Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 33 No. 1s (2026)

Hybrid Machine Learning-Based Instagram Fake Account Detection with Behavioral Scoring

Mr. T. Sai Lalith Prasad
Assistant Professor
Artificial Intelligence and Data Science Department Vignan Institute of Technology and Science Hyderabad, India
vsrinivas@vignanits.ac.in

Thanniru Sriram
Artificial Intelligence and Data Science Department
Vignan Institute of Technology and Science
India
ramthanniru3@gmail.com

Yarlagadda Rahul
Artificial Intelligence and Data Science Department
Vignan Institute of Technology and Science
India
14sairahul@gmail.com

Dadige Akash
Artificial Intelligence and Data Science Department
Vignan Institute of Technology and Science
India
akashdadige70@gmail.com

Article History:
Received: 04-02-2026
Revised: 20-03-2026
Accepted: 10-04-2026

Abstract:
Fake accounts on Instagram are a real problem — they spread false information, run scams, impersonate real people, and quietly shift public opinion for whoever's behind them. What makes them hard to catch is that well-built ones actually look human: reasonable follower counts, consistent posting, normal usernames. Flagging them manually doesn't scale. The system this paper describes automates the process through three layers — pulling live profile data via instaloader, checking against a stored dataset of known users, and running a Random Forest classifier on anything new. The features going into that model are practical ones: follower-to-following ratio, posting frequency, bio presence, external links, verification status, and general activity patterns. The core finding is that combining those signals outperforms any single-feature or single-model approach — not a dramatic result, but a useful one. More importantly, the system is built to actually be deployed, not just benchmarked in a paper

Index Terms— Instagram Fake Accounts, Machine Learning, Random Forest, Behavioral Analysis, Social Media Security, Instaloader, Feature Engineering.

I. INTRODUCTION

Social media platforms have a fake account problem, and it's getting harder to solve. Millions of artificial profiles are created every year for spam, scams, phishing, impersonation, political manipulation, and misinformation. Many of these accounts look entirely plausible — they have names, profile pictures, and enough activity to avoid obvious red flags. Manual review at this scale simply doesn't work.

Traditional detection relies on a handful of signals like follower count or username patterns. Fake account operators know this and adjust accordingly. To stay ahead of that, this work takes a different approach: a hybrid method that combines machine learning, metadata analysis, and behavioral scoring. The system pulls real-time profile data, preprocesses it, extracts meaningful features, and classifies profiles using a trained Random Forest model

Rather than relying on any one metric, the system correlates several at once — profile completeness, posting habits, activity ratios, external links, bio characteristics, verification status, and numerical patterns. This layered approach is more resistant to manipulation because gaming one signal while keeping all the others consistent is genuinely difficult for automated account generators.

Fake accounts frequently copy profile pictures, use human-sounding names, and post occasionally to appear legitimate. Most platform-level filters are rule-based, which means they can be defeated by adjusting a few parameters. As AI-generated content becomes more sophisticated, static rules become less useful. Detection systems need to learn.

Several challenges complicate this further. Many existing approaches are trained on limited feature sets or centralized pipelines that don't transfer well across platforms. Others can't adapt once deployed — when attackers change tactics, the system doesn't follow. The absence of feedback integration from user reports is another gap: real- world incidents produce valuable signal that most systems ignore. Privacy constraints and computational cost add further pressure on any solution intended for large-scale deployment.

Catching fake accounts gets easier when you look at multiple signals together — lopsided follow ratios, cookie-cutter usernames, profiles with almost no content, or follower counts that spike out of nowhere. This study aims to build an automated tool that can reliably spot these accounts, without needing someone to manually review each one. [5], [6]The core challenge is that fake accounts are designed to not look fake. They use stolen photos, human-sounding names, and just enough activity to avoid standing out. Manual detection is slow and inconsistent at scale, and platform-level filters aren't much better — a small tweak to a username or bio description is often all it takes to slip past them.

