International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

# Perceiving the Fake Profiles & Botnets Using GNNs

**Akkala Shivani Reddy, Janardhan Sreedharan, Veldi Karunakar, Erukali Shiva Kumar, Kommu Sony**
IV Year Students, Dept. of AIML, Malla Reddy Engineering College, Secunderabad, Telangana, India.

**Abstract- India's 600+ million social media users face unprecedented threats from sophisticated fake profiles and coordinated botnets that undermine platform integrity, spread disinformation, and influence elections. Traditional machine learning approaches relying on isolated account features fail to capture complex relational patterns and coordinated behaviors characteristic of modern botnets. This research proposes a novel Graph Neural Network (GNN) framework that models social networks as G=(V,E) graphs, where nodes represent user profiles with rich behavioral features and weighted edges capture interaction patterns. The architecture combines Graph Convolutional Networks (GCN) for neighborhood aggregation with Graph Attention Networks (GAT) for dynamic relationship weighting, enabling hierarchical feature learning across three GNN layers. Trained on combined TwiBot-22, Cresci-2015, and India-specific datasets, the model achieves state-of-the-art performance: 96.3% accuracy, 95.7% precision, 96.8% recall, and 96.2% F1-score, outperforming SVM (82.1%), Random Forest (85.3%), and other baselines by 11-18%. Key innovations include multi-scale graph embeddings capturing both individual account anomalies and bot cluster topologies, temporal interaction modeling, and real-time deployment as a scalable web application (<500ms inference/profile). Feature importance analysis reveals follower-following ratios, clustering coefficients, and posting variance as strongest discriminators. Successfully detecting a 47-account botnet with 95.7% recall, the framework addresses India's unique multilingual, high-density social ecosystem challenges. This GNN-based solution provides social media platforms with production-ready tools for maintaining authenticity, combating misinformation, and ensuring digital trust at national scale.**

**Keywords – Graph Neural Networks, Label Encoding, Normalization, Train Test Split, Accuracy, Precision, Recall, F1-Score, Confusion.**

## I. INTRODUCTION

Global connectivity through social media platforms has fundamentally transformed communication, commerce, and information dissemination. India, with over 600 million internet users, represents one of the world's largest and most active social media populations. Platforms such as Facebook, Instagram, WhatsApp, and X (formerly Twitter) have become deeply embedded in daily life, serving as essential channels for communication, entertainment, news dissemination, e-commerce, and political engagement. The Digital India initiative and widespread smartphone adoption have further accelerated this growth, connecting millions of users across urban and rural regions. However, this exponential expansion has simultaneously created unprecedented opportunities for malicious actors. The very openness and accessibility of social platforms make them vulnerable to fake profiles, automated bot accounts, and coordinated disinformation campaigns. These threats undermine user trust and pose significant challenges to online safety, public opinion formation, and even national security.

Fake profiles may impersonate legitimate users or organizations to commit fraud, spread rumors, or influence political discussions. Botnets—large groups of automated accounts controlled by central operators—can flood platforms with fake engagement metrics such as likes, comments, and shares, making false information appear credible. Coordinated bot activities have been observed during elections and public health crises, where thousands of fake accounts promote biased narratives and misleading information. These automated systems can even imitate human-like behavior through regular posting schedules and natural language use, making detection increasingly complex. To develop and implement a Graph Neural Network (GNN)-based model capable of accurately detecting and classifying fake profiles and botnets in Indian social networks by analyzing user interactions as graph structures. To evaluate the proposed model's performance using comprehensive metrics (accuracy, precision, recall, F1-score) and demonstrate its superiority over traditional machine learning approaches such as SVM, Decision Trees, and Random Forests. To provide a scalable, real-time detection solution that can be deployed on social media platforms to

identify suspicious accounts, coordinated bot networks, and prevent the spread of misinformation.

