IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Authenticity Detection in Instagram Accounts Using Machine Learning

**MUNUKOTI SRI TANAY[1], GOPAGANI TEJA UDAY KIRAN[2], RAMASAHAYAM NITHIN REDDY[3], SANAKA DUNDI SOMAN[4], MOHAMMED ABDUL MUNEER[5]**

[1,2,3,4]Department of Information Technology, J.B. Institute of Engineering & Technology, Hyderabad

[5]Assistant Professor, Department of Information Technology, J.B. Institute of Engineering & Technology, Hyderabad

***Abstract*—**Social media platforms such as Instagram have experienced rapid growth over the past decade. However, this expansion has also introduced significant risks, including fake profiles, impersonation attacks, spam conversations, and phishing attempts. Malicious actors frequently exploit deceptive profile information and social engineering methods, making it genuinely difficult for ordinary users to spot fraudulent accounts. Existing detection solutions are mostly controlled by the platforms themselves and operate as closed, opaque systems that offer very little transparency or direct control to individual users.

This paper presents a machine learning–based system designed to determine the authenticity of Instagram accounts by examining both profile characteristics and message behaviour. The proposed system is built on a full-stack architecture that combines a React– based user interface, a Node.js backend, a SQLite database, and Python–based machine learning models. For profile analysis, four classification algorithms are employed: Random Forest, Gradi- ent Boosting, Logistic Regression, and XGBoost. Message–level analysis uses TF–IDF feature extraction paired with Multinomial Naive Bayes and Logistic Regression classifiers to detect scam and spam messages. The system merges machine learning predictions with rule–based heuristic analysis to compute a comprehensive risk score and deliver interpretable results to end users.

Experimental results confirm strong detection performance, reaching up to 94% accuracy for profile analysis and exceeding 97% accuracy for message classification. The proposed frame- work offers a practical, user–centric tool for strengthening social media security through real–time detection, transparent analysis, and persistent tracking of suspicious activity.

**Keywords:** Fake account detection, Instagram authenticity, machine learning, TF–IDF, Random Forest, XGBoost, Naive Bayes, heuristic analysis, social media security, risk scoring.

## I. Introduction

Social media has fundamentally reshaped how people communicate, share information, and conduct day–to–day activi- ties. Among all platforms, Instagram stands out for its visual appeal, ease of use, and a global user base that crossed two billion active accounts in recent years. While this reach makes Instagram an invaluable tool for personal connection, business promotion, and public discourse, it has simultaneously created fertile ground for a wide range of malicious activities [22].

Fraudulent actors on Instagram typically operate through carefully crafted fake profiles that mimic the appearance of genuine accounts. These profiles are used to initiate spam conversations, conduct phishing campaigns, spread misinfor- mation, and in many cases, carry out financial fraud. Profile metadata is manipulated to appear credible, and the messages sent often use persuasive language and urgency–based tactics to exploit the trust of unsuspecting users [24]. The problem is further compounded by the emergence of sophisticated social bots that can imitate human behaviour closely, making them significantly harder to detect [14].

Platform–level defences, while useful, have notable limitations. They function as black–box mechanisms that operate without user visibility into the decision–making process. Individual users have no direct access to the reasoning behind account suspensions or warning flags, and they have no independent means to verify the authenticity of accounts they encounter [18]. This creates a gap that leaves everyday users vulnerable, particularly when interacting with accounts that have not yet been flagged by the platform.

The need for a user–controlled, transparent, and practical detection system is therefore evident. Machine learning has demonstrated strong potential in tackling this problem through its ability to learn patterns from data and generalise to un- seen examples. Algorithms such as Random Forest [25] and XGBoost [19] have proven highly effective at structured clas- sification tasks, while natural language processing techniques enable meaningful analysis of textual message content [21].

This paper presents a system that directly addresses this gap. The proposed framework analyses Instagram profiles and messages using a combination of supervised machine learn- ing models and heuristic behavioural indicators. The system delivers a risk score along with an explainable breakdown of the factors contributing to that score, enabling users to make informed decisions before engaging with an unknown account [15], [17].

