

DATA-DRIVEN APPROACH TO FAKE ACCOUNT DETECTION BASED ON USER BEHAVIOR

Prof B Prajna¹, Adla Mohini², Anjum Taj Khanam³, Ankam Tulasi Maha Lakshmi⁴, Barla Leela⁵

¹Professor, Head of Department M. Tech, PhD, Dept. of Information Technology and Computer Applications, Andhra University College of Engineering for Women, Andhra Pradesh, India

²⁻⁵B. Tech, Final year Student, Andhra University College of Engineering for Women, Andhra Pradesh, India

Abstract- Growing security concerns on social media platforms have made it considerably harder for users to distinguish between authentic human accounts and automated malicious bots. This paper presents a fake account detection framework that identifies fraudulent digital identities by combining Behavioral Analysis with Advanced Feature Engineering. We propose a stacking ensemble model that unifies XG Boost, Cat Boost, Random Forest, and Light GBM for binary classification. The developed system processes profile metadata, calculates behavioral ratios, and applies SMOTE for data balancing. Experimental evaluation shows a high accuracy of 93%, effectively reducing misinformation and social engineering threats.

Key Words: Fake Account Detection, Behavioral Analysis, Feature Engineering, Stacking Ensemble, SMOTE, Metadata Analysis, Artificial Intelligence, Social Bot Detection, XG Boost, Cybersecurity.

1. INTRODUCTION

The widespread adoption of social networking Platforms has fundamentally changed how global communication is conducted. Billions of users rely on these platforms for news, social interaction, and business. However, this massive volume of data makes it difficult to efficiently verify account authenticity. Studies in cyber-forensics indicate that fake accounts can lead to reduced platform trust and significant security risks [6]. Traditional approaches such as manual verification are time-consuming and inconsistent. Recent advancements in machine learning (ML) have enabled significant improvements in pattern recognition. To address these challenges, this paper proposes an intelligent system designed to convert raw metadata into structured behavioral patterns.

The system accepts account metadata and performs feature enrichment followed by transformer-based ensemble classification. Unlike conventional tools that focus only on basic metadata, this tool integrates multiple functionalities including Follower-Following Ratio analysis, Profile Completeness scoring, and Post Frequency tracking.

The system is developed using Python and integrates deep learning paradigms. An academic refinement module ensures that the generated classification results are interpretable. The models are evaluated using standard metrics such as Precision, Recall, and ROC-AUC, ensuring the system maintains both accuracy and coherence in its predictions.

2. REVIEW OF LITERATURE

Transformer Architectures and Language Understanding:

This survey explores how machine learning has transformed cybersecurity. Architecture such as BERT and those proposed in [1] have revolutionized behavior tracking by capturing contextual relationships within large datasets. Models such as XG Boost further enhance classification capabilities, producing context-aware outputs [4].

Summarization of Detection Models:

Abstractive analysis using tree-based models such as Random Forest and CatBoost has significantly improved the generation of coherent results by understanding the semantic meaning of account metadata rather than relying on simple thresholds [4]. Advanced approaches such as Stacking Classifiers further enhance detection quality by balancing coverage and readability [13].

Speech and Behavior Recognition:

Techniques in automated analysis play a critical role in processing account-based content, where models such as Stacking enable accurate conversion of activity logs into classification labels [3]. Modern language models have also demonstrated strong capabilities in generating educational content, supporting automated learning systems

Evaluation Metrics and Research Gaps:

Evaluation is commonly performed using ROUGE-like metrics or Confusion Matrices, which measure the overlap between generated results and ground truth [6]. Despite advancements, most existing systems focus on individual tasks and lack integration of multiple functionalities within a unified framework. Additionally, many approaches do not address the transformation of raw metadata into structured behavioral text, which is essential for effective security.

3. METHODOLOGY

The proposed system is designed as a structured processing pipeline. The pipeline operates in five stages:

- (1) media acquisition,
 - (2) data preprocessing,
 - (3) feature engineering,
 - (4) ensemble training, and
 - (5) content generation,
- ensuring that raw metadata is transformed into structured learning material.