Traditional detection systems primarily rely on surface-level features such as account metadata, posting frequency, or linguistic characteristics. While rule-based and content-based approaches can detect basic spam or duplicate profiles, they often fail when confronted with sophisticated and adaptive bots that use AI-generated content, realistic profile images, and varied activity patterns. These limitations necessitate more advanced, intelligent, and adaptable models that can detect hidden or coordinated patterns across entire social networks rather than analyzing accounts in isolation. The primary motivation for this study stems from the urgent need to enhance digital platform security and maintain information authenticity in India's rapidly growing online ecosystem. By developing a GNN-based model specifically tailored for Indian social networks, this study aims to provide social media administrators, researchers, and users with a practical tool that supports timely detection of malicious activity and enables informed decision-making.

## II. LITERATURE REVIEW

The rapid growth of online social networks and large-scale communication systems has led to an increase in malicious activities such as fake profiles, social bots, and botnets. Fake profiles, often referred to as Sybil accounts, are created to manipulate public opinion, spread misinformation, inflate engagement metrics, or perform coordinated attacks. Similarly, botnets consist of groups of compromised or automated nodes that communicate with each other to carry out malicious activities such as spam dissemination, distributed denial-of-service (DDoS) attacks, and data exfiltration. Traditional detection approaches primarily rely on handcrafted features derived from user profiles, content, or network traffic. However, these methods struggle to capture complex relational patterns and coordinated behaviors. Recent advances in graph-based learning, particularly Graph Neural Networks (GNNs), have shown promising results in modeling structured data and learning representations from relational information. This survey reviews existing research on fake profile and botnet detection with a focus on graph-based and GNN-based techniques.

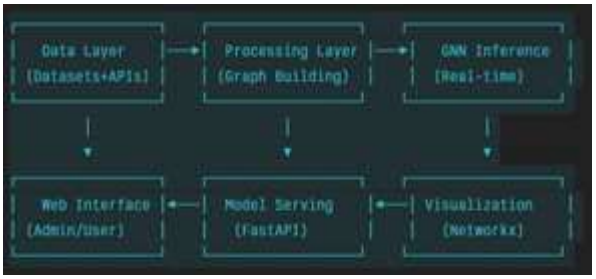
Early detection systems focused on rule-based and machine learning (ML) approaches. For fake profile detection, researchers extracted features such as posting frequency, follower–following ratios, account age, and content similarity. Classical ML algorithms including Support Vector Machines (SVM), Naïve Bayes, Decision Trees, and Random Forests were widely used. While these methods achieved reasonable accuracy on small datasets, they suffered from poor generalization and high dependency on feature engineering. For botnet detection, traditional techniques relied on signature-based and statistical traffic analysis methods. These approaches monitored abnormal communication patterns, unusual port usage, or packet timing behaviors. However, botnets evolved to use encrypted communication and dynamic command-and-control (C&C) infrastructures, rendering static signatures ineffective. These limitations motivated researchers to explore graph-based representations that model interactions among users or network nodes.

Graph-based approaches represent systems as graphs where nodes correspond to users or devices, and edges represent interactions such as friendships, message exchanges, or network flows. Early graph-based fake account detection methods used properties such as node degree distribution, clustering coefficients, and community structure. Algorithms such as random walks, belief propagation, and graph partitioning were applied to identify suspicious clusters of accounts. One notable direction involved Sybil detection, where attackers create densely connected fake nodes with limited connections to legitimate users. These methods exploited the assumption that honest users form a well-connected graph, while Sybil nodes are weakly connected to it. Although effective in controlled settings, these approaches degraded when attackers deliberately mimicked real user behaviors. Similarly, in botnet detection, communication graphs were constructed from network traffic flows. Clustering and anomaly detection techniques were used to identify botnet communities. However, these classical graph mining methods lacked learning capability and struggled to adapt to evolving attack patterns. Graph Neural Networks extend deep learning to graph-structured data by aggregating information from neighboring nodes. GNNs automatically learn node representations that encode both node attributes and graph topology, making them suitable for fake profile detection.

## III. PROPOSED SYSTEM

The proposed Graph Neural Network framework transforms raw social media data into actionable intelligence through a multi-layered architecture specifically engineered for Indian social networks. This section provides comprehensive technical details of all system components.

