iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85263

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

SocialGuard: An Explainable Multi-Signal Machine Learning Framework for Social Media Fake Account Detection

Prof. Pallavi A. Pathare[1], Krushna Jagtap[2], Aaditya Kulkarni[3], Rohit Gosavi[4], Harshal Gosavi[5], Prof. Suvarna, S. Wakchaure[6], Dr. Rahul M. Dhokane[7]
Department of Computer Engineering, Sir Visvesvaraya Institute of Technology, Nashik, India

Abstract: ***Social media platforms are increasingly exposed to fake, spam, bot-operated and otherwise deceptive accounts that rule-based filters struggle to identify. This paper presents SocialGuard, an explainable multi-signal machine learning framework for fake-account detection. Instead of judging an account from a few profile fields, SocialGuard fuses four independent families of evidence: profile attributes, behavioral patterns, content characteristics and social-network structure. A classification layer that compares Logistic Regression, Random Forest, gradient-boosted trees and a neural network produces a fake-probability and a 0–100 risk score. An explainability layer built on Shapley-value attributions turns each score into evidence-based, feature-level reasoning, and a graph-analysis layer exposes suspicious clusters of accounts with unusually dense interaction. We formulate a research question and a four-step feature-ablation protocol that tests whether combining signal families improves detection over profile-only models. Because the evaluation on real benchmark data is still in progress, we validate the end-to-end pipeline on a controlled synthetic dataset of 3,000 accounts; those numbers demonstrate the methodology and must not be read as real-world performance. The framework is delivered as a web application with a REST API and an investigation dashboard that ends in a final investigation report.***

Keywords: fake account detection; explainable AI; social network analysis; community detection; bot detection; risk scoring; ensemble learning.

I. INTRODUCTION

Social media has become a primary channel for communication, commerce and public discourse, and with its growth has come a large population of fake, duplicate, spam and bot-operated accounts. Such accounts are used for scams, spam campaigns, artificial amplification and misleading activity, and they erode user trust in the platforms on which they operate. Manual review does not scale to the volume of new accounts, and static rules are brittle because adversaries adapt as soon as a rule becomes known. Machine learning (ML) offers an automated alternative that learns suspicious patterns directly from data [1]–[4].

Existing student and research systems commonly follow a narrow pattern: a user enters an account, a single classifier outputs “fake” or “real”, and the decision is presented without justification. This design has three weaknesses. First, it often depends on profile fields alone, which sophisticated fake accounts can imitate. Second, it treats each account in isolation, ignoring the coordinated structure in which fake accounts typically operate. Third, a bare label is hard for a moderator to trust, audit or appeal.

SocialGuard addresses these weaknesses by treating detection as a complete AI system rather than a single model. The main contributions of this work are as follows.

1) A multi-signal detection engine that fuses profile, behavioral, content and network features into one unified feature vector and compares four model families using precision, recall, F1-score, ROC-AUC and PR-AUC.
2) An explainable investigation layer that converts a prediction into a 0–100 risk score, a list of supporting and contradicting evidence, and per-signal-group feature contributions.
3) A suspicious account-cluster detector that applies community detection and graph-density analysis so that coordinated groups can be reported together instead of one account at a time.
4) A reproducible experimental protocol, based on four feature-ablation experiments, that isolates the contribution of each signal family to detection quality.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 defines the problem and its main challenges. Section 4 describes the proposed system. Section 5 gives the evaluation protocol, Section 6 reports a preliminary pipeline validation, and Sections 7 and 8 discuss limitations and conclude.

©IJRASET: All Rights are Reserved

II. LITERATURE SURVEY

Recent work on fake-account detection has largely applied supervised classifiers to account-level features. Table 1 summarizes four representative studies from 2024–2025 that informed this project. The accuracy figures are those reported by the respective authors on their own datasets and are not directly comparable with one another.

