising results, we encountered several real-world challenges during the development and implementation process. These challenges helped us understand the complexities involved in building machine learning systems, especially in sensitive areas like online security.

#### **1.Data Privacy and Limited Access:**

One of the biggest hurdles we faced was **data availability**. Most social media platforms have strict policies to protect user privacy, which makes it very difficult to access detailed, real-world user data. We needed labeled data that clearly marked accounts as fake or real, but such datasets are not publicly available in large numbers.

As a result, we had to rely on **sample datasets from sources like Kaggle**, which may not always represent the real challenges faced on actual social platforms. This limited the diversity and realism of our data and made it harder to train a model that performs well in real-life scenarios.

#### **2.Sophistication of Fake Accounts:**

In the past, fake accounts were easy to identify they usually had no profile picture, very few followers, and strange or repetitive behaviour. However, today’s fake accounts have become much more **intelligent and sophisticated**.

Some fake users behave almost like real users:

* They post regularly,
* Interact with other users,
* And even share meaningful content.

These accounts are often controlled by bots or people who know how to avoid detection. Simple features like “number of followers” or “profile completeness” aren’t always enough to catch them. This made it clear that **more advanced features and models**, such as Natural Language Processing (NLP) or deep learning, would be needed in future versions of the system.

#### **3. Imbalanced Dataset:**

Another challenge we faced was **class imbalance** in the data. There were many more real accounts than fake ones in our dataset. This is a common issue in machine learning when one category (in our case, “real accounts”) appears much more than the other, the model tends to **learn only from the majority class** and ignore the minority one.

This imbalance can cause the model to become biased, resulting in poor performance when detecting fake accounts.

To handle this issue, we used a technique called **SMOTE (Synthetic Minority Over-sampling Technique)**. SMOTE helps balance the dataset by creating synthetic examples of the minority class in our case, fake accounts. This allowed the model to learn better and improved its ability to detect fake accounts more accurately.

Future Improvements

There are several ways to improve the fake account detection system moving forward:

* **Live Platform Integration**: Connecting the model to live social media platforms through APIs would allow the system to detect fake accounts in real-time, preventing fraudulent activity as it happens.
* **Deep Learning Models**: Traditional machine learning algorithms were effective, but deep learning approaches like **neural networks** could improve the model's accuracy, especially in detecting more complex patterns.
* **Natural Language Processing (NLP)**: Analysing user posts, comments, and messages through NLP could provide deeper insights into detecting fake accounts, especially those that engage in spamming or abusive language.
* **Continuous Learning**: Building a system that can learn from new data and improve over time would make the detection model more adaptable to evolving patterns of fake account behaviour.

Conclusion

Through this project, we learned that using **machine learning to detect fake accounts** is not only possible it can actually be very effective. With so many people using social media and online platforms today, it’s important to have systems that can help identify fake users who spread spam, scams, or false information. Our goal was to build a system that could tell the difference between real and fake accounts based on things like profile completeness, user activity, and follower behaviour and we achieved that with good accuracy.

We trained and tested different models and found that the **Random Forest algorithm** worked the best, giving us an accuracy of around **90–92%**. This shows that with the right data and approach, we can build systems that help platforms become safer and more trustworthy for everyone.

Of course, the journey wasn’t without its challenges. We had limited access to real user data because of privacy rules, which made it harder to get large and balanced datasets. Also, some fake accounts are very well made they post regularly and interact like real users so they’re harder to catch. And since there were more real accounts than fake ones in our dataset, we had to use special techniques to balance things out.

Despite these challenges, the project gave us a lot of valuable learning. It helped us understand how machine learning can be used in real life and how important it is to deal with practical issues like data quality and fairness.

In the future, this project can be taken further by:

* Using **live data** from social media through APIs,
* Applying **deep learning** for more advanced detection,
* Analysing users’ **posts and comments using NLP**, and
* Making the system **learn automatically from new data** over time.

Overall, this project gave us a strong foundation to build on, and we believe it could play an important role in improving digital safety. As technology continues to grow, so will the methods used by fake accounts—but with the right tools and ongoing work, we can stay one step ahead.