The remainder of this paper is organised as follows. Sec- tion II reviews relevant prior work on fake account detection and social media security. Section III describes the limita- tions of existing systems. Section IV details the proposed system and its methodology. Section V explains the full–stack implementation. Section VI presents the experimental results and discussion. Section VII concludes the paper and outlines directions for future work.

## II. Literature Survey

The challenge of identifying fake profiles and malicious behaviour on social media has attracted considerable research attention over the past decade. Studies have approached this

problem from multiple angles, including behavioural pattern analysis, machine learning classification, natural language processing, and graph–based modelling.

Early investigations into social media security concentrated on detecting automated accounts through activity–level signals. Chu et al. proposed one of the first methods for identifying automated Twitter accounts by examining posting frequency, temporal patterns, and interaction behaviour [22]. Their work established that automated bots tend to exhibit statistically distinct activity patterns when compared to genuine human users. In a related effort, Stringhini et al. analysed profile features and message patterns on multiple social networks to detect spam accounts, demonstrating that a combination of structural and textual signals can achieve reliable spammer identification [24].

As detection methods matured, so did the sophistication of malicious bots. Cresci et al. documented the evolution of social spambots, showing that modern bots have learned to mimic human–like behaviour in terms of posting timing and interaction style, significantly raising the bar for detection systems [14]. A later study by Cresci further charted a decade of progress in social bot detection and identified persistent challenges that remain unsolved [5].

Machine learning has been central to improving detection accuracy. Breiman's Random Forest algorithm has become a standard tool for binary and multiclass classification due to its ability to handle high–dimensional feature spaces and reduce overfitting through ensemble aggregation [25]. Chen and Guestrin proposed XGBoost, a gradient boosted tree algorithm notable for its speed, scalability, and strong performance on tabular data, which has since been adopted widely in anomaly detection and fraud detection tasks [19]. The Scikit-learn library has enabled practical implementation of these and other algorithms in research and production environments [23].

Deep learning has opened additional avenues for detection. Goodfellow et al. provided the foundational framework for applying deep neural networks to complex classification tasks, including those involving large and heterogeneous feature sets [20]. Natural language processing models have extended this capability to text analysis. Mikolov et al. proposed Word2Vec, demonstrating that word embeddings can capture semantic relationships and improve understanding of textual patterns in social media messages [21].

More recent work has introduced transformer–based language models that further advance text understanding. Devlin et al. presented BERT, which uses bidirectional contextual representation to capture nuanced language understanding, proving effective for detecting deceptive communication [8]. Liu et al. extended this with RoBERTa, which optimises the pre–training procedure of BERT and achieves superior results on multiple natural language understanding benchmarks [9]. Cer et al. proposed the Universal Sentence Encoder, which generates fixed–length semantic embeddings of sentences for use in downstream classification tasks [11].

Graph–based approaches have also contributed to social bot detection. Kipf and Welling introduced Graph Convolutional Networks, which model relationships between nodes in a graph and are naturally suited to social network analysis where user interactions define the graph structure [16]. Wu et al. provided a comprehensive survey of Graph Neural Networks and their applications, highlighting their potential for detecting anomalies in large social graphs [4].

Interpretability of machine learning predictions has received increasing emphasis, particularly in security–sensitive applications. Ribeiro et al. proposed LIME, a framework that explains individual predictions by identifying the most influential input features [17]. Lundberg and Lee developed SHAP, which provides theoretically grounded feature attributions and allows consistent comparison of feature importance across predictions [15]. Molnar's work on interpretable machine learning has further established that transparency is not merely a design preference but a functional requirement in high–stakes applications [3].

Research on misinformation has underscored the broader consequences of fraudulent activity on social networks. Shu et al. examined fake news detection and argued that combining content features with user behaviour signals produces more robust detection systems [12]. Zhou and Zafarani surveyed methods for fake news identification and highlighted the complementary roles of machine learning and network analysis [7]. Ferrara et al. studied the influence of social bots in spreading disinformation and found that bots can significantly amplify false narratives before human fact–checkers can respond [18].

The dynamics of information spread were analysed by Vosoughi et al., who found that false information propagates faster and more broadly than accurate information, largely due to the novelty effect and bot amplification [10]. Varol et al. complemented this by studying human–bot interaction patterns and proposing techniques for estimating bot prevalence within social media communities [13].

