Final Project Report

AI-Enhanced Detection of Fake and Bot Profiles on Social Media

Internship Group Project | ITSOLERA, Islamabad

Introduction

The widespread use of social media platforms has created opportunities for malicious actors to exploit systems by creating fake and bot accounts. These profiles can manipulate trends, impersonate users, or propagate disinformation at scale. This project aims to develop a robust AI-based system to identify and classify social media accounts into three categories: Real, Fake, and Bot.

Our approach includes traditional machine learning models, deep learning models, anomaly detection techniques, and a full deployment pipeline using Flask and Docker.

Project Objectives

- Develop a classification model to detect fake, bot, and real social media profiles.
- Engineer meaningful features from user metadata and behavioral patterns.
- Evaluate different models using cross-validation and performance metrics.
- Build a deployable web service that integrates the model for real-time inference.

Dataset Overview

We used the LIMFADD (Instagram Multi-Class Fake Account Detection Dataset), a multi-class dataset containing labeled instances of:

- Real accounts
- Fake accounts
- Bot accounts (Scam class was removed for this project.)

Each entry contains:

- Profile metadata (followers, following, posts, bio, username, is_private, etc.)

- Behavioral indicators (has_profile_picture, is_verified, etc.)

Data Preprocessing

- Cleaned inconsistent values such as "Yes" and "N" into lowercase and unified categories.

- Used LabelEncoder to convert object-type columns to numeric values without deleting any important columns.

- Created new features:

- followers_following_ratio

- bio_length

- username_length

- Removed rows with label "Scam".

- Checked and handled missing/null values appropriately.

- Verified all data types were suitable for model input.

Machine Learning Models

1. Logistic Regression

- Trained as a simple interpretable baseline

- Performed well in classifying real vs. non-real accounts

2. Random Forest

- Captured non-linear relationships

- Improved accuracy over logistic regression

- Feature importance was analyzed to interpret model decisions

Cross-Validation

- Applied 5-fold cross-validation for both logistic regression and random forest.
- Ensured generalizability and avoided overfitting.
- Accuracy and F1-scores were recorded for each fold and averaged.

Anomaly Detection

- Implemented Isolation Forest and One-Class SVM to detect outliers and suspicious behavior patterns.
- Anomaly scores were analyzed to further flag accounts that might not fall into any known class confidently.

Evaluation Metrics

Used the following metrics across models:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

All metrics were calculated for each class (real, fake, bot) and macro-averaged.

Visualizations

- Confusion matrix heatmaps
- Accuracy vs. epochs for neural network
- Feature distribution plots for high-impact variables
- ROC curves (where applicable)

Deep Learning Model (Neural Network)

Architecture:

- Input layer with 128 units (ReLU)
- Batch normalization and dropout
- Two hidden layers (64 and 32 units)
- Output layer (3 units with Softmax)

Training:

- Categorical Crossentropy loss
- Adam optimizer
- Early stopping and model checkpointing
- Achieved 94% test accuracy

Deployment Pipeline

1. Flask Web Application

- A REST API was created to accept profile input features
- The model predicts the class (Real, Fake, Bot) and returns the result
- predict() method accepts POST requests with JSON payload

2. Dockerization

- Dockerfile was created to containerize the Flask app
- requirements.txt included all necessary libraries (Flask, TensorFlow, pandas, etc.)
- Docker ensures easy deployment across systems

Future Enhancements

- Integrate text-based analysis using LSTM or Transformers on bio or post content.
- Extend to Twitter and Facebook datasets.

- Improve UI/UX with a front-end interface.
- Add model explanation (SHAP or LIME) for transparency.

Technologies Used

- Language: Python
- ML Libraries: scikit-learn, TensorFlow, Keras
- Visualization: Matplotlib, Seaborn
- Web Framework: Flask
- Containerization: Docker
- IDE/Environment: Google Colab, VS Code

Conclusion

The project successfully delivered a robust machine learning pipeline capable of detecting fake and bot profiles with high accuracy. The system was tested, fine-tuned, and deployed as a scalable web API using modern tools.

This solution can contribute to improving the integrity of social media ecosystems and detecting malicious activities more effectively.