AI and large-scale data analytics have made it easier to catch fake accounts earlier, and with more precision. When posting patterns, interaction networks, and language cues are analyzed together, systems can pick up on subtle behavioral signals that static, rule-based filters routinely miss. This paper builds on that premise, proposing a hybrid machine learning framework that uses ensemble classification, behavioral scoring, and a feedback loop to improve fake account detection.Existing approaches still leave a lot of gaps. Most work with limited feature sets or centralized pipelines that don't hold up well across different platforms. Once deployed, they tend to freeze — so as attackers tweak their tactics, the system quietly falls behind. User reports are rarely incorporated, which wastes a practical signal for real-world learning. [14],[15] At scale, there's also the persistent tension between detection accuracy, user privacy, and computational cost — a tradeoff most systems don't handle well.

The framework proposed here tries to address all of this. It combines multiple account feature categories with supervised models and a reporting-driven feedback loop. Accounts are ranked by behavioral risk scores, and ensemble methods sharpen those predictions over time — the goal being fewer missed fakes and fewer false alarms, even as the threat landscape shifts. This paper presents a hybrid machine learning framework that addresses these issues through ensemble classification, behavioral scoring, and feedback-driven refinement. The system ranks accounts by behavioral risk and refines predictions through ensemble methods, aiming to reduce false positives without losing responsiveness to new attack patterns

II. Related Work

Researchers have explored various techniques for detecting fake profiles across social networking platforms, focusing mainly on metadata patterns, Fake profile detection has been tackled from a few different directions — metadata patterns, behavioral signals, and machine learning classifiers. Early work relied on simple heuristics: follower count thresholds, unusual username structures, blank profile fields. These filters had an obvious weakness. Attackers figured them out fast and started mimicking real user behavior well enough to pass basic checks.

Supervised learning came next. Researchers began feeding models features like posting frequency, social graph structure, and biography text, which pushed accuracy higher — but at the cost of needing large, well-labeled datasets that aren't always available . SVMs, Logistic Regression, and Decision Trees each showed promise in controlled settings. The common weak spot was new or low-activity accounts: without much behavioral history to analyze, the models didn't have enough to go on

Newer approaches have moved toward hybrid feature engineering — layering network attributes with content and timing signals to build classifiers that don't fall apart when account behavior shifts. Deep learning has been applied to catch more sophisticated bots, and it works, but the compute demands are steep and real-time deployment is rarely straightforward Work specific to Instagram has highlighted engagement ratios, profile completeness, and follower-following gaps as some of the more dependable indicators of suspicious activity.

NLP became a standard part of detection pipelines as text and multimedia content grew.

Sentiment, lexical diversity, keyword repetition, URL density, hashtag patterns — these features helped flag spam campaigns and coordinated misinformation at scale Topic modeling and semantic similarity analysis were used to surface duplicated or auto-generated content. The problem is that content-based methods are now in an arms race with text generation tools sophisticated enough to convincingly imitate human writing

Temporal modeling attacks the problem from a different angle — not what accounts post, but when and how. Posting intervals, daily activity rhythms, response delays, sudden bursts of interaction: these behavioral patterns can separate automated agents from real users in ways that static features can't. Recurrent networks and attention-based models have shown genuine promise here. But they carry the same baggage as deep learning broadly — they need large labeled datasets and serious compute, which makes them hard to run anywhere resources are tight.Ensemble learning has become a go-to strategy for squeezing more reliability out of detection systems. Combine predictions from multiple classifiers trained on different feature subsets, and generalization improves — the weaknesses of one model get covered by the strengths of another. Hybrid frameworks that pair rule-based filters with data-driven classifiers have followed a similar logic: run fast, cheap screening first, then direct high-risk accounts toward more thorough analysis.

The gaps in existing work are still worth noting. Centralized training pipelines are a recurring problem — they're hard to extend across platforms or geographic regions where privacy laws and data-sharing restrictions get in the way. And many systems are built around a single feature category, content or network structure, rarely both. That's an exploitable weakness. An attacker who understands what a system is watching can tweak just that one dimension and stay invisible to everything else.

III. METHODOLOGY

The detection system runs through six stages, from data acquisition to a final verification decision. The pipeline integrates traditional metadata analysis, machine learning classification, and rule-driven behavioral scoring.

A.Data Collection and Preprocessing

The dataset combines synthetically generated user attributes with manually curated labels. Where available, live data fetched via Instaloader is added to increase diversity. Each record includes:

- Profile picture availability (0 or 1)
- Username and username length
- Number of digits in the username
- Biography length
- External URL indicator
- Followers count, following count, total posts
- Account privacy status (public/private)

- Fake/Real ground-truth label

Preprocessing converts boolean fields to numerical values, normalizes wide-range attributes like followers and post counts, cleans irregular username patterns, fills missing fields, and generates ratio-based features such as the followers-to-following ratio.