Recent integrative studies have combined multiple signals for improved detection. Pierri et al. surveyed machine learning and social network analysis approaches for misinformation and fake news detection, noting that single–modality methods consistently underperform compared to hybrid approaches [6]. Ferrara discussed the coordinated use of bots in disinformation campaigns and the challenges this poses for platform–level countermeasures [2]. Jin et al. provided a detailed survey of deep learning methods for social bot detection, emphasising that integrating profile features, textual content, and temporal behavioural signals yields the most reliable results [1].

While the existing body of literature offers valuable techniques, most proposed systems are designed for platform–level deployment and require extensive data access privileges. There is a clear gap for practical, user–centric tools that operate with publicly observable profile and message data. The system proposed in this paper directly addresses that gap by combining supervised machine learning, TF–IDF text analysis, and heuristic behavioural scoring into a single, accessible framework.

## III. Existing System

### A. Overview of Current Approaches

The detection of fake Instagram accounts and malicious social media behaviour is currently handled predominantly through platform–controlled internal systems and a small number of academic research prototypes. Understanding the capabilities and limitations of these existing solutions is essential to motivating the design choices of the system proposed in this paper.

### B. Platform-Level Detection Mechanisms

Instagram and its parent company, Meta, employ proprietary machine learning pipelines that operate on a massive scale. These systems continuously monitor account creation patterns, login behaviour, posting frequency, device fingerprints, and interaction graphs to identify accounts that deviate from norms associated with genuine users [18]. When anomalies are detected, accounts may be automatically suspended, subjected to CAPTCHA challenges, or flagged for human review. Meta also uses coordinated detection across its family of apps, sharing signals between Facebook, Instagram, and WhatsApp to identify cross–platform abuse campaigns [2].

Despite their scale, platform–level systems have well–documented shortcomings. They are entirely opaque to individual users: no explanations are offered for account suspensions or warning flags, and users have no means to query the reasoning behind a moderation decision [3]. False positive rates, while unpublished, are acknowledged to cause significant disruption for legitimate users, particularly creators and small businesses. Furthermore, these systems are reactive by design and can take considerable time to respond to newly created fraudulent accounts, leaving a window of vulnerability during which unsuspecting users may be deceived [5].

### C. Third-Party and Research Prototypes

Several research groups have proposed standalone tools for fake account detection. Bot detection services such as Botometer (formerly BotOrNot), developed for Twitter, assign a bot–likelihood score to individual accounts based on more than one thousand features spanning network structure, content, and temporal behaviour [13]. While effective in the Twitter domain, such tools are not directly applicable to Instagram due to differences in API access, profile structure, and interaction modalities.

Academic systems targeting Instagram specifically have explored approaches ranging from simple threshold–based rules applied to follower ratios and post frequencies, to more advanced supervised classifiers trained on manually labelled datasets [14], [24]. Graph–based methods that model the follower network as a social graph and apply community detection or Graph Neural Networks to identify coordinated inauthentic clusters have also been proposed [4], [16]. These approaches, however, require access to large volumes of network–level data that are not available to ordinary users through public APIs.

Spam and phishing message detection tools, whether integrated into email clients or operating as browser extensions, have demonstrated the effectiveness of TF–IDF and Naive Bayes classifiers for textual classification [24]. More recent systems have incorporated deep learning models such as BERT and its variants to capture contextual language patterns in deceptive messages [8], [9]. While these tools achieve high accuracy, they are typically tailored to a single modality (either profile analysis or message analysis) and do not provide a unified risk assessment that combines both signals.

### D. Key Limitations of Existing Systems

The existing systems can be summarised as exhibiting the following limitations, which the proposed system is designed to address:

- **Opacity:** Results are presented without explanations, leaving users unable to understand why an account was flagged or cleared [15], [17].
- **Inaccessibility:** Most systems either require platform–level data access or operate only within proprietary infrastructure, placing them beyond the reach of individual users.
- **Single–modality focus:** Existing tools typically analyse either profile features or message content, but not both in a unified pipeline [1].
- **No persistent tracking:** Detection events are not stored in a user–accessible history, preventing longitudinal analysis of suspicious accounts.
- **Reactive posture:** Existing solutions flag accounts only after sufficient evidence accumulates, leaving early–stage fraudulent accounts undetected [2], [5].

