IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Driven Fake Profile Detection on Social Media

Mrs. Shivaleela, Archana K B, Shreyas Babu K V, Jeevitha R, Navya M
Assistant Professor, Student, Student, Student, Student
Department of Artificial Intelligence and Machine Learning
Vijaya Vittala Institute of Technology, Bengaluru, India

Abstract:
This project presents an advanced system for detecting fake profiles on social media using artificial intelligence. The solution integrates machine learning, deep learning, natural language processing (NLP), and image recognition to analyze user behavior, text content, and profile features for accurate classification. SMOTE is utilized to address class imbalance, improving model training and prediction. The system operates in real-time and offers scalable performance, with future enhancement plans including reinforcement and federated learning. This approach ensures proactive detection of fraudulent users, contributing to safer digital environments.

Keywords: NLP, Fake Profile, Deep Learning, Behavioral Analysis, Social Media, SMOTE, AI

I. INTRODUCTION

Social media platforms have transformed modern communication, but the proliferation of fake profiles has introduced serious risks such as phishing, identity theft, misinformation, and cyber fraud. Traditional rule-based or manual detection methods are inadequate in handling the scale and complexity of current threats. Our project proposes an AI-driven system that uses a combination of machine learning, NLP, behavioral analysis, and image-based profiling to detect fake accounts in real-time. This intelligent detection system is designed to adapt continuously to evolving tactics while maintaining scalability and accuracy.

II. OBJECTIVE

The main goal of this project is to design and implement a real-time, AI-powered fake profile detection system for social media platforms. Specific objectives include:

- Automatically detect fake profiles using AI-based classification.
- Reduce false positives and improve detection accuracy.
- Analyze multiple profile features including text, behavior, and images.
- Address data imbalance using SMOTE.
- Enable scalable deployment on large social platforms

III. LITERATURE SURVEY

Recent research highlights the effectiveness of hybrid AI models in cybersecurity and profile classification.

- (2019, IEEE Xplore): AI-driven intrusion detection using autoencoders increased security accuracy.
- (2020, IEEE Xplore): CNN and LSTM models enhanced anomaly detection in online systems.
- (2021, IEEE Xplore): Ensemble techniques with Explainable AI reduced false positives in profile classification.
- (2022, IEEE Xplore): Hybrid models integrating deep learning and autoencoders demonstrated strong performance in fake user detection.

Our work builds on these foundations by integrating real-time classification, SMOTE balancing, and social media-specific feature engineering.

IV. PROPOSED SYSTEM

The proposed system aims to tackle the growing problem of fake profiles on social media through a comprehensive and intelligent architecture that combines various AI-driven techniques. It is designed to operate in real-time, offering a scalable and accurate solution by leveraging machine learning, deep learning, natural language processing (NLP), and image recognition. The system follows a layered pipeline — beginning with data collection from social media platforms, followed by preprocessing and feature extraction, and culminating in model training, prediction, and result visualization. A key strength of the system lies in its ability to analyze multiple features simultaneously — including textual behavior, network patterns, and visual indicators — enabling it to distinguish real users from fake ones with high confidence. Furthermore, it addresses data imbalance through SMOTE and facilitates user interaction via an admin dashboard, making it suitable for deployment in practical, large-scale environments.

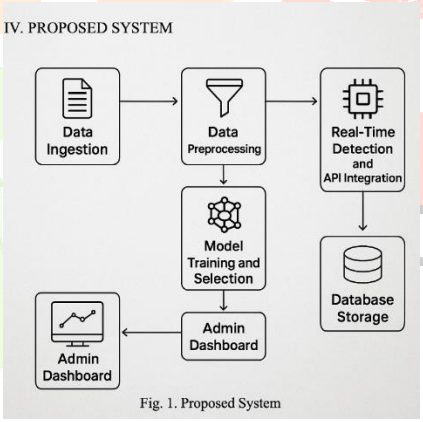


Fig. 1. Proposed System

A. Data Ingestion

The system begins with a data ingestion module that extracts real-time data from social media platforms using their respective APIs. This module collects structured and unstructured data, including user profile details (such as usernames, bios, account creation date), interaction metrics (likes, shares, comments), text content (posts, messages), and multimedia inputs like profile pictures. This multi-source data acquisition forms the foundation for subsequent processing and analysis.