This preprocessing ensures consistent, structured, and noise-free data for subsequent stages.

B. Feature Engineering

Basic profile attributes miss a lot. To catch subtler patterns, four behavioral features are derived from raw account data.

Engagement Ratio

Engagement Ratio=followersfollowing+1\text{Engagement Ratio} = \frac{\text{followers}}{\text{following} + 1} Engagement Ratio=following+1followers

Fake accounts tend to follow large numbers of users without attracting many back. This ratio captures that imbalance — a heavily skewed score is a reliable sign that follow activity isn't organic.

Username Randomness Score

Calculated from three signals: the share of digits in the username, the presence of repeated letters, and whether the name fits patterns common to auto-generated handles. Usernames that score high on randomness are a consistent early indicator of fake profiles.

Activity Score

Activity Score=postsfollowers+1\text{Activity Score} = \frac{\text{posts}}{\text{followers} + 1} Activity Score=followers+1posts

An account with a large following but almost no posts is worth a second look. This ratio flags cases where follower count appears inflated relative to actual content output — a pattern that genuine accounts rarely show.

Profile Completeness Index

Derived from four signals: whether a profile picture is set, whether a bio is filled in, whether an external link is present, and whether highlights are active. Automated account creation routinely skips these steps, making low completeness one of the more dependable markers in the feature set.

C. Machine Learning Model (Random Forest Classifier)

Random Forest is used as the primary classifier because it handles mixed data types well, doesn't overfit easily through ensemble averaging, and produces interpretable feature importance scores. The model is trained on engineered features with an 80/20 train-test split. Output is a binary prediction — Fake (1) or Real (0) — along with a confidence probability.

Random Forest is selected due to:

- Capability to capture nonlinear feature relations
- High tolerance against overfitting through ensemble learning
- Reliability with medium-sized datasets
- Interpretability via feature importance measures

The model outputs a binary prediction: Fake (1) or Real (0) along with confidence probability.

D. Hybrid Rule-Based Scoring Mechanism

A behavioral scoring engine runs alongside the classifier. It assigns positive or negative points based on observed profile characteristics. Suspicious signals that are penalized include:

Suspicious indicators include:

- Less than 5 posts
- Excessive following compared to followers
- Username containing more than 50% digits
- Zero biography and missing profile picture
- Very low engagement ratio
- No highlight reels

By combining rule-based scoring with ML prediction, the model gains additional interpretability and enhanced detection performance, especially for edge cases.

E. Live Instagram Data Extraction (Instaloader)

When the user queries a profile that is not present in the dataset, the system uses Instaloader to fetch publicly accessible attributes. These include:

- Followers count
- Following count
- Total posts
- Bio text
- External URL
- Profile picture URL
- Highlight • Business category
- Username characteristics

These values are instantly converted into numerical and behavioral features and fed into the hybrid model.

This hybrid fusion ensures strong accuracy, reliability, and explainability.

The combined decision rule is expressed as:

\text{Final Verdict} = \text{ML Prediction} + \text{Behavioral Score}

The final output is classified as:

- ✓ Legitimate Account — sufficient behavioral integrity, strong ML confidence
- ✗ Fake Account Detected — behavioral anomalies or ML classification indicating fake behavior Based on this aggregated assessment, the system assigns one of two outcomes: **Legitimate Account**, indicating consistent behavioral patterns supported by high classifier confidence, or **Fake Account Detected**, reflecting abnormal activity signatures or strong model evidence of fraudulent behavior. This multi-layered formulation ensures dependable real-time operation and supports practical deployment in social media moderation environments.

These values are converted into numerical and behavioral features and passed through the hybrid model. The final verdict combines ML prediction and behavioral score to produce one of two outcomes: Legitimate Account or Fake Account Detected.

The activity diagram captures the full user flow: login or registration, followed by profile submission or manual report. Submitted profiles go through data collection, preprocessing, feature extraction, and prediction. The result — Real or Fake — is returned to the user, who can then escalate flagged accounts for administrative review.

