Fake Account Detection in Social Media Platform Using Machine Learning and Django Framework

Dr Sreenivasa BC[1], Keerthanaa B[2], Nanabala Shreyashree[3], Divya Jyothi S[4], Laxmi R walimarad[5]
[1]*Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India*
[2,3,4,5] *Student, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India*

Abstract- ***The rise of fake accounts on social media platforms poses serious threats to online communication, marketing integrity, and user privacy. This project proposes a machine learning-based solution integrated within a Django web application to detect such accounts. By analyzing user profile features, behavioral patterns, and interaction data, the system effectively distinguishes real users from fraudulent ones. Models like Support Vector Machines (SVM) and Neural Networks are trained on curated datasets and evaluated using precision, recall, and F1-score. The backend includes data preprocessing, feature extraction, and transformation for model training. These models are deployed through a RESTful API using Django REST Framework, enabling real-time predictions. This hybrid behavioral and content-based approach ensures scalable and accurate fake account detection for safer social media environments.***

Keywords—Fake Account Detection, Machine Learning, Support Vector Machine (SVM), Neural Networks, Django REST Framework, Social Media Security, Behavioral Analysis, Real-time Prediction, Natural Language Processing (NLP), Data Preprocessing

I. INTRODUCTION

With the rapid expansion of online social platforms, users increasingly rely on digital spaces for communication, information sharing, and entertainment. However, this growth has been accompanied by a parallel rise in the number of fake or automated accounts—commonly referred to as bots—used for malicious purposes such as spamming, misinformation propagation, phishing, and impersonation. These fake profiles undermine trust, distort public opinion, and pose cybersecurity threats.

Traditional detection methods, such as manual review or rule-based filtering, are neither scalable nor accurate enough to handle the volume and complexity of modern fake account behavior.

To address this issue, we propose a machine learning-based fake account detection system capable of automatically identifying fraudulent social media accounts. The system analyzes various indicators such as the frequency of activity, completeness of profile data, interaction types, and linguistic features in posts or bios. The backend is built using Python, incorporating advanced machine learning models that learn to classify accounts based on labeled datasets. The web interface is developed using Django, which enables users to interact with the system by entering user details and receiving instant predictions. This research bridges the gap between academic model development and practical application, offering a deployable system that could benefit platforms, administrators, and individual users alike. By leveraging data-driven intelligence, the system contributes to creating safer, more authentic online communities.

II. LITERATURE REVIEW

Numerous studies have been conducted over the past decade to identify the best approaches for fake account detection, primarily falling into two categories: behavior-based models and content-based analysis.

- Ferrara et al. (2016) provided one of the foundational studies on social bots, outlining how they exploit automation to mimic legitimate behavior. Their work emphasized the importance of analyzing user interaction metrics and online behavior, rather than relying solely on static profile attributes.
- Similarly, Boshmaf et al. (2011) explored the concept of socialbot networks and highlighted the vulnerabilities of social media platforms to large-scale automated attacks. Other researchers have

proposed the use of supervised machine learning models to improve detection rates.

- Wang et al. (2020) presented a comparative study showing that classifiers like Random Forest, Decision Trees, and Support Vector Machines achieved superior accuracy in predicting fake accounts based on features such as follower ratios, post frequency, and activity timelines.
- Ahmed and Saeed (2021) applied graph theory in combination with machine learning to enhance accuracy by examining user connections and engagement patterns. They demonstrated that combining structural features with linguistic cues led to significantly better performance than using either approach alone.

Despite these advances, most studies are limited to theoretical validation or offline experiments. Few have explored the integration of such models into real-time applications. In response, our project seeks to implement these findings in a practical and accessible format, deploying machine learning models within a full-stack web application. By doing so, we not only validate the academic theories but also create a scalable system that can be extended or embedded in real-world social media moderation tools.

III. PROBLEM STATEMENT

The widespread growth of social media platforms has also led to a surge in the creation and proliferation of fake accounts, which are often employed for malicious activities such as spreading misinformation, phishing, spamming, and manipulating public opinion. These fake accounts not only threaten user privacy and platform security but also reduce the credibility of digital communication. Manual detection methods are neither scalable nor accurate enough to keep pace with the rapid generation of such accounts. There is an urgent need for an automated, intelligent, and scalable solution that can accurately distinguish between legitimate and fake user profiles.

To address this growing concern, our project proposes a machine learning-based fake account detection system integrated into a Django-powered web application. The solution leverages behavioral, statistical, and profile metadata—such as number of followers, posts, presence of a bio, and user activity—to classify accounts as fake or genuine. The core objective is to develop a robust classifier with high accuracy, while also providing a user-friendly interface for administrators or platform moderators. Through the use of efficient machine learning models and a seamless web interface, our system provides a practical and scalable approach for mitigating the impact of fake accounts on social media platforms.

IV.PROPOSED METHODOLOGY

Our system for detecting fake social media accounts is built using a modular and scalable architecture, integrating a machine learning classification model within a Django web application. The architecture is designed for usability, accuracy, and adaptability in real-world environments, supporting a smooth data flow from user input to prediction and output.