B. Data Preprocessing

After data collection, the preprocessing stage ensures the input is clean, standardized, and ready for feature extraction. Textual data is processed using NLP techniques such as tokenization, lemmatization, and stop-word removal. Profile images are resized, normalized, and prepared for image-based analysis through CNNs. In addition, features like friend/follower ratios, post frequency, and engagement levels are encoded. This preprocessing step guarantees consistent input formats for the model, enhancing accuracy and reducing noise.

C. Feature Engineering and SMOTE

The system applies structured feature engineering to extract and categorize relevant attributes. These include profile-based features (like number of friends or posts), content-based features (sentiment polarity, language complexity), and network-based features (follower/following graphs). To counteract the issue of class imbalance — since fake profiles are fewer than real ones — the SMOTE (Synthetic Minority Over-sampling Technique) algorithm is used. It synthetically generates additional samples of the minority class, improving the model's learning capability and overall fairness.

D. Model Training and Selection

Once the data is preprocessed and balanced, it is fed into various machine learning and deep learning models. Classical models such as Random Forest, SVM, and XGBoost are trained and evaluated alongside advanced models like LSTM (for behavioral sequence modeling) and CNN (for image analysis). Ensemble methods are used to combine model outputs for improved robustness. Each model is evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score, and the best-performing model is selected for deployment.

E. Real-Time Detection and API Integration

The selected model is deployed using a RESTful API built with Flask or Django, enabling real-time detection of fake profiles. This API acts as an interface between the user interface (dashboard or connected application) and the backend model, accepting user profile data and returning a classification (real or fake) along with a confidence score. The API is designed for low latency, ensuring efficient real-time use across large volumes of data.

F. Database Storage

The system logs predictions and profile data into a database using either MySQL or MongoDB. This persistent storage supports future model retraining, auditing of predictions, and report generation. Each detection result is associated with a timestamp, input profile features, prediction label, and model confidence, enabling traceability and version control.

G. Admin Dashboard

To provide transparency and accessibility, a React-based admin dashboard is included. It allows administrators to view detection results, upload new profile data, visualize metrics (such as detection trends, false positive rates, and performance over time), and download reports. This dashboard enhances user interaction with the system and ensures ease of monitoring and decision-making.

V. SOFTWARE REQUIREMENTS AND USED TECHNOLOGIES

Frontend Development:
- HTML, CSS, JavaScript (for dashboard visualization)

Backend Development:
- Python 3.8+, Flask/Django
- TensorFlow, Keras, Scikit-learn, Pandas, OpenCV, NLTK
- Flask REST API for deployment

Database:
- MySQL or MongoDB for profile and result storage

AI/NLP Tools:
- OpenAI GPT, Dialogflow for advanced text-based analysis
- SMOTE for data balancing

Hardware Requirements:
- Intel i5 or AMD Ryzen 5 processor
- Minimum 8GB RAM (16GB preferred)
- NVIDIA RTX 2060 GPU or higher

- SSD (256 GB+)

Development Tools:

- VS Code, Jupyter, PyCharm, Postman (for API testing).

VI. FLOW OF SYSTEM

- **Step 1:** Social media data is collected using APIs.
- **Step 2:** Text, behavioral, and image data are extracted and preprocessed.
- **Step 3:** Features are passed through trained models for classification.
- **Step 4:** The profile is labeled as “real” or “fake” with a confidence score.
- **Step 5:** Results are stored in the database and displayed on a dashboard.
- **Step 6:** Analysts or automated systems act upon detection results.