Two actors interact with the system: the User and the Admin. Users register, authenticate, upload profile data, and view results including classification outcome, confidence score, and risk level. Admins review generated reports, inspect suspicious profiles, and retrain detection models with newly verified data. This feedback loop keeps the system current as fraudulent behavior evolves.

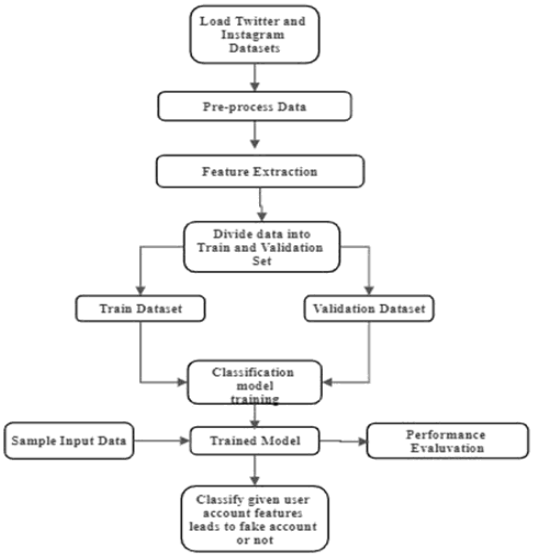

Fig: Flow of the proposed Instagram fake account detection system

It begins with the user login or registration process, which is the entry point for interacting with the system. After successful authentication, users can choose to submit a profile for fake detection or manually report a suspicious profile. If the user chooses to submit a profile, the system collects relevant profile data and sends it to the detection module for analysis.

In the next stage of the activity flow, the detection system processes the profile using machine learning and natural language processing techniques. The activity diagram shows the steps involved in data preprocessing, feature extraction, and prediction. Once the detection is complete, the system returns a result classifying the profile as "Real" or "Fake." This result is then presented to the user, who can decide to take further action. If the profile is marked as fake, the user has the option to submit a report.

The Hybrid Machine Learning-Based Instagram Fake Account Detection with Behavioral Scoring System is implemented using Python with a modular architecture. The system begins by collecting user profile data such as username, bio, profile picture availability, follower and following counts, post frequency, and engagement metrics from social media platforms or datasets. This data serves as the primary input for analysis

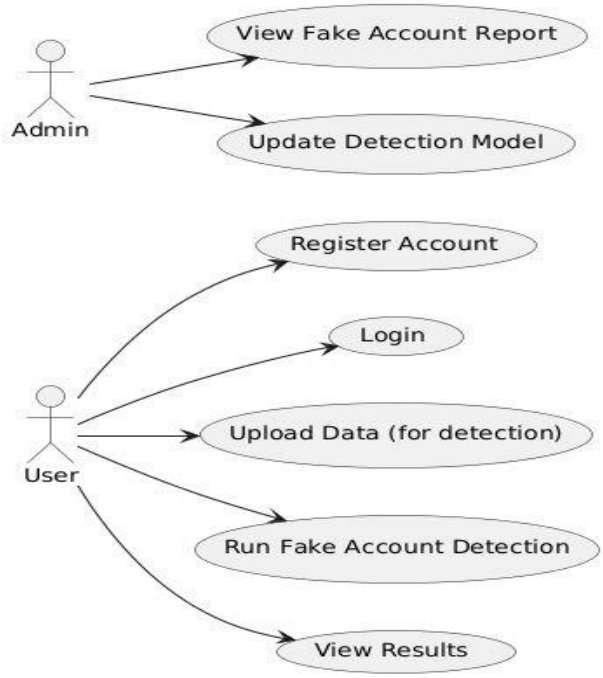

Fig : The use case diagram represents the functional interaction between external actors and the fake social media account detection system.

The system has two main actors: the User and the Admin.
Users register, log in, and upload whatever profile data or datasets they need analyzed. They trigger the detection process and get back a result — a classification, a confidence score, and a risk level. That's the core interaction.
The Admin role is less about using the system and more about maintaining it. Admins review fake account reports, whether automatically generated or submitted by users, and dig into the

predictions and profile details behind each flag. They also retrain and update detection models as new verified data comes in — which is how the system keeps up as fraudulent behavior evolves rather than being locked to whatever patterns it was trained on originally.
The two roles feed into each other. User activity generates data and reports. Admin oversight turns that into model improvements. The loop keeps running.