TABLE 1. SUMMARY OF RELATED WORK

Paper	Authors (Year)	Methodology	Key Findings	Limitations
Detection of Fake Profiles on Social Networking [1]	J. Sathiya Jothi, A. K. Akash, R. Akash, M. Deepak (2024)	ML-based analysis of profile, behavior, network and content signals.	Separates genuine, suspicious and fake profiles using several indicators.	Needs continuous adaptation to changing fake-account behavior.
Fake Social Media Detection using Machine Learning [2]	R. Thangamani, M. Vimaladevi, J. Mohammed Rifath, G. Harini, S. Hari (2024)	Random Forest, XGBoost and CNN on account features.	High accuracy; XGBoost reported at 99.6%.	Depends on visible data; real-time detection is difficult.
Comparative Analysis of Fake Account Detection using ML Algorithms [3]	A. Padmavathi, K. B. Vaisshnavi (2024)	ANN on profile and behavioral features; compared with SVM, KNN and Logistic Regression.	ANN reached 89.66% accuracy, above the other models tested.	Needs quality training data and heavy compute at scale.
Fake Social Media Accounts Detection using ML [4]	K. Bhargavi, P. M. D. Ali Khan, P. Imran Khan, T. Sivakrishna (2025)	Decision Tree and Random Forest on profile and behavioral features.	Decision Tree reached 97.45% accuracy.	Evolving behavior and real-time scalability remain challenging.

Three observations follow from this survey. First, tree-based ensembles and neural networks are the dominant model families, and the highest reported accuracies come from tabular account features [2], [4]. Second, most studies use profile and behavioral features only, with network and content signals appearing less often, even though [1] explicitly argues for combining all four. Third, every study lists adaptation to evolving fake-account behavior and real-time use as open limitations, and none of the surveyed works centers explanation of individual predictions or coordinated-cluster discovery. SocialGuard is designed to fill precisely these gaps: it combines all four signal families, evaluates them with imbalance-aware metrics, and makes explanation and cluster detection first-class outputs.

III. PROBLEM DEFINITION AND CHALLENGES

A. *Problem Definition*

Given a social-media account *a* described by profile attributes, behavioral records, public content and its position in a social graph, the task is to (i) classify *a* as genuine or potentially fake, (ii) produce a risk score that expresses the estimated likelihood that *a* is fake, and (iii) provide explainable evidence for the prediction, including whether *a* belongs to a suspicious cluster. The system should therefore be an automated, multi-signal ML pipeline rather than a threshold on any single attribute such as follower count.

B. *Major Challenges*

Six challenges shape the design. Evolving fake accounts: adversaries create increasingly sophisticated and realistic accounts, so signals that work today may fail tomorrow. Limited and noisy data: public datasets are incomplete, imbalanced and noisy. Class imbalance: genuine accounts far outnumber fake or suspicious ones, which makes raw accuracy misleading; a classifier that labels every account genuine can score high accuracy while detecting nothing. Data and concept drift: feature distributions and attacker tactics change over time, which requires monitoring and periodic retraining. Explainability: moderators need to understand why an account was flagged. Scalability and privacy: the system must handle large volumes of data while protecting user information.

IV. PROPOSED SYSTEM

A. Architecture Overview

Figure 1 shows the SocialGuard architecture. An account enters from a structured dataset (CSV upload), or, where a legitimate API is available, through a platform adapter that feeds the same engine. Profile, behavioral and content features are extracted and fused into a unified feature vector, which the ML classification layer converts into a fake-probability and risk level. An explainability module attributes the prediction to individual features, and a network-analysis module computes graph-level scores and detects suspicious clusters. The results are combined in an investigation dashboard and a final investigation report.