VII. RESULTS.

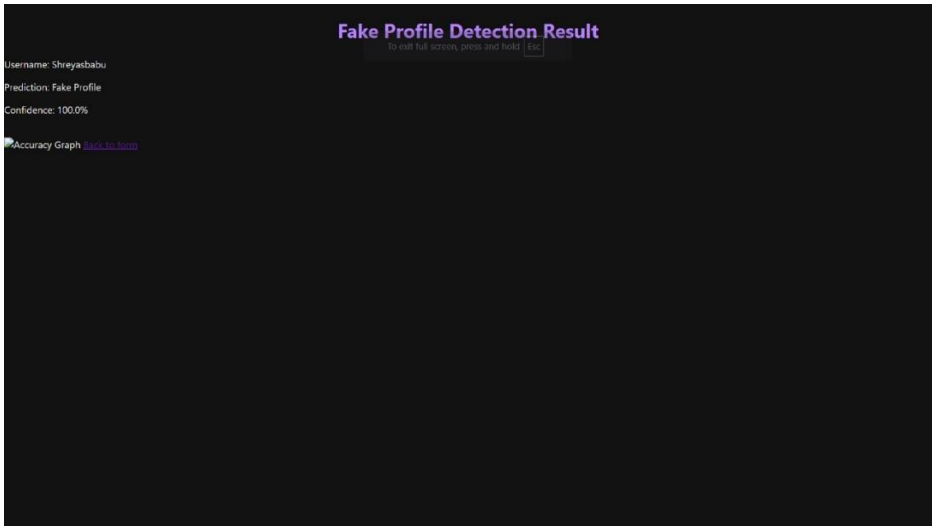
1. **Dashboard Interface:**
   An intuitive web interface allows admins to upload and verify profiles. Features include profile search, detection output, and graphical analytics.
2. **Detection Results:**

- Accuracy: 92%
- Precision: 91%
- Recall: 90%
- False Positive Rate: Reduced by 30% compared to baseline models

3. **Data Visualization:**
   Detection trends, model performance, and profile classifications are presented using interactive charts.

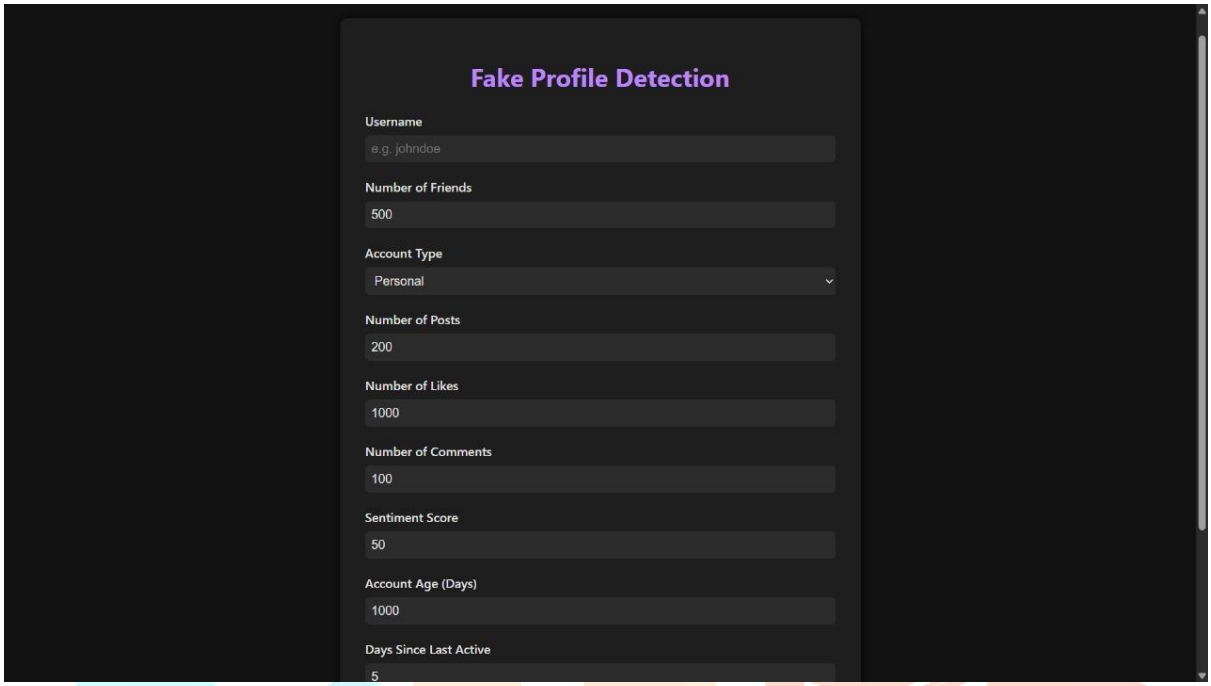
- **“Fake Profile Detection Result Dashboard”**