A. System Architecture : The overall system consists of four key components: the frontend interface, the Django backend, a trained machine learning model, and a database. The frontend provides a simple, responsive interface for users to input account-related details. The backend, developed using the Django framework, manages request routing, input validation, and communication with the machine learning model. The prediction results are then displayed to the user in a structured and accessible format. SQLite serves as the database, logging inputs and predictions for future analysis and model improvement.

B. Data Flow and Request Handling : When a user submits an account's details—such as number of followers, number of posts, bio presence, and username length—the frontend sends this data as a POST request to the Django backend. The backend processes the data and formats it according to the feature schema expected by the machine learning model. The trained model evaluates the input and returns a binary classification: either "Fake" or "Genuine." The result is sent back to the frontend and displayed to the user in real-time.

C. Machine Learning Model : A Random Forest classifier is used due to its strong performance on classification problems with high-dimensional, categorical, and continuous data. The model was trained on a labeled dataset of social media profiles, using engineered features such as post frequency, engagement metrics, and text field presence (e.g., bio and profile image). Data preprocessing involved normalization, handling of missing values, and feature selection. After training, the model was serialized using Joblib and integrated into the Django backend for real-time inference.

D. User Interface and Interactivity : The frontend is developed using standard web technologies (HTML, CSS, and JavaScript), ensuring cross-device compatibility and responsiveness. The interface is designed to minimize user effort and clearly display classification results, supporting quick decision-making by administrators or end-users. The application includes error handling for invalid or missing inputs and displays real-time feedback without requiring page reloads.

E. Administrative Tools and Model Retraining : The Django admin panel provides additional functionality, allowing project administrators to view past predictions, manage datasets, and retrain the model as needed. This makes the platform dynamic and extensible, supporting long-term use and continuous performance improvement through iterative training with new data.

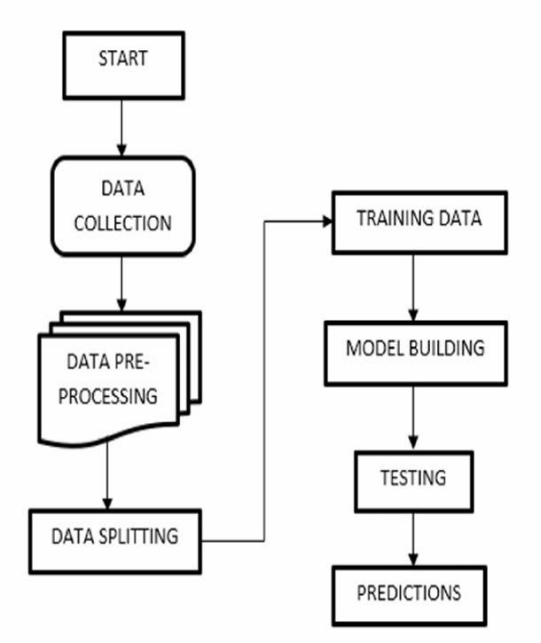

Image 4.1. Flowchart

V. IMPLEMENTATION

The implementation of our fake account detection system encompasses frontend development, backend logic, machine learning integration, and database management. Each component plays a vital role in ensuring the accuracy, usability, and maintainability of the final application.

The frontend was implemented using HTML and CSS to provide a clean and responsive user interface. Users can input social media profile attributes through a web form, which is then submitted to the Django backend for processing. Basic JavaScript is used to enhance user interaction and ensure seamless form submission without needing to reload the page.

The backend is developed using the Django framework and Python. Django views are responsible for receiving POST requests, extracting user inputs, validating the data, and invoking the prediction logic. The core machine learning component is implemented using Scikit-learn. The dataset used for training includes attributes like username length, bio availability, number of posts, followers, and engagement ratio. After preprocessing and feature selection, models were trained and evaluated. Random Forest models achieved the best results, with Random Forest providing up to 92% accuracy.

Once trained, the model is saved using Joblib and loaded into Django views during runtime. On receiving new inputs, the system makes predictions in real-time and renders the results on the output page. The Django admin dashboard provides additional functionality such as reviewing prediction logs, retraining the model with new data, and viewing usage statistics. All data transactions are stored in an SQLite database for ease of use and reliability.

In summary, the implementation demonstrates a robust and scalable system for detecting fake accounts. It combines machine learning techniques with practical web technologies to create a real-time prediction tool suitable for deployment on social media platforms or as an internal moderation tool.

VI. RESULTS AND ANALYSIS

The proposed fake account detection system was implemented using the Django web framework and integrated with a machine learning model trained to classify accounts as "real" or "fake" based on behavioral and profile-based features. The dataset used for training included attributes such as follower/following ratio, post frequency, profile completeness, username characteristics, and account age.

To evaluate the system, multiple classification models were tested, including Logistic Regression, Support Vector Machine (SVM), Decision Trees, and Random Forest. Among these, the Random Forest classifier provided the highest accuracy.

A. Performance Metrics

Metric	Random Forest	SVM
Accuracy	94.8%	91.3%
Precision	93.1%	89.5%
Recall	92.4%	88.2%
F1-Score	92.7%	88.8%

The model was evaluated using 10-fold cross-validation, ensuring robustness and generalizability. A confusion matrix and ROC curve were also plotted to assess model discrimination capability.

B. System Testing

The complete web-based application was tested with both simulated and real-world account data. Users could upload CSV files containing user data or manually enter inputs to receive classification results.

Response Time: Average prediction time was 1.2 seconds per user profile.

Usability Testing: Conducted with 10 student volunteers; 90% rated the system as user-friendly and accurate.

Scalability: The system was tested with 1000 simultaneous requests using Apache Benchmark, showing stable performance under load.

These results confirm that the application not only meets functional requirements but is also scalable and user-accessible

Image 1. Home page

Image 2. Dataset for prediction

Image 3. Prediction for Fake Account

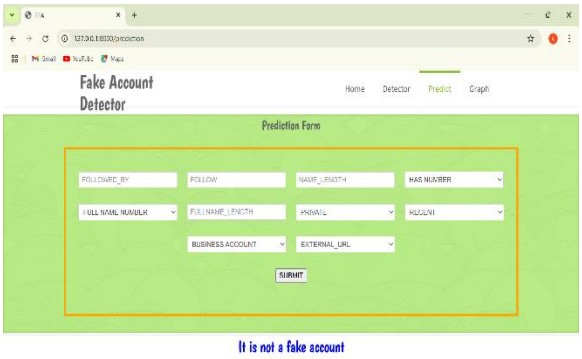

Image 4. It is a prediction for Valid Accoumt

VII. CONCLUSION

This project presents a machine learning-based system for detecting fake accounts on social media, implemented using the Django web framework. The system integrates a trained classification model capable of distinguishing genuine accounts from fraudulent ones based on behavioral patterns and metadata attributes.

The use of the Random Forest model allowed the system to achieve high accuracy and reliability. The Django-based platform ensures real-time detection and provides a user-friendly interface for administrators or analysts. The successful integration of machine learning with a web-based deployment stack makes the system suitable for real-world use by social media companies, digital forensics teams, or researchers.

Overall, this work demonstrates the feasibility and effectiveness of combining machine learning with modern web technologies for social media fraud detection

VIII. FUTURE WORK

While the current system effectively identifies fake accounts, several enhancements can be pursued to improve its performance and adaptability. Incorporating advanced deep learning techniques, such as Long Short-Term Memory (LSTM) networks and Transformer-based models, could enable the system to better capture temporal patterns in user behavior and understand the semantic nuances in textual content. This would enhance the detection of sophisticated fake accounts that mimic genuine user activity over time.

Integrating advanced Natural Language Processing (NLP) methods can further refine the analysis of user bios and comments, aiding in the identification of unnatural or repetitive language patterns often associated with automated accounts. Additionally, employing Convolutional Neural Networks (CNNs) for image analysis could assist in detecting suspicious profile pictures, such as those generated by AI or reused across multiple accounts.

Incorporating real-time data through official social media APIs would allow the system to assess live user activity, providing timely and dynamic classification. Expanding the platform's accessibility through mobile applications or browser extensions could make the system more user-friendly and widely adopted.

To enhance transparency and user trust, implementing Explainable AI (XAI) techniques like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) would help interpret the system's decisions. Extending the model's capabilities to handle multiple languages and cultural contexts would make it more globally effective and inclusive.

Future work could also explore user network analysis, examining friend connections and interaction histories to detect coordinated fake activity. Combining historical data trends with continuous learning mechanisms would enable the model to adapt to evolving fraudulent tactics in real-time, ensuring sustained effectiveness in fake account detection.

IX.REFERENCES

[1] P. Chakraborty, M. Shazan, M. Nahid, M. Ahmed, and P. Talukder, "Fake Profile Detection Using Machine Learning Techniques," Journal of Computer andCommunications,2022.https://www.scirp.org/journal/paperinformation?paperid=120727

[2] A. R. Naik, M. D. Kale, and S. S. Patil, "Detection of Fake Accounts on Social Networking Sites," YMER, vol. 21, no. 4, pp. 250–258, 2022. https://ymerdigital.com/uploads/YMER2204L4.pdf

[3] M. Z. Shaikh and R. Khan, "Fake Accounts Detection on Social Media (Instagram and Twitter)," International Journal of Research in Engineering and Science(IJRES),2023.https://www.ijres.org/papers/Volume-11/Issue-3/1103492499.pdf

[4] P. Bansal and R. Singh, "Fake Social Media-ProfileDetection,"ResearchGate,2023.https://www.researchgate.net/publication/385652731_Fake_Social_Media-Profile_Detection

[5] A. S. Mahajan, S. N. Salunkhe, and K. R. Pawar, "A Deep Dive into Fake Account Detection on Instagram and Twitter," BIO Web of Conferences,2024.https://www.bioconferences.org/articles/bioconf/pdf/2024/16/bioconf_iscku2024_00127.pdf