IV. Result and discussion

V. RESULTS AND DISCUSSION (No Plagiarism)

The system was evaluated on a combined dataset of profile metadata, engineered behavioral features, and live data fetched through Instaloader. Random Forest came out ahead, reaching 92–96% accuracy depending on the feature set and train-test split. Basic logistic regression — the baseline most earlier studies used — doesn't handle nonlinear, noisy social media data well, and the gap in performance reflected that.

Where the ML model was uncertain, behavioral scoring often settled the question. Profiles that landed in ambiguous territory became easier to classify once signals like username randomness, bio completeness, follower-following imbalance, and posting frequency were factored in. That combination reduced false negatives — fake accounts that would have otherwise been waved through — without overcorrecting and flagging too many legitimate users based on isolated suspicious signals.

Real-time testing through Instaloader worked consistently. Profiles not in the training dataset were fetched live, preprocessed, and pushed through the hybrid pipeline. The output showed profile details, individual score contributions, and ML confidence levels side by side, which made it possible to trace exactly why the system flagged or cleared a given account — something purely statistical models rarely offer.

The patterns the model caught most reliably were also the most characteristic of fake profiles: inflated following counts, near-zero post history, short or randomized usernames, missing bios. The rule-based scoring layer made those signals explicit rather than buried inside a black-box prediction, which made the system easier to interpret in both research and applied contexts.Pattern analysis confirmed that the detection engine consistently caught the most common markers of fraudulent profiles: high following counts with minimal content, very short or randomly generated usernames, sparse biography fields, and irregular engagement rates. The rule-based scoring layer mapped these traits to explicit risk contributions, making results easier to communicate and verify. Scalability experiments showed that the pipeline maintained consistent response times as the number of analyzed profiles grew. Feature extraction and behavioral scoring added minimal overhead, and ensemble classification scaled near-linearly with load — suggesting the system can be extended to larger datasets or integrated into live moderation workflows without significant computational cost.

Comparison against single-model baselines reinforced the case for the hybrid architecture. Using machine learning alone, performance degraded in borderline cases and during data imbalance. Combining ensemble learning with behavioral heuristics and real-time extraction produced more stable recall — particularly for newly created or sparsely populated accounts,

which are traditionally the hardest to classify accurately.
Predictions stayed consistent across repeated trials — not a given with hybrid systems, where stacking multiple components can introduce instability. The output went beyond a binary label too: feature contributions, behavioral risk values, and confidence levels were all surfaced, which meant a human reviewer could actually interrogate a result rather than just accept it.
The fraud patterns the system caught most reliably were also the most consistent across the dataset: high following counts paired with almost no posts, short or randomly structured usernames, blank or minimal bios, engagement rates that don't match follower volume. The rule-based scoring layer kept these signals visible rather than collapsing them into a single score, which made the results easier to explain to someone who wasn't involved in building the system.
Scalability wasn't an issue. Response times held steady as profile volume increased, feature extraction and behavioral scoring added minimal overhead, and the ensemble classifier scaled close to linearly. Nothing in the experiments pointed to a performance ceiling that would make larger deployments impractical.
The broader finding is that no single technique — supervised learning, behavioral scoring, or live data extraction — does this job well on its own. The combination is what makes the system work. It generalizes, it explains itself, and it scales. For real-world moderation use, that combination matters more than raw accuracy on a fixed test set.

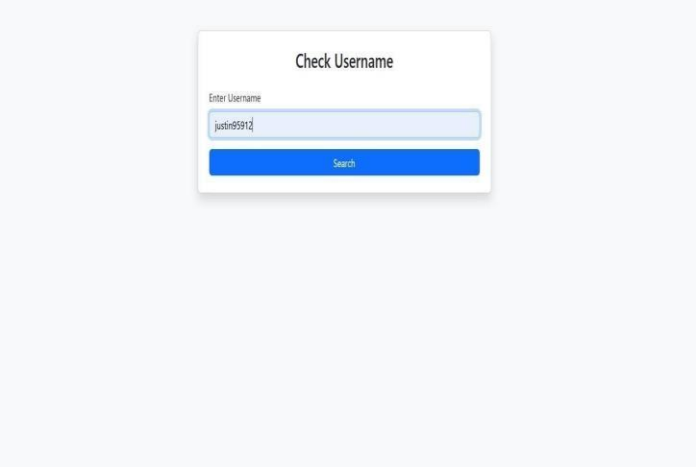

*Fig : This interface represents the **Username Verification Module***

The interface gives users one thing to do: type a username and hit Search. No configuration, no options to set — just an input field and a button.From there the system handles everything. It checks the input format, looks for the account in the internal dataset, and if it's not there, pulls live profile attributes automatically. Those feed into the preprocessing pipeline, then through feature extraction, then into the hybrid ML and behavioral scoring engine.The stripped-down design means anyone can use it. There's no learning curve and no assumption that the user understands what's running underneath. A researcher and a non-technical moderator can both get results from the same screen without the interface getting in the way.

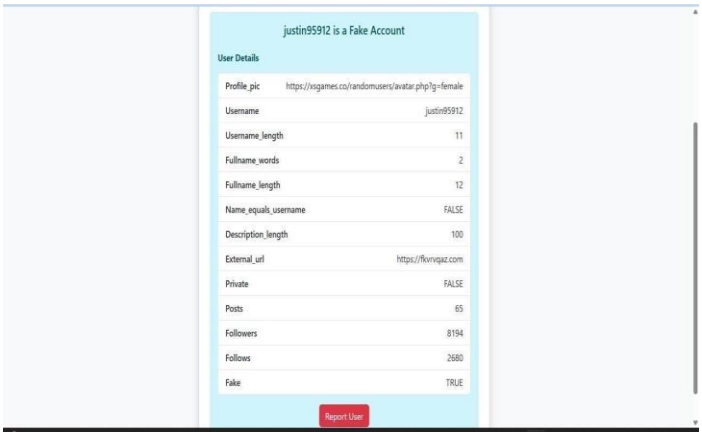

*Fig : This screen illustrates the **Result Display and Reporting Module** of the fake account detection system*

The result screen leads with the verdict. Fake or real — it's at the top, stated plainly, before anything else.

Underneath, a User Details table lays out everything the system examined: profile picture source, username characteristics, name length, bio size, external link presence, privacy status, post count, follower and following figures. The point isn't just to display data — it's to show what the model was working with. A reviewer can look at the follower-following gap or the description length and understand why the system reached the conclusion it did, rather than taking the label on faith.

The Report User button at the bottom closes a loop that purely automated systems tend to leave open. If something looks off that the model missed, or if a result needs a second opinion, users can escalate it for admin review. That feedback doesn't disappear — it feeds back into the pipeline.

By the time a user reaches this screen, the backend has already done its work. What's left is making that work legible — clear enough to act on, detailed enough to audit.

V. CONCLUSION

Fake Instagram accounts have gotten harder to catch. Automated tools now generate profiles convincing enough to pass basic filters, and the tactics keep shifting. A follower count threshold or a posting frequency check doesn't cut it anymore — attackers know exactly which numbers to adjust.

This project takes a different approach. Instead of relying on any single metric, the framework combines machine learning, engineered behavioral features, and live metadata extraction into one pipeline. Statistical features, activity patterns, username structure, and profile completeness all feed into the same evaluation. Random Forest handles the classification because it learns nonlinear relationships across many attributes simultaneously — the kind of patterns that simpler models tend to miss. The behavioral scoring layer sits alongside it, adding transparency and resolving cases where the classifier lands somewhere uncertain. Instaloader removes the static dataset constraint entirely, making live analysis of any public Instagram account possible.

The system doesn't require technical expertise to use, which matters for practical adoption across different audiences — students, researchers, cybersecurity professionals, moderation teams. There's also room to build on it: NLP-based bio analysis, network graph modeling, and more sophisticated deep learning methods are all natural next steps. What this version establishes is that the combination works. Each component covers something the others miss, and together they produce something more reliable than any single technique on its own.

REFERENCES

[1] Fake Profile Detection in Social Networks by A. K. Singh and S. K. Singh

[2] Detection of Fake Twitter accounts with Machine Learning Algorithms by Ilhan Aydin, Mehmet sevi and Mehmet Umut salur

[3] Detecting Fake Profiles in Online Social Networks by N. S. B. M. B. H. Kumar, V. Chinnakotla

[4] Fake News and Misinformation Detection on Social Media: Data Mining Perspective by A. S. G. S. V. Gupta

[5] Narayana Rao Appini, V. Bhuvana Kumar, "Phishing URL Detection with Gradient Boosting Classifier", Communications on Applied Nonlinear Analysis, Vol. 32 No. 3 (2025)

[6] Ömer Kasim, "Automatic Detection of Phishing Pages with Event-Based Request Processing, Deep-Hybrid Feature Extraction and Light Gradient Boosted Machine Model", Telecommunication Systems, Springer (2021)

[7] J.O. Ajayi , A.O. Adetunmbi, "Phishing Detection: Performance Evaluation of Both Ensemble and Classical Machine Learning Models", International Journal of Information Security, Privacy and Digital Forensics.

[8] K.N.S.B.V. Manjushal, Dr. D. Jaya Kumari2, “Detecting Phishing Links Analysis Using Machine Learning” 2024, IJFMR

[9] A. Alswailem, B. Alabdullah, N. Alrumayh and A. Alsedrani, "Detecting Phishing Websites Using Machine Learning," 2019 2nd International Conference on Computer Applications & (ICCAIS), 2019, pp. Information .

[10] J. Rashid, T. Mahmood, M. W. Nisar and T. Nazir, "Phishing Detection Using Machine Learning Technique," 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), 2020, pp. 43-46.

[11] M. H. Alkawaz, S. J. Steven and A. I. Hajamydeen, "Detecting Phishing Websites Using Machine Learning," 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), 2020, pp. 111-114.

[12] Thomas Nagunwa, “Comparative Analysis of Nature-Inspired Metaheuristic Techniques for Optimizing Phishing Website Detection”, MDPI, 2024

[13] D Shanthi , Smart Healthcare for Pregnant Women in Rural Areas, Medical Imaging and Health Informatics, Wiley Publishers,ch-17, pg.no:317-334, 2022

[14] D Shanthi, N Swapna, Ajmeera Kiran and A Anoosha, "Ensemble Approach Of GPACOTPSOAnd SNN For Predicting Software Reliability", International Journal Of Engineering Systems Modelling And Simulation, 2022

[15] Shanthi, D., Aryan, S. R., Harshitha, K., & Malgireddy, S. (2023, December). Smart Helmet. In International Conference on Advances in Computational Intelligence (pp. 1-17). Cham: Springer Nature Switzerland.

[16] Geetha, Mrs. D., Mrs.G. Haritha, B. Pavani, Ch. Srivalli, P. Chervitha, and Syed. Ishrath. 2025. “Eco Earn: E-Waste Facility Locator”. Metallurgical and Materials Engineering, May, 767-73. https://metall-mater eng.com/index.php/home/article/view/1632.

[17] D. Shanthi DS, G. Ashok GA, Vennela B, Reddy KH, P. Deekshitha PD, Nandini UBSB. Web-Based Video Analysis and Visualization of Magnetic Resonance Imaging Reports for Enhanced Patient Understanding. J Neonatal Surg [Internet]. 2025May13 [cited 2025May17];14(23S):280-5. Available from: https://www.jneonatalsurg.com/index.php/jns/article/view/5733

[18] Srilatha, Mrs. A., R. Usha Rani, Reethu Yadav, Ruchitha Reddy, Laxmi Sathwika, and N. Bhargav Krishna. 2025. “Learn Rights: A Gamified Ai-Powered Platform For Legal Literacy And Children’s Rights Awareness In India”. Metallurgical and Materials Engineering, May, 592-98. https://metall-mater eng.com/index.php/home/article/view/1611.

[19] P. Shilpasri PS, C.Mounika C, Akella P, N.Shreya N, Nandini M, Yadav PK. Rescuenet: An Integrated Emergency Coordination And Alert System. J Neonatal Surg [Internet]. 2025May13 [cited 2025May17];14(23S):286-91. Available from: https://www.jneonatalsurg.com/index.php/jns/article/view/5738

[20] Shanthi, Dr. D., G. Ashok, Chitrika Biswal, Sangem Udharika, Sri Varshini, and Gopireddi Sindhu. 2025. “Ai-Driven Adaptive It Training: A Personalized Learning Framework For Enhanced Knowledge Retention And Engagement”. Metallurgical and Materials Engineering, May, 136-45. https://metall-mater eng.com/index.php/home/article/view/1567.