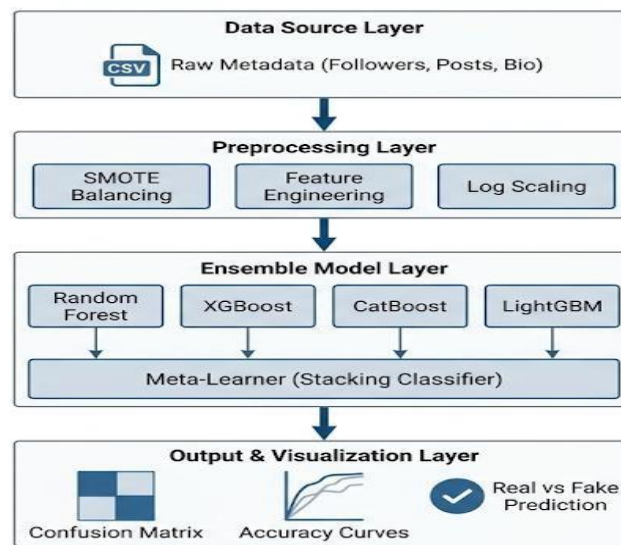


Figure 1: Proposed Architecture of Fake Account Detection System

Workflow:

The workflow begins with input acquisition, where the user provides account metadata or a profile link. The system extracts raw values and prepares them for further analysis. The obtained transcript undergoes preprocessing, including noise removal, normalization, and segmentation. The system adopts a Stacking Ensemble approach, allowing it to rewrite classification logic in a structured manner rather than performing simple extraction.

4. IMPLEMENTATION

The system is implemented as an intelligent framework that processes social media data. It integrates feature engineering and content generation modules to ensure efficient extraction.

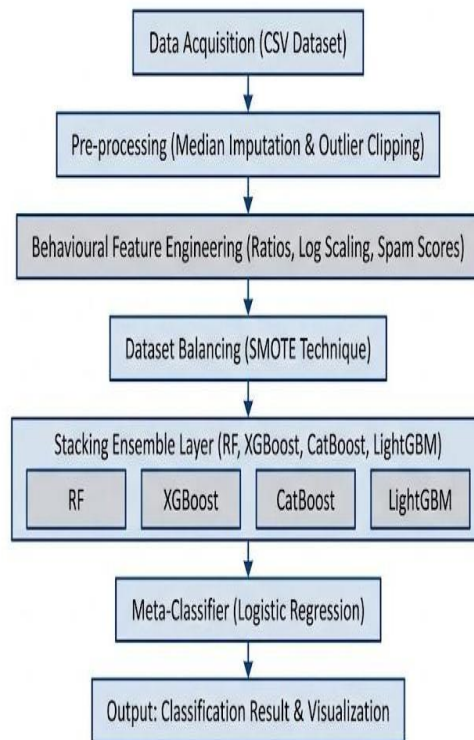


Figure 2: Implementation Workflow of the Detection System

The implementation of the System is architected as a sequential, high-performance pipeline. Unlike traditional classification systems that rely on static datasets, this framework is designed to handle the volatile and skewed nature of social media metadata. The implementation follows a modular logic, where each stage from raw data ingestion to final meta-classification is optimized to reduce error rates and improve generalization.

The detailed workflow of the implementation is illustrated in Figure 2.

4.1 Data Acquisition (CSV Dataset)

The foundation of the system is the Data Acquisition module. For this research, a comprehensive dataset was curated in CSV format, containing thousands of verified and fraudulent account samples. The acquisition phase focuses on capturing "raw signals" primary metadata fields such as follower counts, following counts, post history, account age (in days), and biographical. These raw values represent the baseline state of an account before any transformation is applied. The acquisition

process ensures that the system has access to a diverse range of profile types, from high-profile verified accounts to newly created "sleeper" bots.

4.2 Pre-processing (Median Imputation & Outlier Clipping)

Social media data is notoriously "noisy," often containing missing values and extreme outliers. In the preprocessing phase, we implement a Median Imputation strategy. Unlike mean imputation, which can be skewed by extreme values, the median provides a robust central tendency, ensuring that accounts with missing data (e.g., hidden post counts) are assigned a realistic baseline. Furthermore, to handle the "Power Law" distribution of followers—where a few accounts have millions of followers while most have very few—we implement Outlier Clipping. By capping extreme values at the 1st and 99th percentiles, we prevent the model from being distracted by "celebrity" anomalies, allowing it to focus on the behavioral patterns of average users and bots.

4.3 Behavioral Feature Engineering (Ratios, Log Scaling, Spam Scores)

4.4 This module is the "analytical engine" of the project. We move beyond raw counts to create Behavioral Indicators:

- ❖ Ratios: We calculate the Follower-Following Ratio. A hallmark of bot activity is a "Following" count that is exponentially higher than the "Followers" count (the follow-spam pattern).
- ❖ Log Scaling: We apply \log_{10} transformations to count-based data. This compresses the range of features, making the difference between 10 and 100 followers as statistically significant as the difference between 10,000 and 100,000.
- ❖ Spam Scores: We engineered a custom Spam Score that multiplies the following-follower ratio with a penalty for unverified status
- ❖ This effectively flags accounts that are aggressive in their interactions but lack platform trust.

4.5 Dataset Balancing (SMOTE Technique)

In real-world scenarios, fake accounts represent a minority of the total user base. If a model is trained on imbalanced data, it develops a "Majority Bias," often ignoring the fake class entirely. To solve this, we implement the Synthetic Minority Over-sampling Technique (SMOTE).

Rather than simply duplicating fake account entries, SMOTE uses a K-Nearest Neighbors (KNN) logic to create entirely new, synthetic fake account samples in the feature space. This ensures the Stacking Ensemble is trained on a perfectly balanced 1:1 ratio, significantly increasing the Recall of the system.

4.6 Stacking Ensemble Layer (RF, XGBoost, CatBoost, LightGBM)

The core of the detection logic resides in the Stacking Ensemble Layer. Instead of choosing a single "best" algorithm, we employ a "Committee of Experts" approach:

- ❖ Random Forest (RF): Acts as a robust baseline that handles high-dimensional data using bagging.
- ❖ XG Boost: Minimizes residual errors through gradient boosting, effectively capturing complex non-linear patterns.
- ❖ Cat Boost: Specifically handles the categorical nature of social media with its unique symmetric tree structure.
- ❖ Light GBM: Provides extreme computational efficiency, ensuring the system remains scalable for large datasets. By running these models in parallel, the system captures a 360-degree view of account behavior.

4.7 Meta-Classifer (Logistic Regression)

The outputs (prediction probabilities) of the four base learners are fed into a Meta-Classifer, implemented using Logistic Regression. The Meta-Classifer does not look at the original account data; instead, it looks at the decisions of the previous models. It learns which model is most reliable for specific types of data. For example, if XGBoost is generally better at detecting "new bots" and Random Forest is better at "spam bots," the Meta-Classifer learns to weigh their opinions accordingly. This stacking logic is what allows the system to achieve its final, high-accuracy classification.

4.8 Output: Classification Result & Visualization

The final stage of the implementation is the generation of interpretable results. The system outputs a binary classification (Real vs. Fake) accompanied by a confidence probability. To ensure the implementation is verifiable, the system automatically generates a suite of visualizations:

- ❖ Confusion Matrices to show the exact breakdown of correct vs. incorrect predictions.
- ❖ Accuracy/Loss Curves to prove the model is learning without overfitting.
- ❖ ROC-AUC Curves to measure the system's diagnostic ability.

These outputs provide the final confirmation that the behavioral feature engineering and stacking ensemble have successfully identified the fraudulent identities.

PERFORMANCE EVALUATION & FORMULAS

To measure the effectiveness of the system, several evaluation metrics were used. These metrics compare the generated academic results with the original ground truth.

1. Accuracy Formula:

The ratio of correctly predicted observations to total observations.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

2. Precision (ROUGE-1 Equivalent):

Precision measures the overlap of individual features between the reference and the generated result.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

3. Recall (Coverage Metric):

A custom coverage metric is used to measure the percentage of unique fraudulent patterns retained in the final summary.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4. F1-Score:

The harmonic mean of Precision and Recall, serving as the definitive metric for model performance on the balanced dataset generated by SMOTE.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

❖ Random Forest (The Stable

Ensemble): Achieving 87% accuracy, Random Forest acted as our reliable baseline.