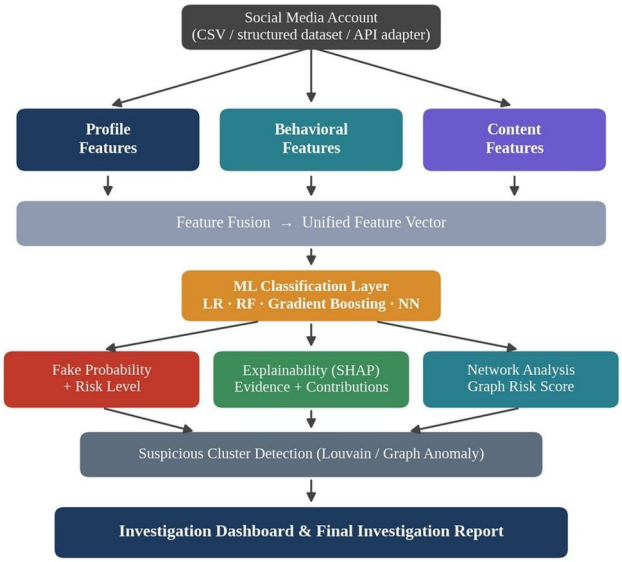


Fig. 1. Architecture of the proposed SocialGuard framework.

B. Data Sources and Unified Schema

A system trained on one dataset inherits that dataset's biases, so SocialGuard builds a unified dataset from several sources wherever licensing and access permit: public fake-account datasets from Kaggle, the TwiBot-20 Twitter bot-detection benchmark [5], other public research datasets with profile, behavioral, content and network data, and a custom structured dataset derived from them after preprocessing and feature engineering. All sources are normalized into a single schema so that the same pipeline can consume any of them. Table 2 lists the signal families and the features extracted for each.

Table 2. Feature Families In The Unified Schema

Signal family	Features
Profile	Account age, followers, following, follower/following ratio, number of posts, bio length, profile completeness, profile-image availability, username characteristics, external links, verified status (where available)
Behavioral	Posting frequency, posting-time patterns and variance, follow/unfollow behavior, follower-growth pattern, engagement ratio, repeated interactions, burst activity, abnormally regular activity
Content	Text similarity, repeated captions, average caption length, hashtag ratio, sentiment distribution, URL frequency, spam-like keywords, duplicate-content ratio
Network	Degree, in-degree, out-degree, clustering coefficient, PageRank, community membership, mutual-follow ratio, cluster density

C. Multi-Signal Feature Extraction

Profile signals capture the static footprint of an account, such as its age, completeness and follower/following balance. They are cheap to compute but easy for a careful adversary to imitate. Behavioral signals describe how the account acts over time; abnormally regular posting, burst activity and unnatural follower growth are difficult to disguise and are typically more informative than static fields. Content signals are computed from public posts and expose automation through duplicated or near-duplicated captions, heavy URL usage and spam-like vocabulary; sentence embeddings from Hugging Face Sentence Transformers can be used to measure text similarity. Network signals represent the social graph and extract the structural position of the account, including degree, clustering coefficient, PageRank, community membership and the density of the community around it. Because fake accounts are frequently operated in groups that follow and interact with one another, network features can reveal coordination that no single-account feature can show.

D. Feature Fusion and Classification

The four feature groups are concatenated into a unified feature vector. Skewed count features such as followers, posts and degree are log-transformed, and features are standardized for scale-sensitive models. Rather than relying on one algorithm, SocialGuard compares four model families: Logistic Regression as an interpretable linear baseline, Random Forest [6] as a robust bagged-tree ensemble, gradient-boosted trees (XGBoost or LightGBM) [7] as a strong tabular learner, and a feed-forward neural network. Class imbalance is handled with class weighting and, where appropriate, resampling. Models are compared on precision, recall, F1-score, ROC-AUC, PR-AUC and the confusion matrix. Precision and recall matter because a false positive suspends a genuine user while a false negative leaves a fake account active, and PR-AUC is particularly informative under heavy imbalance, where raw accuracy can be misleading.

E. Risk Scoring and Explainable AI

The calibrated fake-probability is mapped to a risk score between 0 and 100 and to a risk level (low, medium or high). To make the score actionable, SocialGuard does not report a bare label such as “FAKE — 92%.” It reports the score together with evidence, as in the investigation report of Figure 6: indicators that raise risk (for example, an imbalance between following and followers, a recently created account, repetitive captions or membership in a suspicious interaction cluster) and indicators that lower it (for example, a populated biography or normal variation in posting time). Model-level and instance-level attributions are computed with SHAP [8], which assigns each feature an additive contribution to the prediction. Contributions are also aggregated by signal family, so that an investigator can see at a glance how much of the risk stems from network behavior, account behavior, content patterns or profile signals.

F. Suspicious Account-Cluster Detection

Detecting a group of accounts is more informative than scoring accounts independently. SocialGuard represents the platform as a graph, extracts the network features in Table 2 with NetworkX [10], and partitions the graph with the Louvain algorithm [9] (Leiden is an alternative). For each community it computes the internal edge density and combines it with the aggregate fake-probability of its members into a graph-based anomaly score. A community that is small, internally dense and composed largely of high-risk accounts is reported as a suspicious cluster, for example: “five accounts exhibit unusually dense interaction patterns and similar behavioral characteristics.” A graph neural network is not required for this purpose, which keeps the approach lightweight and interpretable.

G. Implementation Stack

The system is implemented in Python with Pandas, NumPy, Scikit-learn [11], XGBoost, SHAP, NetworkX and Hugging Face Transformers. A FastAPI backend exposes four endpoints: POST /predict returns the fake-probability and risk score, POST/explain returns evidence and feature contributions, POST/network/analyze returns graph metrics and cluster membership, and GET /model/metrics returns evaluation results. A PostgreSQL database stores accounts, features, predictions, explanations, network edges, communities and model versions. A React dashboard built with Vite presents the account input, risk score, classification, evidence list and network graph. Development uses Visual Studio Code, Jupyter Notebook and Git/GitHub on a workstation with at least an Intel Core i5 (10th generation) or AMD Ryzen 5 processor, 16 GB of RAM and a 512 GB SSD.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

H. Scope Decision: Structured Input Instead of Scraping

SocialGuard deliberately does not depend on scraping Instagram, X or Facebook. Scraping introduces API restrictions, authentication difficulties, rate limits, instability, terms-of-service concerns and poor reproducibility. The engine instead accepts a structured profile and network dataset (CSV upload followed by feature extraction and prediction), and a data adapter can later connect a legitimate platform API to the same detection engine. This separation keeps the architecture clean and the experiments reproducible

V. EXPERIMENTAL DESIGN

A. Research Question

The central research question is: *Does combining profile, behavioral, content and network-level signals improve fake-account detection compared with profile-only machine learning models?* To answer it, the framework uses a four-step feature-ablation protocol in which one signal family is added at a time.

B. Experiments

Experiment 1 trains a Random Forest on profile features only and serves as the baseline. Experiment 2 adds behavioral features, Experiment 3 adds content features, and Experiment 4 adds network features and uses a gradient-boosted model. To separate the effect of the added features from the effect of the model change, Experiment 4 should also be run with Random Forest. All experiments use the same stratified split and the same metrics: precision, recall, F1-score, ROC-AUC and PR-AUC. A complete study should additionally report cross-validated means with confidence intervals and evaluate on at least one dataset not used for training, to test generalization.

VI. PRELIMINARY PIPELINE VALIDATION

A. Important Note on the Data

Evaluation on real data has not yet been completed. To verify that the pipeline runs end to end and to illustrate the form of the outputs, we executed the full protocol on a synthetic dataset generated by a script. The dataset contains 3,000 accounts (616 fake, 20%) described by 23 features in the four families of Table 2. Network features were computed from a simulated social graph of 15,364 edges in which most fake accounts form dense groups, and Louvain detection returned 31 communities. A stratified 70/30 split produced 2,100 training and 900 test accounts. In the generator, a share of fake accounts deliberately imitates genuine profile, behavioral or content statistics. The reported values therefore reflect how the data were constructed and are not evidence of real-world performance; they will be replaced by results from the real datasets in Section 4.2. Two substitutions were made because of the execution environment: scikit-learn's histogram-based gradient boosting ("GB") stands in for XGBoost, and the signal-group contributions in Figure 6 come from a linear-model decomposition (coefficient × standardized value) instead of SHAP.