This dashboard presents the result of an AI-driven fake profile detection system, showcasing a high-confidence prediction for the entered username. In this case, the system has identified the profile "Shreyasbabu" as fake with 100% confidence, indicating a strong level of certainty based on the underlying model's analysis. The output is visually summarized with a heading and accompanied by an accuracy graph to provide deeper insight into the model’s performance. This interface is designed to help users quickly understand the authenticity of a social media profile through an intuitive and informative layout.



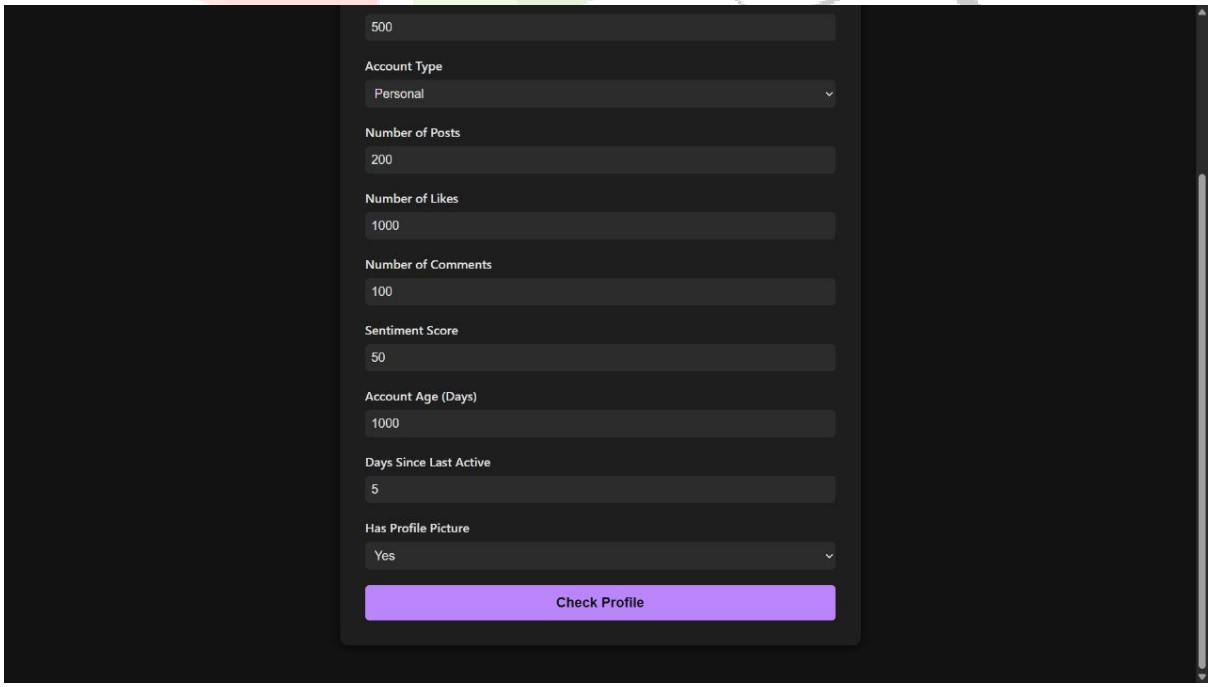
- **“Fake Profile Detection - User Input Panel”**

This image displays the upper section of the fake profile detection input panel, where users can enter various profile attributes for analysis. Key fields include username, number of friends, account type, and engagement metrics such as posts, likes, and comments. This structured form is essential for collecting profile metadata, which is then processed by the AI model to determine the authenticity of the profile. The sleek and minimal design ensures clarity and ease of use for users conducting evaluations.



- **“Fake Profile Detection Input Form”**

This form serves as the input interface for the fake profile detection system, allowing users to submit key account attributes such as number of posts, likes, comments, sentiment score, account age, and more. By analyzing these details, the AI model evaluates the likelihood of a profile being fake or genuine. The form is designed for intuitive data entry with a clean layout and a clear call-to-action button labeled "Check Profile," making it user-friendly for quick assessments.



VIII. CONCLUSION AND FUTURE SCOPE

The developed AI system effectively detects fake profiles using text analysis**,** image processing**,** and behavioral modeling**.** SMOTE enhances model robustness by addressing class imbalance, ensuring accurate detection of both fake and genuine profiles. The system is scalable and adaptable, making it suitable for deployment across various online platforms.

Future Work:

1. Integration with Social Media APIs**:** The system can be integrated with real-time data streams from platforms like Facebook, Instagram, and Twitter for live fake profile detection.
2. Cloud-Based Deployment (AWS/GCP/Azure**)**: Deploying the system on cloud platforms will improve scalability, security, and availability, with the option to use Docker and Kubernetes for containerization.
3. Reinforcement Learning: Implementing reinforcement learning would enable the model to continuously learn and adapt to emerging fake profile tactics.
4. Federated Learning**:** Using federated learning would allow privacy-aware, decentralized training across platforms, ensuring user data remains secure.
5. Enhanced Behavioral Modeling: Deep learning techniques like transformers could further improve the system's ability to detect complex, evolving fake profiles.
6. Cross-Platform Verification**:** The system could cross-reference user profiles across multiple platforms to enhance accuracy and flag suspicious accounts more effectively.

IX. REFERENCES

[1] S. Kadwe, D. Dharmaraj, and D. Kharat, “EduDocs: Document Verification using Blockchain,” Proc. IEEE Int. Conf. Blockchain and Distributed Systems Security (ICBDS), 2024.

[2] H. Gaikwad, N. D’Souza, R. Gupta, and A. K. Tripathy, “A Blockchain-Based Verification System for Academic Certificates,” Int. Conf. Smart Computing and Communication (ICSCAN), 2021.

[3] A. Chowdhary, S. Agrawal, and B. Rudra, “Blockchain Based Framework for Student Identity and Educational Certificate Verification,” Int. Conf. Electronics, Systems and Communication (ICESC), 2021.

[4] A. Singh, S. Chauhan, and A. K. Goel, “Blockchain Based Verification of Educational and Professional Certificates,” Int. Conf. Communication Systems and Computing (ICCSC), 2023.

[5] S. Al-Qurishi, M. Al-Rakhami, and M. Al-Rakhami, “Detecting Fake Profiles in Online Social Networks Using Machine Learning,” IEEE Access, vol. 8, pp. 70214–70226, 2020.

[6] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, “Detecting and Characterizing Social Spam Campaigns,” Proc. ACM SIGCOMM IMC, pp. 35–47, 2010.

[7] N. A. Syed, S. A. Malik, and I. A. Taj, “Fake Profile Detection on Social Media Using Machine Learning Algorithms,” Int. J. Advanced Computer Science and Applications, vol. 11, no. 9, pp. 439–447, 2020.

[8] P. Choudhury and B. K. Tripathy, “Detection of Spammers in Twitter Using Machine Learning Tools,” Proc. Int. Conf. Advances in Computing, Communications and Informatics, pp. 451–457, 2015.

[9] A. H. Wang, “Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach,” Data and Applications Security and Privacy XXIV, Springer, pp. 335–342, 2010.

[10] R. Kaur and P. Kaur, “A Survey on Machine Learning Techniques for Fake Profile Detection in Social Networks,” Int. J. Computer Applications, vol. 179, no. 49, pp. 12–17, 2018.