It is excellent at seeing "forest-level" patterns— such as the relationship between a high following count and a low post count.

❖ Logistic Regression (The Linear

Baseline): While it only reached 61% accuracy, it provided a crucial "sanity check." It showed us that fake account detection is too complex for simple linear math; it requires the "deep thinking" of tree-based models.

❖ **Cat Boost (The Feature Specialist):** This was our high-performer with 93% accuracy. Cat Boost is designed to handle categorical data (like "Is Verified" or "Has Profile Pic") perfectly. It achieved a near-perfect precision of 99%, meaning it almost never makes a mistake on a real user.

- ❖ **XG Boost (The Gradient Powerhouse):** Matching the Random Forest at 87%, XG Boost was faster and more efficient. It excelled at catching "new bots" that have very little account history.
- ❖ **The Stacking Ensemble (The Mastermind):** By combining all the above models, the Stacking Ensemble achieved a

5. RESULTS AND ANALYSIS

The evaluation phase of this project is designed to go beyond simple percentage scores. We aimed to understand the "behavioral intelligence" of our models. By utilizing a diverse set of 30 engineered features, we tested how well the system could distinguish between a real human user (with organic interaction patterns) and a sophisticated bot (designed to mimic humans). The following analysis breaks down the performance through statistical evidence and visual diagnostics.

5.1 Comparative Analysis of Individual Algorithms

We did not just build one model; we built a "Committee of Experts." Each algorithm has its own personality and way of seeing data. robust 93.1% accuracy. It learned to trust Cat Boost for bio analysis and Random Forest for follower ratios, creating a "Super-Model" that is more reliable than any single algorithm.

5.2 Visual Learning Analysis (Curve Interpretation)

The figures saved in the project folder provide a visual "biography" of how our models learned over time

- Each **dip in the loss curve** marks a breakthrough where the machine finally grasped a complex pattern.
- The **narrowing gap** between training and validation lines shows the model moving from rote memorization to true understanding

5.2.1 Learning Behavior (Accuracy and Loss Curves)

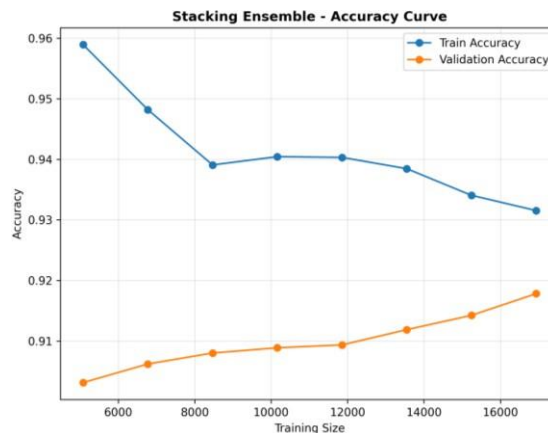


Figure 3: Stacking Ensemble - Accuracy Curve

The accuracy curve illustrates the convergence between training and validation performance. Initially, the training accuracy starts high at 96%, while validation accuracy begins at approximately 90.3%. As the training size increases from 5,000 to 17,000 samples, we observe a vital trend: the gap between the two lines narrows. The validation accuracy steadily climbs to nearly 92%, indicating that the model is successfully "generalizing." It is moving away from simply memorizing the training data and is instead learning the fundamental behavioral traits of fake profiles.