B. Feature-Ablation Results

Table 3 and Figure 2 report the ablation on the held-out test set. On this synthetic data, recall rises from 0.454 with profile features alone to 0.908 once behavioral, content and network features are added to the Random Forest, and F1 rises from 0.598 to 0.946. Replacing Random Forest with gradient boosting in the last step gives F1 = 0.965. This pattern is the kind of evidence the real experiments are designed to test, but its size here is a consequence of the simulation.

Table 4 compares the four model families on all features, and Figure 3 shows their ROC curves. All four reach high scores on this data and differ only slightly, so no ranking of the algorithms should be inferred from a single split of synthetic data. Gradient Boosting obtained the highest F1-score (0.965), and its confusion matrix is shown in Figure 4: of 185 fake test accounts, 178 were detected and 7 were missed, while 6 of 715 genuine accounts were wrongly flagged.

Table 4. Model Comparison Using All Features (Synthetic Data)

Model	Precision	Recall	F1	ROC-AUC	PR-AUC
Logistic Regression	0.913	0.968	0.940	0.996	0.987
Random Forest	0.988	0.908	0.946	0.998	0.993
Gradient Boosting	0.967	0.962	0.965	0.997	0.991
Neural Network (MLP)	0.950	0.930	0.940	0.992	0.984

Table 3. Feature-Ablation Results On Synthetic Data (Held-Out Test Set)

Configuration	Precision	Recall	F1	ROC-AUC	PR-AUC
Profile only (RF)	0.875	0.454	0.598	0.770	0.656
+ Behavior (RF)	0.949	0.708	0.811	0.936	0.890
+ Content (RF)	1.000	0.768	0.869	0.973	0.951
+ Network (RF)	0.988	0.908	0.946	0.998	0.993
+ Network (GB)	0.967	0.962	0.965	0.997	0.991

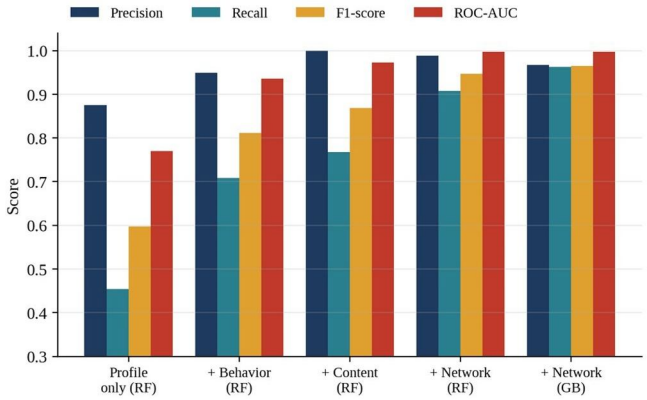


Fig. 2. Effect of adding signal families on detection quality (synthetic data).

C. Model Comparison

Figure 3 compares the ROC curves of four machine learning models: Logistic Regression, Random Forest, Gradient Boosting, and Neural Network (MLP). The models were evaluated using a synthetic dataset containing profile, behavioral, content, and network features. The ROC-AUC values were 0.996 for Logistic Regression, 0.998 for Random Forest, 0.997 for Gradient Boosting, and 0.992 for Neural Network (MLP). Random Forest achieved the highest ROC-AUC of 0.998, indicating strong discrimination between genuine and fake accounts on this synthetic dataset.