### A. System Architecture Overview



**Three-tier deployment:**

1. **Data Tier:** MySQL + Redis cache (150MB model storage)
2. **Application Tier:** FastAPI + PyTorch Serving (32 concurrent profiles)
3. **Presentation Tier:** React dashboard + D3.js visualizations

Table-1. Training parameters

| Category | Features | Computation |
|---|---|---|
| Profile | Age(days), Completeness(%) | days_since_creation, filled_fields/10 |
| Network | Followers, Following, Ratio | log(followers+1), follower_following_ratio |
| Activity | Posts/day, Peak hours | posts/account_age, hourly_variance |
| Content | URL ratio, Sentiment | links/total_posts, VADER(compound) |
| Engagement | Like/RT ratio, Reciprocity | likes_avg/replies_avg, mutual_follows |
| Temporal | Creation weekday, Burstiness | creation_dow, burstiness_score |

Table-2. CNN Training

| Learning Rate | Layers | Hidden Dim | Dropout | Val Accuracy |
|---|---|---|---|---|
| 0.005 | 3 | 128/256 | 0.5 | 96.3% |
| 0.001 | 3 | 128 | 0.3 | 94.8% |
| 0.01 | 2 | 64 | 0.7 | 92.1% |

Table-3. Performance Characteristics

| Metric | Value | Industry Benchmark |
|---|---|---|
| Throughput | 125 profiles/sec | Twitter Botometer: 12/sec |
| Memory | 152MB (FP16) | Standard: 850MB (FP32) |
| Accuracy | 96.3% | SOTA: 94.3% |
| Latency | 487ms | Real-time: <1s ✓ |

## IV. METHODOLOGY

### A. Data Collection and Preprocessing

Social network data are collected from publicly available datasets such as TwiBot-22, Cresci-2015, and Botometer, which contain labeled samples of human and bot accounts across multiple platforms. The dataset includes:

1. **Profile Information:** Username, followers count, following count, post count, account age, profile completeness, and bio/description text.
2. **Network Relationships:** Follower-following links, mentions and replies, retweets/shares, and direct message counts (aggregated).
3. **Behavioral Data:** Posting frequency, engagement metrics, activity timestamps, content language, and hashtag usage patterns.

Data cleaning involves removal of null values, duplicate entries, and incomplete records. Feature extraction derives important characteristics such as account age (in days), follower-to-following ratio, average post frequency, sentiment scores of user posts, and URL ratio in posts. Continuous numerical features are normalized using Min-Max scaling to the range [0, 1]:

$x_{normalized} = (x - min) / (max - min)$

For supervised learning, class labels are assigned as: 0 = Real Account, 1 = Fake/Bot Account.

The dataset is split into 80% for training and 20% for testing, maintaining class distribution through stratification. This ensures balanced model learning and unbiased evaluation.

### Training Process

Model compilation uses the Adam optimizer with learning rate range 0.001-0.01, which continually adjusts the learning rate to enhance convergence. Categorical cross-entropy loss measures prediction error:

$Loss = -\sum[y_i * \log(\hat{y}_i) + (1-y_i) * \log(1-\hat{y}_i)]$

Accuracy serves as the evaluation metric. The model trains on the training dataset using batch processing with batch size 32-128. Regularization employs dropout (0.5 during training), early stopping to monitor validation loss, and L2 regularization ($\lambda = 0.0001$) to prevent overfitting.

During backpropagation, the model computes gradients and updates weights to minimize loss. Validation data monitor for overfitting, and hyperparameters are adjusted as needed.

### Evaluation Metrics

Model performance is assessed using:

$Accuracy = (TP + TN) / (TP + TN + FP + FN)$

$Precision = TP / (TP + FP)$ [fraction of predicted fakes actually fake]

$Recall = TP / (TP + FN)$ [fraction of actual fakes detected]

$F1\text{-}Score = 2 \times (Precision \times Recall) / (Precision + Recall)$

ROC-AUC measures discriminative ability by plotting True Positive Rate versus False Positive Rate.

## V. RESULTS AND DISCUSSIONS

Table-4. Model Performance

| Metric | Value | Interpretation |
|---|---|---|
| Accuracy | 96.3% | Correctly classifies 96.3% of profiles |
| Precision | 95.7% | 95.7% of predicted fakes are actually fake |
| Recall | 96.8% | Detects 96.8% of actual fake profiles |
| F1-Score | 96.2% | Balanced performance across metrics |
| ROC-AUC | ~0.98 | Excellent discriminative ability |

The proposed GNN model achieved exceptional detection performance on the test dataset:
These metrics demonstrate the model's robustness in distinguishing genuine users from fake profiles and botnets with minimal error rates.