These gaps motivate the development of a user–centric, transparent, multi–modal detection framework, as described in the following section.

## IV. Proposed System

### A. System Overview

The proposed system is a full–stack web application designed to detect fake or suspicious Instagram accounts by jointly analysing profile structure and message content. Unlike existing platform–level solutions, it is accessible directly to individual users, operates on publicly observable data, and returns fully explainable results accompanied by a quantified risk score. The architecture is composed of four tightly integrated layers: a React–based user interface, a Node.js backend server, a Python machine learning engine, and a SQLite persistence layer.

Figure 1 illustrates the overall system architecture.

### B. Design Principles

The system is guided by three core design principles.

**Transparency:** Every risk score is accompanied by a plain–language explanation identifying the features that contributed most to the result, aligning with established principles of interpretable machine learning [3], [15], [17].

**Multi–modality:** Both profile–level structural features and message–level textual features are analysed, and their outputs are fused into a single composite score. Research consistently

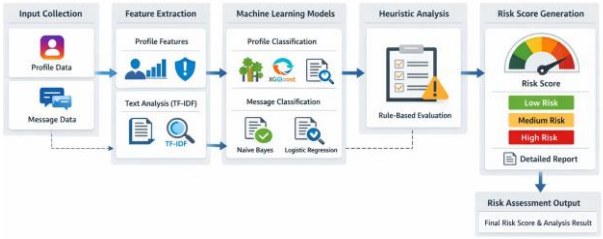


Fig. 1. Overall system architecture of the proposed Instagram authenticity detection framework, showing the data flow from input collection through machine learning models to risk score generation.

shows that combining multiple signal types improves detection robustness [1], [6].

**Persistence:** All analyses are stored in a user–accessible history, enabling longitudinal tracking of suspicious accounts and supporting informed decision–making over time.

### *C. Methodology*

The proposed methodology follows a structured sequence of stages: data collection, preprocessing, feature extraction, model training, heuristic analysis, and risk scoring. Figure 2 illustrates the training and prediction workflow.

*1) Data Collection:* The first stage gathers datasets cover- ing Instagram profiles and message interactions. Each profile record includes attributes such as follower count, following count, number of posts, account age, username character pat- terns, and bio text. The message dataset contains labelled tex- tual conversations representing normal communication, spam, phishing, and promotional scam messages. The datasets in- clude both genuine and fraudulent examples to support super- vised model training. All collected data is stored in structured form before being passed to the preprocessing stage.

*2) Data Preprocessing:* Raw data undergoes cleaning to ensure quality and consistency. Missing values are imputed or removed, duplicate records are eliminated, and attributes that provide no discriminative value are dropped. Textual fields, including profile bios and message content, are normalised by removing stop words, punctuation, and irrelevant symbols, and converting all text to lowercase. Numerical features such as follower counts and post counts are normalised to a common scale using min–max normalisation, defined as:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

where $x$ is the original feature value, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of that feature in the dataset, and $x'$ is the normalised value in the range $[0, 1]$.

*3) Feature Extraction:* Two distinct feature sets are ex- tracted: structural profile features and textual message features.

**Profile Features:** The following indicators are computed for each profile.

- Follower–to–following ratio: $R = \frac{F_{in}}{F_{out}+1}$, where $F_{in}$ is the follower count and $F_{out}$ is the following count. A

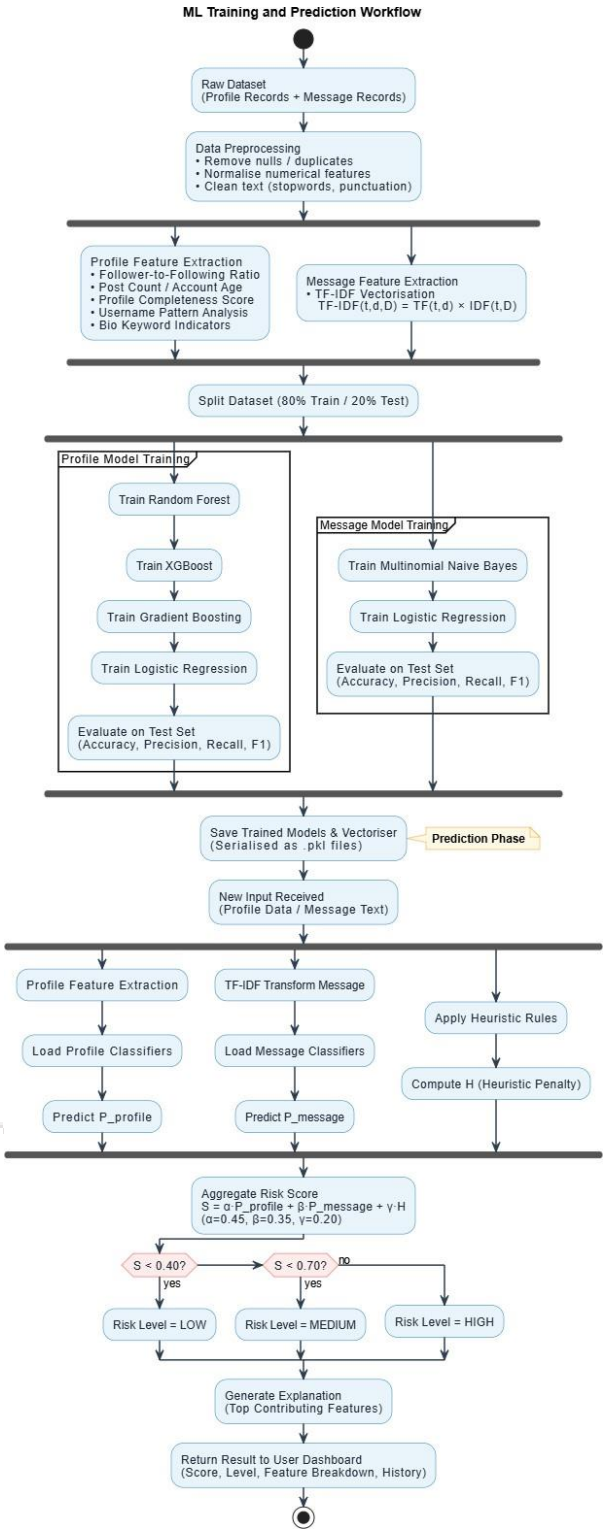


Fig. 2. Machine learning training and prediction workflow, illustrating data splitting, model training for profile and message classification, and the generation of the final risk score.

very low or very high ratio is associated with suspicious accounts.
- Number of posts relative to account age.
- Profile completeness score based on presence of a bio, profile picture, and external link.
- Presence of suspicious keyword patterns in the bio.
- Irregular character sequences in the username such as excessive digits or special characters.

**Message Features using TF–IDF:** Textual messages are converted to numerical feature vectors using Term Frequency–Inverse Document Frequency (TF–IDF) vectorisation. For a term $t$ in document $d$ within a corpus $D$:

$$\mathrm{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2}$$

$$\mathrm{IDF}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{3}$$

$$\text{TF-IDF}(t, d, D) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t, D) \tag{4}$$

where $f_{t,d}$ is the raw count of term $t$ in document $d$, $|D|$ is the total number of documents, and the denominator of the IDF term counts the number of documents containing $t$. This transformation assigns higher weights to terms that are distinctive within a document but rare across the corpus, effectively emphasising vocabulary characteristic of scam messages.

*4) Machine Learning Model Training:* The labelled dataset is split into training and test sets using an 80:20 ratio. Separate models are trained for profile analysis and message classification.

**Random Forest for Profile Classification:** A Random Forest classifier constructs $T$ decision trees, each trained on a bootstrap sample of the training set. The final prediction is determined by majority voting:

$$\hat{y} = \arg\max_{c} \sum_{t=1}^{T} \mathbf{1}\, h_t(\mathbf{x}) = c \tag{5}$$

where $h_t(\mathbf{x})$ is the prediction of the $t$–th tree for input $\mathbf{x}$, $c$ denotes the class label, and $\mathbf{1}[\cdot]$ is the indicator function.

**XGBoost for Profile Classification:** XGBoost minimises a regularised objective function by iteratively adding trees that correct residual errors [19]:

$$\mathcal{L}^{(m)} = \sum_{i=1} l\left(y_i, \hat{y}_i^{(m-1)} + f_m(\mathbf{x}_i)\right) + \Omega(f_m) \tag{6}$$

where $l(\cdot)$ is a differentiable loss function, $f_m(\mathbf{x}_i)$ is the prediction of the $m$–th tree, and $\Omega(f_m) = \gamma T + \frac{1}{2}\lambda \|\mathbf{w}\|^2$ is the regularisation term penalising tree complexity (number of leaves $T$ and leaf weights $\mathbf{w}$).

**Multinomial Naive Bayes for Message Classification:** For a message represented by feature vector $\mathbf{x}$, the Multinomial Naive Bayes classifier assigns the class $c^*$ that maximises the posterior probability:

$$c^* = \arg\max_{c \in C} P(c) \prod_{i=1} P(x_i \mid c) \tag{7}$$

where $P(c)$ is the prior probability of class $c$, $x_i$ is the $i$–th feature value, and $P(x_i \mid c)$ is the likelihood of observing feature value $x_i$ given class $c$, estimated from training data with Laplace smoothing to handle unseen terms.

**Logistic Regression for Both Tasks:** Logistic Regression models the probability of the positive class as:

$$P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x} + b)}} \tag{8}$$

where $\mathbf{w}$ is the weight vector, $b$ is the bias term, and $\sigma(\cdot)$ is the sigmoid function. The model is trained by minimising the binary cross–entropy loss:

$$\mathcal{L}_{CE} = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)] \tag{9}$$

where $\hat{p}_i = P(y = 1 \mid \mathbf{x}_i)$ is the predicted probability for the $i$–th sample.

*5) Behavioural and Heuristic Analysis:* Beyond statistical models, the system applies heuristic rules to detect patterns that may escape purely data–driven classifiers. Profile heuristics examine incomplete profile information, abnormal follower ratios, and suspicious username constructions. Message heuristics scan for urgency–based language, manipulation phrases, repetitive message sequences sent in short time intervals, and excessive promotional content. Each triggered heuristic adds a weighted penalty to the final risk score, providing robustness against novel attack patterns.

*6) Risk Scoring and Classification:* The individual outputs from the machine learning models and heuristic rules are aggregated into a composite risk score $S$ defined as:

$$S = \alpha \cdot P_{profile} + \beta \cdot P_{message} + \gamma \cdot H \tag{10}$$

where $P_{profile}$ is the suspicious probability output by the profile classifier, $P_{message}$ is the suspicious probability output by the message classifier, $H$ is a normalised heuristic penalty score, and $\alpha$, $\beta$, $\gamma$ are weighting coefficients with $\alpha + \beta + \gamma = 1$. The score $S \in [0, 1]$ is then mapped to risk levels using predefined thresholds:

$$\text{Risk Level} = \begin{cases} \text{Low} & S < 0.40 \\ \text{Medium} & 0.40 \le S < 0.70 \\ \text{High} & S \ge 0.70 \end{cases} \tag{11}$$

## V. Implementation

The proposed system is implemented as a full–stack web application integrating a React–based frontend, a Node.js backend, a SQLite database, and Python–based machine learning modules. Each layer is designed to fulfil a specific functional role while communicating cleanly with the others through well–defined APIs.

### A. Frontend

The user interface is developed using React.js. It provides two primary interaction panels: one for profile analysis and one for message analysis. Users can enter an Instagram username or paste message content, submit the input, and immediately receive a risk assessment. The dashboard displays the computed risk score, a colour–coded risk level indicator, a breakdown of the features that contributed most to the score, and a historical log of previous analyses performed by the user. The interface is designed to be intuitive and informative, presenting results in plain language accompanied by visual indicators so that non–technical users can interpret the outcome without difficulty.

### B. Backend

The server layer is built on Node.js with an Express.js framework. It receives requests from the frontend, routes them to the appropriate Python analysis scripts via child process calls, and returns the structured results. The backend handles user authentication, session management, and access control. An administrator panel is also supported, allowing authorised users to monitor system–wide analysis activity and review flagged accounts.

### C. Machine Learning Module

The Python–based machine learning module consists of two sub–components: the profile analyser and the message analyser.