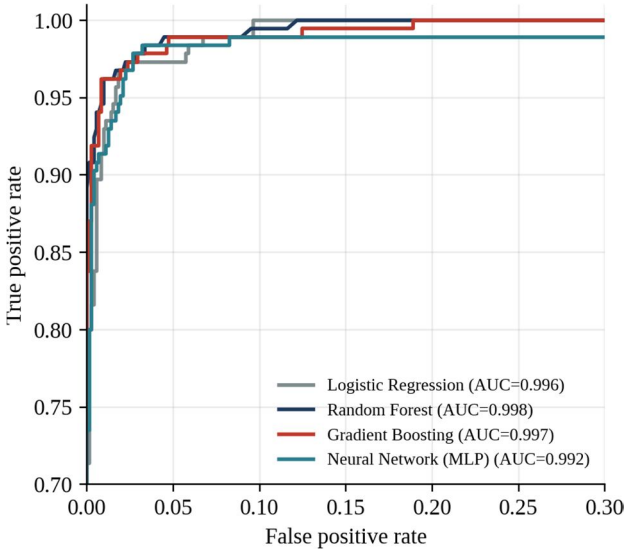


Fig. 3. ROC curves of the four model families (synthetic data; axes zoomed).

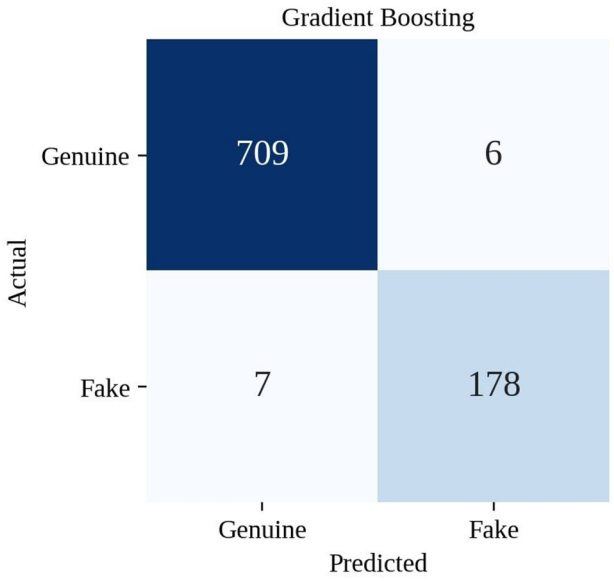


Fig. 4. Confusion matrix of the best-scoring model (Gradient Boosting).

D. Explainable Investigation Report

Figure 5 shows the investigation report for one fake test account that the model scored at 92/100. The gauge gives the overall risk, the group bars show that content and network signals drove the score while profile signals actually pointed toward "genuine," and the feature bars show the individual contributions behind each group. The last panel restates the evidence in plain language. This example illustrates why the explanation layer is useful: the account imitates a genuine profile, so a profile-only model would be likely to miss it, yet the content and network evidence still exposes it.

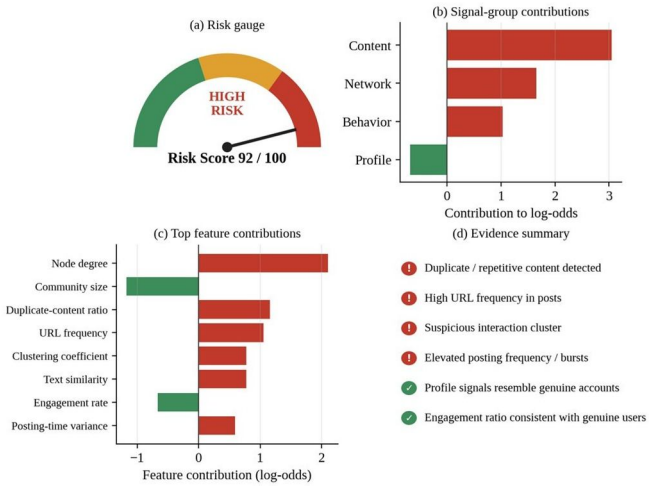


Fig. 5. Example investigation report with risk gauge, signal-group contributions, feature contributions and evidence (synthetic data; illustrative).

E. Suspicious Cluster Detection

Figure 6 shows a cluster discovered by Louvain community detection. The 12 accounts in the cluster have an internal edge density of 0.68, whereas the large genuine communities in the same graph have densities between roughly 0.02 and 0.04. Every member of the cluster is a fake account in the simulation. Reporting such a group as a single finding is more informative to an investigator than twelve separate scores, and it illustrates the added value of the network layer.

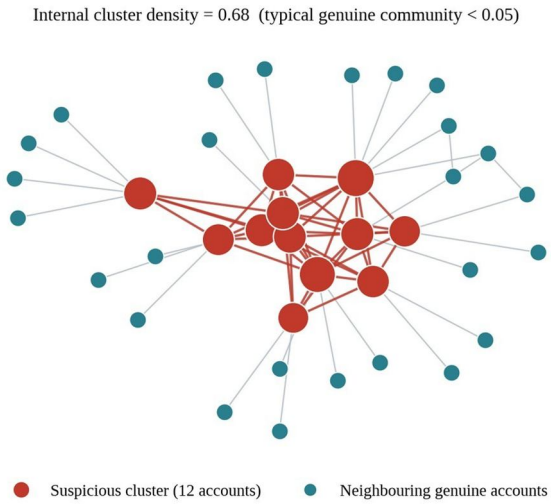


Fig. 6. A suspicious account cluster detected by community analysis (synthetic data). Node size reflects PageRank within the subgraph.

VII. DISCUSSION AND LIMITATIONS

The preliminary validation confirms that each stage of SocialGuard—feature fusion, multi-model comparison, risk scoring, explanation and cluster detection—works together in one pipeline, but it cannot establish how well the approach performs on real platforms. Several limitations must be addressed in the full evaluation. Real labels are noisy, and fake accounts adapt, so performance measured on a static benchmark will degrade unless models are monitored for data and concept drift and retrained. Network features require graph data, which many public datasets lack and which platforms may restrict. SHAP explanations describe the model's behavior and do not prove that an account is fake, so risk scores should support human review rather than trigger automatic punishment, particularly because false positives affect genuine users. Finally, account data are personal data; the system should minimize the data it stores, use only data that are legitimately available, and respect platform terms and applicable privacy law.

VIII. CONCLUSION AND FUTURE WORK

This paper presented SocialGuard, an explainable multi-signal framework that detects fake social media accounts by fusing profile, behavioral, content and network evidence, explains each prediction through a risk score and feature-level evidence, and reveals coordinated groups through cluster detection. A four-step ablation protocol was defined to test whether multi-signal fusion improves on profile-only models, and the pipeline was validated end to end on synthetic data. Future work will (i) build the unified dataset from Kaggle, TwiBot-20 and other public sources and run the experiments on real data with cross-validation and external testing, (ii) replace the stand-in components with XGBoost and SHAP as planned, (iii) add drift monitoring and periodic retraining, and (iv) implement the FastAPI backend, PostgreSQL storage and React investigation dashboard, with optional data adapters for legitimate platform APIs.

IX. ACKNOWLEDGMENT

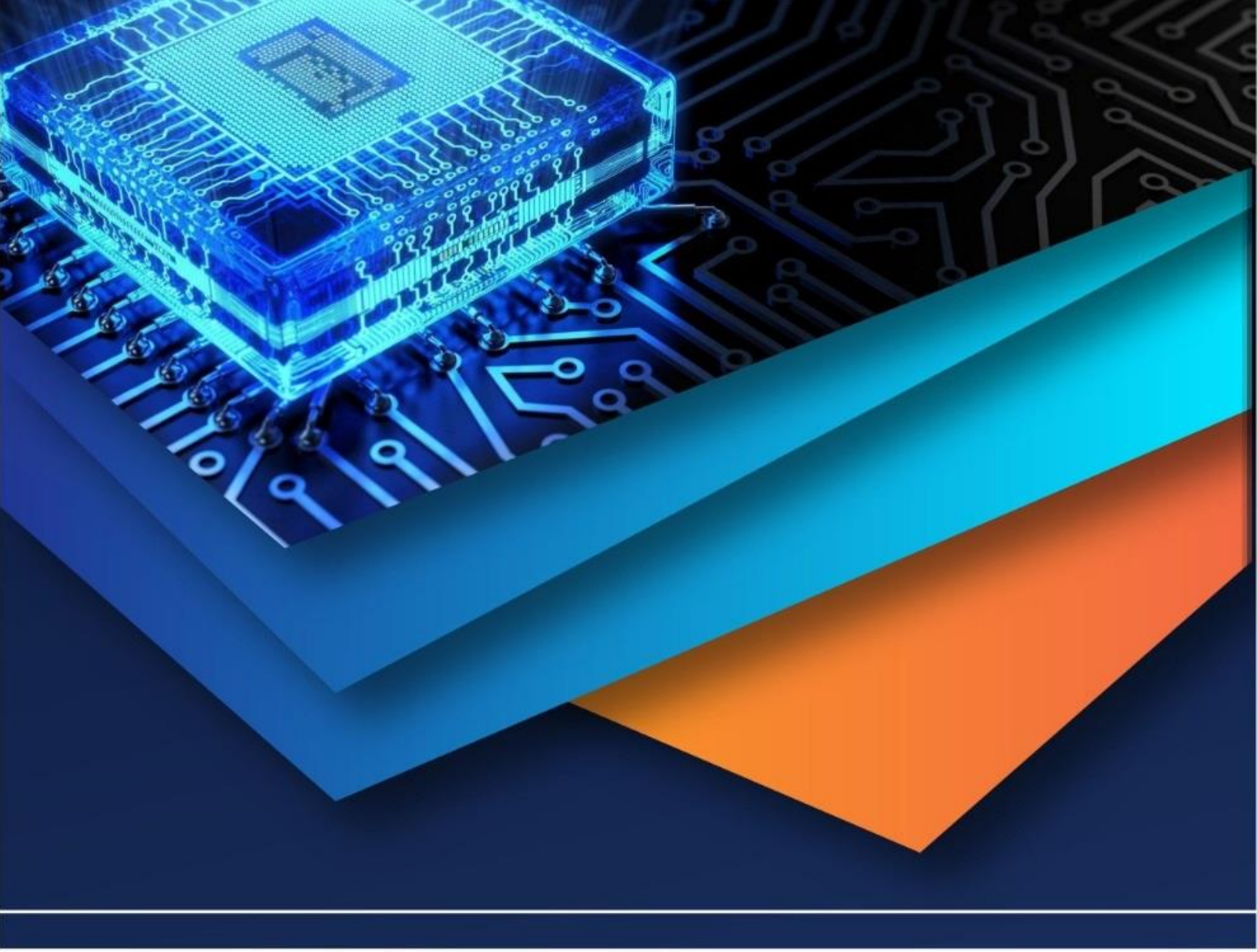
The authors thank their project guide, Prof. Pallavi A. Pathare, the project coordinator, Prof. Suvarna S. Wakchaure, and the Head of the Department of Computer Engineering, Dr. Rahul M. Dhokane, for their guidance and support.

REFERENCES

[1] J. S. Jothi, A. K. Akash, R. Akash, and M. Deepak, "Detection of fake profiles on social networking," 2024.
[2] R. Thangamani, M. Vimaladevi, J. Mohammed Rifath, G. Harini, and S. Hari, "Fake social media detection using machine learning," 2024.
[3] A. Padmavathi and K. B. Vaisshnavi, "Comparative analysis of fake account detection using machine learning algorithms," 2024.
[4] K. Bhargavi, P. M. D. Ali Khan, P. Imran Khan, and T. Sivakrishna, "Fake social media accounts detection using machine learning," 2025.
[5] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "TwiBot-20: A comprehensive Twitter bot detection benchmark," in Proc. 30th ACM Int. Conf. Information and Knowledge Management (CIKM), 2021.
[6] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[7] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems 30 (NeurIPS), 2017.
[9] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, no. 10, P10008, 2008.
[10] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in Proc. 7th Python in Science Conference (SciPy), 2008, pp. 11–15.
[11] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)