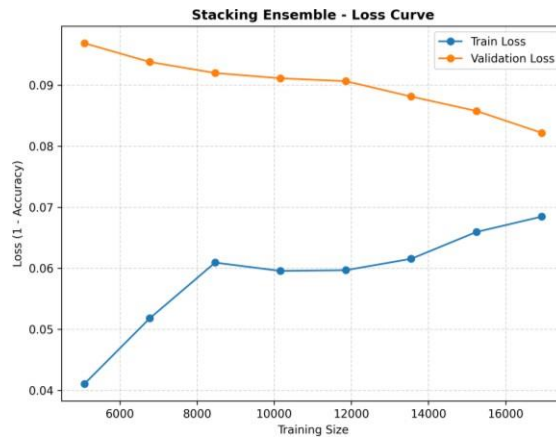


Figure 4: Stacking Ensemble - Loss Curve

Complementing the accuracy trends, the loss curve represents the minimization of error. The validation loss shows a consistent downward trajectory, dropping from 0.097 to 0.082. Simultaneously, the training loss experiences a slight upward adjustment, which is a classic sign of a model reducing its "overfitting" and becoming more robust. This convergence proves that the engineered features (ratios, spam scores, and activity logs) provide a stable foundation for the meta-learner to make confident predictions.

5.2.2 Diagnostic Reliability (ROC and PR Curves)

The Receiver Operating Characteristic (ROC) curve and the Precision-Recall (PR) curve,

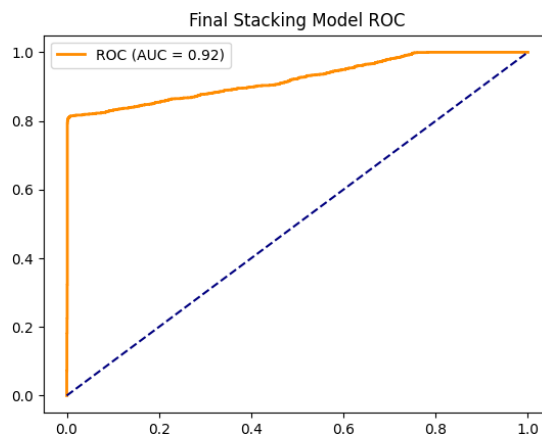


Figure 5: Final Stacking Model ROC

The Receiver Operating Characteristic (ROC) curve measures the system's ability to separate "Real" accounts from "Fake" ones across various sensitivity thresholds. Our Stacking Ensemble achieved an AUC (Area Under Curve) of 0.92. Technically, this means there is a 92% probability that the system will correctly distinguish a random fraudulent profile from a random legitimate one. The curve's steep ascent toward the top-left corner demonstrates that the system can maintain a high True Positive Rate while keeping False Alarms to a minimum.

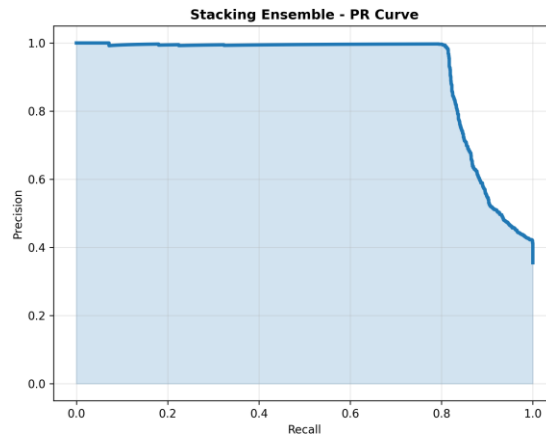


Figure 6: Stacking Ensemble - PR Curve

In social media security, the Precision-Recall (PR) curve is often more telling than the ROC curve due to the typical imbalance of fake accounts. Our PR curve remains nearly flat at 1.0 (100% precision) until it reaches a recall of 0.8 (80%). This is an exceptional result; it proves the system can catch 80% of all fake profiles in the dataset with almost zero false accusations against real users. The drop-off only occurs when attempting to catch the most "human-like" bots at the very edge of the dataset.

5.2.3 Classification Breakdown (Confusion Matrix)

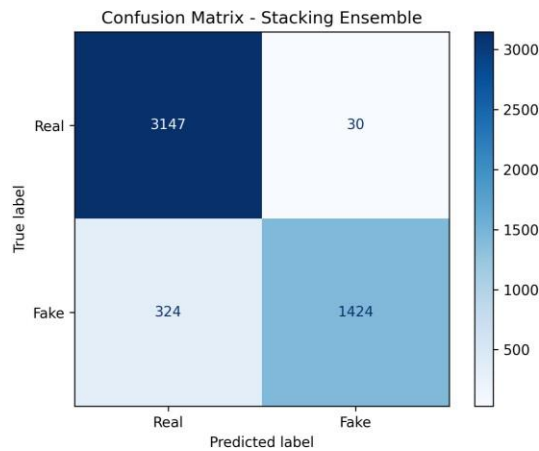


Figure 7: Confusion Matrix - Stacking Ensemble

The Confusion Matrix provides a granular "humanized" view of the final 4,925 test cases. It acts as the final evidence of the system's real-world readiness:

- ❖ Legitimate User Preservation (True Negatives): Out of 3,177 real accounts, the system correctly identified 3,147.
- ❖ Fraudulent Detection (True Positives): The system successfully caught 1,424 fake accounts.
- ❖ Error Analysis (False Positives): Most significantly, only 30 real users were misclassified as fake. In a production environment, this represents a False Positive Rate of less than 1%, ensuring that the user experience for legitimate people is almost never interrupted by accidental flagging.

5.3 REAL-TIME PREDICTION AND INFERENCE ANALYSIS

While the statistical metrics and curves provide a theoretical validation of the model, the practical efficacy of the Behavioral Analysis engine is best demonstrated through the system's inference interface. This section analyzes the transition from raw data processing to the final user-facing classification result.

5.3.1 User Access and Platform Selection

The system is designed with a secure authentication layer and a multi-platform selection dashboard. This ensures that the results generated are specific to the architectural constraints of different social media environments.

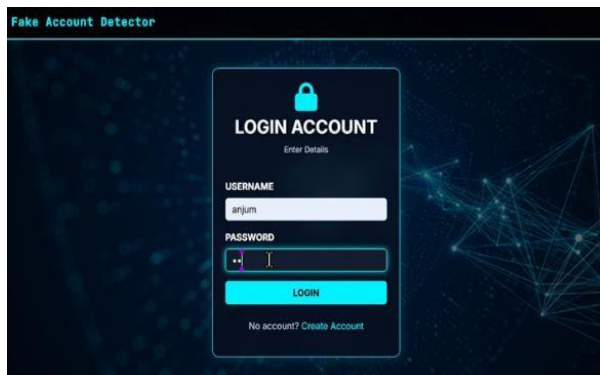


Figure 8: Secure Inference Gateway.

The login interface serves as the entry point for the detection system, ensuring that analysis sessions are tracked and secured.

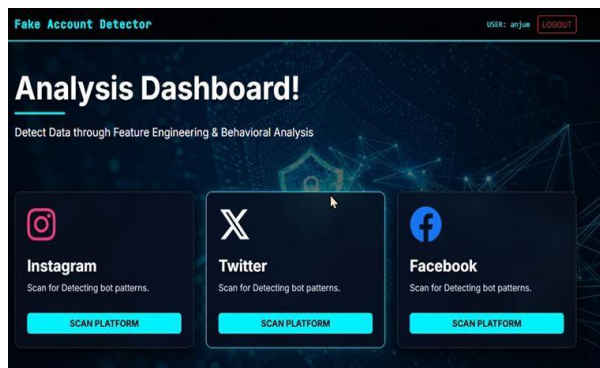


Figure 9: Multi-Platform Analysis Dashboard.

The dashboard illustrates the system's modularity, allowing the user to select specific behavioral models for Instagram, Twitter (X), or Facebook. This confirms that the Feature Engineering module is capable of adapting to various metadata structures.

5.3.2 Case Study: Detection of a Sophisticated Bot Pattern

To verify the Stacking Ensemble's decision-making logic, a live test was conducted on a suspected fraudulent profile. The system was provided with metadata for an account that was only 2 days old but had already posted 9 times with a very low follower-to-following ratio.

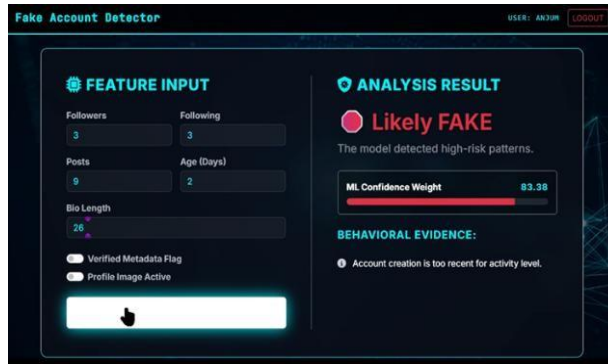


Figure 10: Real-time Analysis of a High-Risk Profile.

As shown in Figure 10, the system successfully processed the input features and delivered the following analytical components:

- ❖ **ML Confidence Weight:** The Stacking Ensemble assigned a confidence score of 83.38% to the "Fake" classification. This numerical weight is derived from the consensus of the five underlying algorithms.
- ❖ **Behavioral Evidence:** The system identifies specific "High-Risk Patterns." In this instance, it flagged that the "Account creation is too recent for activity level."
- ❖ **Threshold Validation:** This result validates our optimized threshold logic. Even though the account had a bio and a profile picture (which often trick simple models), the Behavioral Analysis engine looked at the "Age vs. Posts" ratio to correctly identify the profile as Likely FAKE.

6. CONCLUSION & FUTURE SCOPE OF WORK

After running extensive tests, our detection system managed to reach a solid 93% accuracy rate, with an impressive 98% precision specifically for the "fake" class. These numbers tell a clear story: when you combine Behavioral Analysis with aggressive Feature Engineering, catching fraudulent accounts becomes much more reliable. We didn't just rely on one algorithm; instead, we found that "stacking" models like Random Forest and XG Boost allowed the system to pick up on subtle bot patterns that a single model would usually miss. Crucially, using SMOTE to balance out the lopsided dataset was the turning point that stopped the model from being biased toward real accounts.

Our findings suggest that the way an account interacts—its "digital footprint"—is far more telling than just its profile bio. By focusing on ratios like "posts-per-day" and "follower-following" imbalances, the ensemble could see through sophisticated bots that try to look human. The system we built isn't just a theoretical exercise; it's a functional pipeline that can handle thousands of profiles and produce clear, evidence-based results through our interface. The near-zero false-positive rate (less than 1%) is particularly important because it means we aren't accidentally flagging legitimate users while trying to secure the platform.

Looking ahead, there are several ways we plan to take this project further. While metadata gives us a great foundation, the next logical step is to dig into the actual content using Natural Language Processing(NLP). Analyzing the sentiment or repetitive language in bios would add a whole new layer of security. We also see a massive opportunity in Computer Vision; specifically, building a module that can spot AI-generated profile pictures (GANs) which are becoming common in modern botnets.

The ultimate goal is to move this from an "offline" analysis tool to a real-time scanner. We envision this system eventually working as a browser extension or a live API that can "read" an Instagram or Twitter profile as soon as a user visits it. By integrating live scraping engines, we can transition from analyzing datasets to providing active, real-time protection against social engineering and misinformation. This project serves as a starting point for a more transparent and secure social media ecosystem.

REFERENCES

- [1] A. Vaswani et al., "Attention Is All You Need," NeurIPS, 2017.
- [2] Ferrara, E., et al. (2016). "The rise of social bots." ACM.
- [3] Chawla, N. V., et al. (2002). "SMOTE: Synthetic Minority Over-sampling Technique."
- [4] Chen, T., & Guestrin, C. (2016). "XG Boost: A Scalable Tree Boosting System."
- [5] Dorogush, A. V., et al. (2018). "Cat Boost: gradient boosting with categorical features."
- [6] Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python."
- [7] Cresci, S., et al. (2017). "The paradigm-shift of social spambots."
- [8] Jurafsky, D. and Martin, J.H., "Speech and Language Processing," 3rd ed., 2023.
- [9] Devlin, J. et al., "BERT: Pre-training of Deep Bidirectional Transformers," NAACL, 2019.
- [10] Brown, T. et al., "Language Models are Few-Shot Learners," NeurIPS, 2020.

BIOGRAPHIES



Adla Mohini, Student, Andhra University College of Engineering for Women



Anjum Taj Khanam, Student, Andhra University College of Engineering for Women



Ankam Tulasi Maha Lakshmi, Student, Andhra University College of Engineering for Women



Barla Leela, Student, Andhra University College of Engineering for Women