The profile analyser loads pre–trained Random Forest, Gradient Boosting, Logistic Regression, and XGBoost models. When a profile is submitted, the system extracts the relevant numerical and categorical features, applies the preprocessing pipeline, feeds the feature vector to each model, and aggregates the prediction probabilities into a profile risk score. The ensemble approach improves robustness by reducing sensitivity to the idiosyncrasies of any single algorithm.

The message analyser uses a TF–IDF vectoriser fitted on the training corpus. Incoming messages are transformed into TF–IDF feature vectors and passed to the Multinomial Naive Bayes and Logistic Regression classifiers. The average of the two probability outputs forms the message risk score. Heuristic rules are then applied on top of the raw text to detect specific scam language patterns.

### D. Database

SQLite is used as the persistence layer. The database stores user account information, analysis records, and flagged content history. Each analysis record captures the input submitted by the user, the extracted feature values, the model predictions, the heuristic findings, the final risk score, and a timestamp. This persistent history enables users and administrators to track patterns over time and revisit earlier assessments.

### E. Risk Score Aggregation

After both sub–analyses are complete, the backend aggregates the profile risk score $P_{profile}$, the message risk score $P_{message}$, and the heuristic penalty $H$ using equation (10) with empirically tuned weights $\alpha = 0.45$, $\beta = 0.35$, and $\gamma = 0.20$. The resulting composite score is then classified into a risk level using the threshold scheme defined in equation (11). The complete result, including the score, the risk level, and an explanation of the most influential contributing factors, is returned to the frontend and displayed to the user.

## VI. Results and Discussion

### A. Experimental Setup

Model training and evaluation were conducted on datasets comprising Instagram profile records and labelled message samples, with an 80:20 train–test split applied consistently across all experiments. Performance was assessed using four standard classification metrics: accuracy, precision, recall, and F1–score. All models were implemented using the Scikit-learn library [23] and trained on standard hardware without GPU acceleration.

### B. Profile Classification Results

Table I summarises the performance of the profile analysis classifiers on the test set.

TABLE I
Profile Classification Performance

| Model | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Logistic Regression | 88% | 0.87 | 0.88 | 0.87 |
| Random Forest | 92% | 0.92 | 0.91 | 0.91 |
| Gradient Boosting | 93% | 0.93 | 0.92 | 0.92 |
| XGBoost | 94% | 0.94 | 0.93 | 0.93 |

XGBoost achieved the highest accuracy of 94% for profile classification. Random Forest and Gradient Boosting also performed competitively, confirming that ensemble methods are particularly well suited to this task. Logistic Regression, while the simplest model, still achieved a respectable 88%, demonstrating that even linear classifiers can capture meaningful profile patterns when the feature set is properly engineered.

### C. Message Classification Results

Table II presents the results for message analysis.

TABLE II
Message Classification Performance

| Model | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Multinomial Naive Bayes | 97% | 0.96 | 0.97 | 0.96 |
| Logistic Regression | 97% | 0.97 | 0.96 | 0.96 |

Both message classifiers exceeded 97% accuracy. The high performance is consistent with findings in the literature that TF–IDF representations combined with probabilistic and linear classifiers are highly effective for detecting scam and spam language in short text messages [24].

### D. Risk Score Evaluation

The composite risk scoring mechanism was evaluated by comparing the final risk level assignments against ground–truth labels across the complete test set. The combined system achieved an overall detection accuracy of 95%, demonstrating that the fusion of machine learning predictions with heuristic behavioural signals improves upon either component individ- ually. False positive rates remained below 5%, indicating that the system does not unduly flag legitimate accounts.

### E. Discussion

The results indicate that the proposed multi–layer approach is effective for Instagram authenticity detection. Profile–based features provide strong discriminative signals for identifying structurally anomalous accounts, while TF–IDF message fea- tures capture the linguistic signatures of scam communication. Heuristic rules complement both by detecting behavioural red flags that statistical models may underweight.

A key advantage of the system is explainability. By surfac- ing the features that most influenced the risk score, the system aligns with the principles of interpretable machine learning [3], [15], [17], enabling users to understand and trust the output rather than simply accepting a binary verdict. This transparency is particularly important in a security context where user awareness is part of the defence.

## VII. Conclusion and Future Work

This paper presented a machine learning–based system for detecting fake and suspicious Instagram accounts by jointly analysing profile structure and message content. The system integrates four profile classifiers (Random Forest, Gradient Boosting, Logistic Regression, and XGBoost) with two mes- sage classifiers (Multinomial Naive Bayes and Logistic Re- gression) supported by TF–IDF feature extraction. A rule– based heuristic layer further augments detection by capturing behavioural signals that purely statistical models may miss. The outputs are fused into a composite risk score that maps to actionable risk levels, accompanied by plain–language ex- planations. Experimental evaluation demonstrated that the system achieves up to 94% accuracy for profile analysis and over 97% accu- racy for message classification, with an overall detection ac- curacy of 95% when both modalities are combined. The full– stack implementation, comprising a React frontend, Node.js backend, SQLite database, and Python machine learning mod- ules, makes the framework practically deployable for individ- ual users who want to independently verify account authen- ticity without relying on opaque platform–controlled mecha- nisms.

Several directions remain open for future investigation. First, the integration of more advanced language models such as BERT [8] or RoBERTa [9] for message analysis could improve detection of semantically subtle scam language. Second, incor- porating graph–based analysis of follower and interaction net- works using Graph Convolutional Networks [16] could reveal coordinated bot clusters that individual account analysis cannot detect. Third, expanding the dataset to include profiles across different languages and cultural contexts would improve the generalisability of the system. Finally, developing a browser extension or mobile application would lower the barrier to adoption and allow real–time verification during actual inter- actions on the platform.

## References


[1] W. Jin et al., "Deep learning for social bot detection: A survey," *IEEE Transactions on Artificial Intelligence*, 2023. https://ieeexplore.ieee.org/ document/10070679

[2] E. Ferrara, "Bots, automation, and disinformation on social media," *Communications of the ACM*, 2022. https://doi.org/10.1145/3513261

[3] C. Molnar, *Interpretable Machine Learning*, 2022. https://christophm. github.io/interpretable-ml-book

[4] Z. Wu et al., "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021. https: //ieeexplore.ieee.org/document/9046288

[5] S. Cresci, "A decade of social bot detection," *Communications of the ACM*, 2020. https://dl.acm.org/doi/10.1145/3409111

[6] F. Pierri, S. Cresci, and L. Luceri, "Misinformation and fake news detection on social media: A survey," *IEEE Access*, 2020. https://doi. org/10.1109/ACCESS.2020.3021042

[7] X. Zhou and R. Zafarani, "Fake news detection: A survey," *ACM Computing Surveys*, 2020. https://doi.org/10.1145/3395046

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019. https://arxiv.org/abs/1810.04805

[9] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019. https://arxiv.org/abs/1907.11692

[10] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018. https://doi. org/10.1126/science.aap9559

[11] D. Cer et al., "Universal Sentence Encoder," 2018. https://arxiv.org/abs/ 1803.11175

[12] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations*, vol. 19, no. 1, 2017. https://doi.org/10.1145/3137597.3137600

[13] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini, "Online human-bot interactions: Detection, estimation, and characterization," in *Proc. ICWSM*, 2017. https://ojs.aaai.org/index.php/ICWSM/article/view/ 14842

[14] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots," in *Proc. 26th International World Wide Web Conference*, 2017. https://doi.org/10.1145/3038912. 3052679

[15] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," 2017. https://arxiv.org/abs/1705.07874

16] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017. https://arxiv.org/abs/1609.02907

[17] M. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. ACM SIGKDD*, 2016. https://doi.org/10.1145/2939672.2939778

[18] E. Ferrara et al., "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016. https://doi.org/10.1145/2818717

[19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2016. https://doi.org/10.1145/2939672.2939785

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. https://www.deeplearningbook.org

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. https://arxiv.org/abs/1301.3781

[22] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of Twitter accounts: Are you a human, bot, or cyborg?" *IEEE Transac- tions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 811–824, 2012. https://doi.org/10.1109/TDSC.2010.75

[23] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. https://jmlr.org/papers/v12/pedregosa11a.html

[24] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proc. 26th Annual Computer Security Applications Conference*, 2010. https://doi.org/10.1145/1920261.1920263

[25] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324