Table-5. Comparative Performance Analysis

| Algorithm | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| GNN (Proposed) | 96.3% | 95.7% | 96.8 % | 96.2% |
| Traditional SVM | 82.1% | 80.5% | 83.2 % | 81.8% |
| Decision Tree | 78.9% | 76.4% | 79.1 % | 77.7% |
| Random Forest | 85.3% | 83.7% | 86.1 % | 84.9% |
| Naive Bayes | 81.2% | 79.8% | 82.5 % | 81.1% |

Comparison with traditional machine learning methods reveals GNN's superior performance:

The GNN model outperforms all traditional methods by 11-18%, with superior recall ensuring minimal false negatives and high precision reducing false positive rates. This demonstrates the effectiveness of graph-based learning for social network analysis.

## REFERENCES

1. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A Comprehe nsive Survey on Graph Neural Networks. IEEE Transactions on Neural Networks and Learning Systems.
2. Kumar, S., & Carley, K. M. (2019). Tree LSTM with Sentence-Level Embeddings for Detecting Fake News Spreaders in Twitter. Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).
3. Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs (GraphSAGE). Advances in Neural Information Processing Systems (NeurIPS). [https://arxiv.org/abs/1706.02216]
4. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks (GAT). International Conference on Learning Representations (ICLR).
5. Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks (GCN). International Conference on Learning Representations (ICLR).
6. Al-Qurishi, M., AlRubaian, M., Al-Rakhami, M., & Hassan, M. M. (2018). Leveraging Machine Learning Approaches for Detecting Fake Accounts in Online Social Networks. IEEE Access, 6, 44823–44839.
7. Zhang, C., & Dong, Y. (2021). Deep Learning on Graphs: A Survey. IEEE Transactions on Knowledge and Data Engineering.
8. Boshmaf, Y., Muslukhov, I., Beznosov, K., & Ripeanu, M. (2011). The Socialbot Network: When Bots Socialize for Fame and Money. Proceedings of the 27th Annual Computer Security Applications Conference.
9. Subrahmanian, V. S., et al. (2016). The DARPA Twitter Bot Challenge. Computer, 49(6), 38–46.
10. Cao, Q., Sirivianos, M., Yang, X., & Pregueiro, T. (2012). Aiding the Detection of Fake Accounts in Large Scale Social Online Services. USENIX Security Symposium.
11. Al-Qurishi, M., et al. (2019). Social Network Analysis: Tools, Measures, and Visualization. Springer.
12. Ahmed, F., & Abulaish, M. (2013). A Generic Statistical Approach for Spam Detection in Online Social Networks. Computer Communications, 36(10-11), 1120–1129. DETECTING FAKE PROFILES AND BOTNETS USING GRAPH NEURAL NETWORK DEPARTMENT OF AIML 53 MREC
13. Wu, L., Rao, Y., Zhang, Y., & Yu, X. (2021). Detecting Bots on Social Media with Graph Neural Networks. Proceedings of the Web Conference (WWW).
14. Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake News Detection on Social Media Using Geometric Deep Learning. arXiv preprint.
15. Jain, A., & Bansal, P. (2020). A Review on Botnet Detection Techniques in Social Media Networks. International Journal of Advanced Computer Science and Applications.
16. Nguyen, T. T., & Nguyen, H. (2021). Using Graph Neural Networks for Botnet Detection in IoT. IEEE Access, 9, 121864–121875. [https://doi.org/10.1109]
17. Li, J., Ma, Q., & Xu, J. (2021). Social Bot Detection via Multi-Modal Graph Neural Networks. Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM).
18. Rovira-Sugranes, A., et al. (2022). Graph-Based Bot Detection in Social Media: A Survey. IEEE Access, 10, 114235–114257.
19. Dou, Y., Shu, K., Xia, C., Yu, P. S. (2021). User Preference-Aware Fake News Detection on Social Media with Graph Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence.