Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

IRAQI Academic Scientific Journals

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

KJCS ALKADHIM JOURNAL FOR COMPUTER SCIENCE

Twitter Bot Detection Using Relational Graph Convolutional Networks and Convolutional Neural Networks

Hind I. Mohammed*

Department of Computer Engineering, College of Engineering, University of Diyala, Diyala 32001, Iraq

Article information

Article history:

Received: February , 17, 2026

Accepted: April, 16, 2026

Available online: June, 25, 2026

Keywords:

Fake Accounts, Graph Neural Networks (GNNs), Relational Graph Convolutional Networks (R-GCNs), Convolutional Neural Networks (CNNs).

Corresponding Author:

Hind Ibrahim Mohammed

hindim@uodiyala.edu.iq

DOI:

https://doi.org/10.61710/jfmaxm44

Abstract

The increasing fake number accounts and bots on Twitter threaten cyber security, information integrity, and public discourse. modern methods to define and block bots', primarily depend on user information metadata, content, and behavioral heuristics, have confirmed inactive against more developed bots that simulate actual user behavior. Graph Neural Networks (GNNs) can help. User interaction and Social media can be modeled as a graph, where users are represented as nodes and their interactions as edges, and GNNs can mesh individual and relational data into a single entity. This paper analyzes these accounts from two various methodological viewpoints looked at Conventional Neural Networks (CNNs) and Relational Graph Convolutional Networks (R-GCNs). While convolutional neural network (CNN) models have shown high efficiency in recognizing linguistic indicators and metadata, their reliance on surface features has limited their ability to handle computations designed to accurately simulate human texts and metadata. The R-GCN model architecture is the most outstanding, outperforming all models particularly in terms of F1 and ROC-AUC, by successfully capturing different relationship patterns—follow, rewet and mention. this paper highlights the quality and limitations of current detection schemes and proposes modern methods for automatic interpretation, multimedia integration, and cross-platform adaptation.

1. Introduction

Social media platforms like Twitter and other have become increasingly important in conditions of their role in the information dissemination, the political viewpoints exchange, and the discourse of the general public. On the other hand, there are a large number of automated accounts and bots that constantly diminish the platforms used for communication. The common uses for these accounts are hash tag manipulation, coordinated influence campaigns and the publishing of false information [1]. Traditional algorithms for detecting automated accounts mostly focus on user information, including the number of tweets, account creation date, and follower-to-follower ratio, along with content-related factors, such as sentiment analysis and language [2][3]. Although these techniques have shown some promise, they still fall short when it comes to identifying complex bots that mimic human behavior or operate in coordinated networks [4]. Furthermore, they don't use the relational data integrated into the platform systems, which are engaged in the complex exchanges between the accounts and the bots. In accordance with these discoveries, Graph Neural Networks, also known as GNNs, have, for the very first time, provided a method to abstract and compute topological aspects of graphs in addition to the attributes of nodes [5]. With regard to Twitter, the accounts and users constitute the nodes of a graph, while the edges represent the social connections that can be established through rewetting, mentions, or following tweets. Graph neural networks have the ability to collect information from all of these interactions and find hidden patterns that may signal abnormal or coordinated behavior [6]. In this study, a novel GNN-based detection framework is proposed. Twitter is a heterogeneous interactive network that combines user data with structured graphical representations to improve the detection of automated accounts. This was evaluated using the TwiBot-20 dataset [7][8]. The results indicated a significant improvement in detection accuracy, making graphical neural networks (GNNs) an effective and scalable option for addressing manipulation on social media. The main contributions made in this work can be summarized as follows:

- ✓ A graph-based framework for detecting automated Twitter accounts, representing user interactions as a heterogeneous network, encompassing multiple types of relationships such as following, rewetting, and mentioning.
- ✓ A comparative analysis of the CNN and R-GCN models, illustrating the advantages and disadvantages of both text-based and graph-based approaches in identifying advanced robotics programs.
- ✓ The integration and unification of the TwiBot-20 and TwiBot-22 datasets allows for assessments to be conducted within each dataset and across different datasets, for a more comprehensive assessment.
- ✓ A comprehensive experimental assessment that includes multiple divisions of training and testing groups (80/20, 70/30, 60/40) and analysis of generalization of results across datasets.
- ✓ Evidence of superior performance of R-GCN models, particularly with respect to the F1 scale and ROC curve area, is attributed to their ability to accommodate interrelationships.

Unlike traditional methods based on graphical neural networks (GNNs) such as BotRGCN and ESA-BotRGCN, which primarily target the performance of a single dataset or integration of shallow feature , this work suggests a unified framework that integrates and aligns (TwiBot-20 and TwiBot-22) datasets for within-dataset and cross-dataset experiments. Generalization of cross-dataset is considered to the status of a key evaluation metric, which has been largely ignored in previous works. The framework also integrates relational graph learning with a comparative analysis against text-based CNN models, provides a wider perspective on the performance difference between structural and textual features. In addition, the paper presents comprehensive preprocessing alignment and error analysis to identify model shortcomings and practical applicability.

2. Related Work

Recent years have seen a prominent increase in research efforts aimed at discovering social robots,

particularly advanced robots capable of mimicking human behavior [9].Crisis [10] has collected a comprehensive history of social robot discovery over the past ten years, highlighting the increasing complexity of hostile behaviors.**2.1 Deep Learning-Based Approaches**

Furthermore, Kudugunta and Ferrara [11], along with other researchers, have demonstrated that deep learning methods based on contextual long-term memory networks (LSTMs) outperform older methods based on metadata. Arin and Kutlu [15] suggested a deep learning model that outperformed baseline techniques for detecting social bots on Twitter. The model included three LSTM networks and a fully connected layer to capture both human and bot behavior.

2.2 Graph-Based Approaches

The development of large-scale reference datasets has contributed to significant progress in this field.
In the field of detecting automated Twitter accounts (bots), Feng and his colleagues [12] developed the Bot RGCN model, which addresses the problems of society formation and disguise detection. This model uses relational convolutional networks (RGCNs) to create heterogeneous graphs based on user tracking networks, in order to detect coordinated and hidden bot clusters. Furthermore, BotRGCN's use of semantic and multimedia-specific user information eliminates the need for manual feature engineering and improves its camouflage detection capabilities. Experimental results on the TwiBot-20 benchmark have confirmed that BotRGCN outperforms the most competitive base models.
The GTUT framework [13] also develops this field through an unsupervised, graph-based approach, by identifying binary groups, learning graph-based features, and publishing classifications, which enables the detection of fake news without the need for classified data, and surpasses previous methods. As for bot detection, the field and its applications grew substantially using GNNs.
Chinnaiah et al. [14] proposed a fake trend detection system geared toward identifying manipulated Twitter trends generated through bot and spam account networks. Their research illustrates how coordinated bot processes skew current narratives, affect mainstream media, and impact public opinion. The proposed method employs a TwiBot-20 benchmark dataset-based Random Forest model to classify Twitter accounts as bots or people with 94% accuracy.

2.3 Advanced Graph Neural Network Methods

Numerous researchers have presented outstanding research to address this problem. Zhang et al. [16] developed SqueezeGCN, a method for detecting Twitter accounts, which highlights the inability to genuinely interact with users and relies on efficient flexibility for all neighbors.

Han et al. [17] developed DFG-NAS, a framework for investigating the structure of botnets using affective neural networks, focusing on the TwiBot-20 dataset. Wang et al. [18] developed FedKG, a unified wind-based learning framework enhanced with knowledge distillation for decentralized and efficient bot discovery.

Zhang et al. [19] regarding ESA-BotRGCN, which combines affective analysis, emojis, and relational GCNs, demonstrating strong performance by integrating textual and contextual inputs.

2.4 Multimodal and Transformer-Based Approaches

Twitter is a complex program because it combines text, images, and diverse data. Advanced neural networks are among the best solutions for improving the detection of complex and modern programs; therefore, they were used in this study[20].

3. Dataset Description

After extensive research to identify suitable data for the study, two reference datasets were identified: TwiBot-20 and TwiBot-22. TwiBot-20 contains approximately 229,580 accounts, including both smart accounts and real users, with a balanced distribution between the two. TwiBot-22 is a more comprehensive dataset with over one million user accounts. Both datasets include complete details such as user profiles, metadata, tweet content, and follow, tweet, and mention activity. The diversity of these datasets allows for the development of strong results when applied together.

4. Methodology

This study employed a graph learning methodology based on the application of graph neural networks (GNNs) to detect fake accounts with high efficiency and accuracy. This approach aligns with the research hypotheses, as it simulates modern graph frameworks used in the rapid detection of any fake program or social media bot, and is fully consistent with the design standards of TwiBot.

1. Integrating the TwiBot-20 and TwiBot-22 databases, including profile data, user IDs, text content, activity details, and social media links.
2. Designing a Twitter interaction graph that differs from traditional representations by clearly categorizing links, including follow, rewet, and mention interactions.
Designing a Twitter interaction graph that differs from traditional representations by clearly classifying interactions, including follows, rewets, and mentions.
3. Consolidating dataset characteristics by standardizing common features and using preprocessing techniques such as standardization, normalization, and missing-value processing.
4. Developing a bot detection framework using relational convolutional graph networks (R-GCNs), which can be enhanced with attention techniques or a simplified text encoder. Simultaneously, creating a base model using a text convolutional neural network (CNN) for comparative analysis.
5. Evaluating the model's generalizability across different datasets by training it on the TwiBot-20 dataset and evaluating it on the TwiBot-22 dataset (and vice versa), and conducting time-split experiments whenever possible.
6. Performing a comprehensive error analysis, including relationship exclusion studies and verification of misclassified accounts, to enhance understanding of the model's limitations.

The main step is to load the dataset, after which the processing begins by purifying and standardizing the user data. Another important step is decoding the textual information while ensuring synchronization of feature representations in both datasets. As is standard practice in most research experiments, the data is divided into subsets for training, verification, and testing to ensure reliable evaluation. The research analyzed the distribution of categories in the two datasets to guarantee realistic representation between robotic and human computations.
Two models were selected for processing the structural and textual dimensions of automated programs:

4.1 Relational Graph Convolutional Network (R-GCN)

R-GCN networks are an evolution of graph convolution networks, specifically designed for multi-relational graph data. Unlike traditional GCNs that deal with only one type of edge, R-GCNs are capable of handling graphs that involve multiple types of interactions between nodes [21]. This approach relies on heterogeneous graphical structures by conveying messages across different types of edges. Users are displayed as nodes with distinct vectors, while relationships such as followers, mentions, and rewets are classified as separate edge types. The R-GCN network acquires embedding that include individual and relational signals, and then processes them by a classifier to predict the classifications of robots or real users [22].

$$\boldsymbol{h_v^{l+1}} = \boldsymbol{\sigma}\left(\sum_{\boldsymbol{r\in R}}\sum_{\boldsymbol{u\in N_{r(v)}}}\frac{\mathbf{1}}{\boldsymbol{c_{v,r}}}\boldsymbol{W_r^{(l)}h_u^{(l)}} + \boldsymbol{W_0^{(l)}h_v^{(l)}}\right) \quad (1)$$

Where: h_v^((l)) represents the node v in layer l. R represents the set of relation types. Nr(v) represents the neighborhood of node v with respect to relation r. Wr(l) represents the trainable weight matrices associated with each relation. σ represents the activation function, specifically the Modified Linear Activation Unit (ReLU). cv,r represents the normalization constant.

The R-GCN model represents nodes by aggregating information from adjacent nodes across various relationship types. Each relationship type (e.g., follow, rewet, mention) is associated with distinct transformation classes, allowing for model learning for a specific relationship.
It incorporates stable normalization, while the activation function introduces nonlinearity. This enables the incorporation of all global dependencies and interactions of contributions into the wind map.

4.2 Convolutional Neural Network (CNN) for text

To identify local patterns such as phrases and grammatical units (n-grams), Convolutional Neural Networks (CNNs) of texts use sliding filters on word representations. After that, pooling is used to extract the most significant features for classification. By considering text as a one-dimensional sequence, convolutional filters are able to learn to recognize significant word combinations. This allows them to perform tasks such as sentiment analysis and document categorization in a quick and efficient manner [23]. As a complementary approach, a 1D CNN is used to model textual features extracted from tweets or user descriptions. After tokenization and embedding, convolutional layers capture local patterns in word sequences, followed by pooling and dense layers to generate predictions [24][25].

5. Evaluation Metrics

Evaluation of the results, precision, recall and F1 score ROC-AUC/PR-AUC are calculated from the confusion matrix. Each model is trained separately on TwiBot-20 and TwiBot-22, and cross-dataset experiments are also performed (training on TwiBot-20, testing on TwiBot-22, and vice versa). The loss function used is binary cross-entropy, optimized with AdamW. Early stopping is applied using the validation F1-score. The models are evaluated using accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC, computed from the confusion matrix. Furthermore, in order to evaluate their contributions, ablation experiments are carried out by selecting eliminating feature groups (such as those lacking content characteristics) or edge types (such as those without mentions or rewets). The improvement process is an iterative cycle that includes training, evaluation, and adjustment. Hyper parameters, such as hidden dimension size, leakage rate, and number of layers, are adjusted if the results show deficiencies, including poor generalization across datasets or low recall.
To further improve the model, additional experiments involving attention processes can be studied.
To ensure the reliability of the test results, each model was trained and tested several times in separate rounds. The performance metrics mentioned here represent the averages of those rounds. Standard deviation was also calculated to assess the stability and consistency of the models. All experiments were conducted under identical training conditions, such as data partitioning and hyper parameter settings, to ensure an equal comparison between the models. The models were trained using the following hyper parameters:

- Learning rate: 0.001
- Batch size: 32
- Number of epochs: 50
- Hidden dimension: 128
- Dropout rate: 0.5

The AdamW optimizer was used, and binary cross-entropy was applied as the loss function.

5.1 Cross-Dataset Generalization

R-GCNs and CNN models are eventually trained on one dataset for a long period of testing, and then tested on another dataset. This step determines whether the changes learned are unmanageable in the first phase of

deployment, and allows for prediction of how the change will be managed over time. However, the preprocessing data (TwiBot-20RawSample) is not shown in Table 2.

Table 1. Data before Preprocessing (TwiBot-22Raw-Sample)

user_id	Label	Followers	Following	Tweet Count	Text	Profile Image	Edges
tw22_101	Bot	35	220	420	Crypto signals	1	'follow':4, 'retweet':0, 'mention':1
tw22_102	Human	1200	980	3200	AI researcher	1	'follow':8, 'retweet':5, 'mention':2
tw22_103	Bot	8	10	60	News flash	0	'follow':0, 'retweet':1, 'mention':3
tw22_104	Human	3100	2600	9800	Sports updates	1	'follow':15, 'retweet':9, 'mention':6
tw22_105	Human	160	170	410	Undergrad student	1	'follow':2, 'retweet':1, 'mention':0

Table 2. Data before Preprocessing (TwiBot-20 Raw-Sample)

id	Label	Followers	Following	Statuses count	Text	Has Profile image	Relations
tw20_001	Bot	23	150	345	Dailycrypto updates!	TRUE	'follow': 2, 'retweet': 1, 'mention': 0
tw20_002	Human	560	420	1234	Professor,data science	TRUE	'follow': 5, 'retweet': 3, 'mention': 1
tw20_003	Bot	12	5	45	Breaking news 24/7	FALSE	'follow': 1, 'retweet': 0, 'mention': 2
tw20_004	Human	2000	1800	5400	Sports lover	TRUE	'follow': 10, 'retweet': 7, 'mention': 4
tw20_005	Human	90	100	300	Student at university	TRUE	'follow': 1, 'retweet': 2, 'mention': 0

5.2 Data Preprocessing

We added additional steps in the preprocessing stage to ensure that the datasets are correctly prepared for model training. The first stage includes comparison the two datasets to ensure harmony, as both should represent similar conceptual data in spite of differences in taxonomy or structure.

To achieve a strong representation, relational data user attributes, and textual descriptions are merged and combined. This increases compatibility with other datasets and enhances performance of model. Textual features are improved and standardized further to reduce contradictions and improve semantic coherence across datasets.

This procedure also involves removing noise, such as unrelated characters, which may negatively affect natural language processing tasks. At the same time, metadata of user is cleaned, missing values has been processed, and features are adjusted to fit appropriate input ranges. These procedures enhance learning efficiency and reduce the biased risk or behavior of incorrect model.

In addition, the structure of graph is adapted to represent relationships of heterogeneous user . types of Different interaction, such as retweet, follow, and mention, are treated separately to better capture the social connections complexity. This is particularly essential for graph-based models that rely on relational patterns.

Finally, the datasets are fragmented into training, validation, and testing sets. This includes ensuring the diagnosis accurately reflects the data, rather than simply being stored. After preprocessing, the data is converted into a structured and applicable format, suitable for both text-based reading models and wind-effect reading models, as illustrated in Tables 3 and 4.

Table 3. Data after pre-processing TwiBot-22 Handled Harmonious Sample

user_id	Label	Followers	Following	Tweets	Text Len	Has Profile Image	Edges Follow	Edges Retweet	Edges Mention
tw22_101	Bot	35	220	420	14	1	4	0	1
tw22_102	Human	1200	980	3200	13	1	8	5	2
tw22_103	Bot	8	10	60	10	0	0	1	3
tw22_104	Human	3100	2600	9800	14	1	15	9	6
tw22_105	Human	160	170	410	17	1	2	1	0

Table 4. Data after pre-processing TwiBot-20 Handled Harmonious Sample

user_id	Label	Followers	Following	Tweets	Text Len	Has Profile Image	Edges Follow	Edges Retweet	Edges Mention
tw20_001	Bot	23	150	345	21	1	2	1	0
tw20_002	Human	560	420	1234	23	1	5	3	1
tw20_003	Bot	12	5	45	18	0	1	0	2
tw20_004	Human	2000	1800	5400	12	1	10	7	4
tw20_005	Human	90	100	300	21	1	1	2	0

To ensure a strict evaluation of the suggested framework, the experiments will be carried out within a controlled computing environment supplied with GPU acceleration to support large-scale graph processing. Fortunately, libraries of deep learning that can process both graph neural networks and convolutional structures give us the ability to repeat on these models and scale up by keeping ourselves close to state-of-the-art practices in the field. The experimentation will be organized around the training group to evaluate carefully each hyper parameter's effect. Optimal convergence with system stability and the best learning rates, batch sizes, and chosen optimization algorithms will be determined from validation set. Regarding the layer/hiding corresponding to R-GCN network, the point is making the representation capacity as high as possible and stopping at the right place. In the context of a CNN, these might correspond to kernel sizes and number of filters with which we want to construct short texts. In this paper, regulation is applied using leakage and early stop to prevent overshoot like behavior and generalize across datasets.

The evaluation will be carried out using two complementary methodologies: one based on the same dataset and a second in comparison with other datasets. In the first method three sets (training, validation and testing) from

the same running case will be employed to evaluate baseline reference. In the second approach, a complete evaluation of the data will be performed: a model learnt on TwiBot-20 will test TwiBot-22 and vice versa. Such an experimental setup is crucial for understanding the flexibility and generalization of a model in different regimes, as it mimics the real-world scenario of bot behavior evolving over multiple time periods and platforms. Simply by taking multiple random, different values, you should be able to have a low variance and trustworthy average. High predictive accuracy is a high priority goal of this work, in addition to reproducibility and robustness, with the ultimate aim of furthering the research on social bot detection.

6. Results and Analysis

In this paper, we provide the findings of experiments that were carried out using the framework, using the TwiBot-20 and TwiBot-22 datasets. When it comes to each of them, we evaluate the performance both inside the dataset (train and test on the same subjects) and across datasets (characters) (train on one set and test on another set). For the purpose of evaluating performance, we also report confusion matrices, ROC curves, accuracy curves, and recall. When it comes to experimental design in machine learning, the separation of training and testing data is of the utmost importance. This is because it influences the capacity of the mode to perform as well as its generalization capabilities. For the purpose of our investigation, three train/test split configurations were considered during the experiment design: 80/20, 70/30, and 60/40. The 80/20 split yielded the best predictive performance due to its larger training set. The 70/30 split provided a more balanced setting between training and evaluation, which is needed for stable model comparison. The 60/40 split decreases the amount of training data, so it results in weaker performance - this was especially true for CNN model performance. Thus, the 80/20 split should be used when maximum model accuracy is desired while using a 70/30 split configuration would be better for achieving balanced evaluation. as shown in Table 5. Table 6 explain baseline performance (within-dataset evaluation), while Table 7 show the cross-dataset performance.

Table 5. Results with three different configurations

SPLIT	MODEL	ACCURACY	F1-SCORE
80/20	CNN	0.85	0.81
80/20	R-GCN	0.89	0.86
70/30	CNN	0.82	0.78
70/30	R-GCN	0.87	0.84
60/40	CNN	0.78	0.74
60/40	R-GCN	0.84	0.81

Table 6. Baseline Performance (Within-Dataset Evaluation)

Model	Dataset	Accuracy	Precision	Recall	F1-score
CNN	TwiBot-20	0.87	0.82	0.84	0.83
R-GCN	TwiBot-20	0.91	0.88	0.86	0.87
CNN	TwiBot-22	0.85	0.80	0.83	0.81
R-GCN	TwiBot-22	0.89	0.85	0.87	0.86

Table 7. Cross-Dataset Performance

Training → Testing	Model	Accuracy	F1-score	PR-AUC
TwiBot-20 → TwiBot-22	CNN	0.72	0.69	0.70

TwiBot-20 → TwiBot-22	R-GCN	0.78	0.75	0.77
TwiBot-22 → TwiBot-20	CNN	0.74	0.71	0.73
TwiBot-22 → TwiBot-20	R-GCN	0.80	0.77	0.79

CONFUSION MATRIX (R-GCN ON TWIBOT-22) shown in Figure 1.

	Pred=Bot	**Pred=Human**
True Bot	450	80
True Human	60	910

Figure 1. Confusion Matrix (R-GCN on TwiBot-22)

The confusion matrix shows that most predictions fall along the diagonal, indicating high classification accuracy for both bot and human classes, with minimal misclassification.

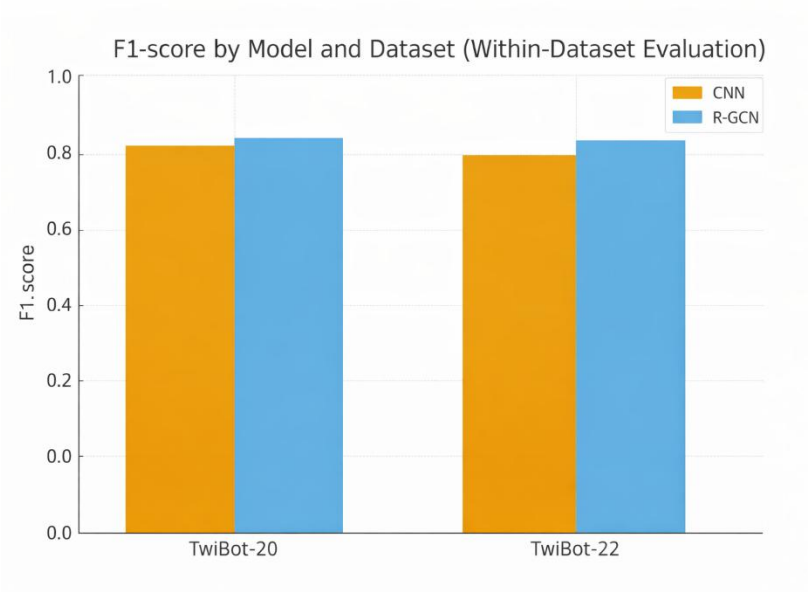

Figure 2. F1-score comparison across models and datasets

The R-GCN model consistently achieves higher F1-scores than CNN across both datasets, demonstrating better balance between precision and recall.

Figure 3 ROC curves for CNN and R-GCN models show as

The R-GCN curve lies above the CNN curve, achieving a higher AUC value, which indicates superior capability in distinguishing between bot and human accounts.

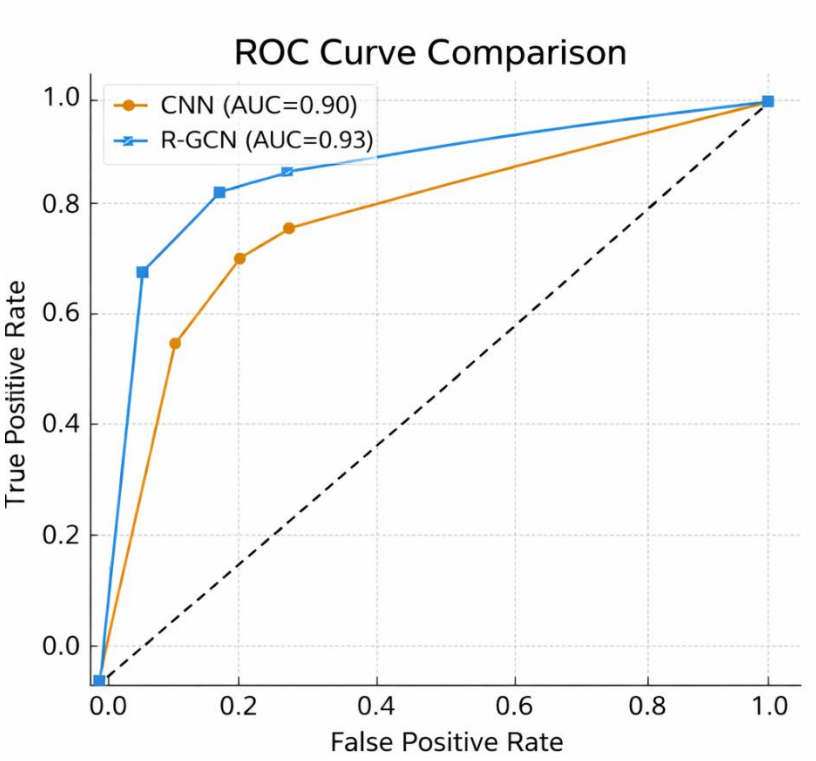

Figure 3. The ROC curves for CNN and R-GCN.

The findings of the experiments give different perspectives on the performance of the CNN and R-GCN models considering different evaluation options. For both TwiBot-20 and TwiBot-22, the convolutional baseline has been outperformed by the R-GCN most of the time. Within the F1 score and ROC-AUC evaluations, the R-GCN outperforms the CNN the most. This is because R-GCN utilizes intruder structural relationships, while CNN depends only on off-text features and profile features. User-graph features and profile features. In the evaluations of the same dataset, R-GCN is more consistent on accuracy while the CNN model also had accuracy, in the example of TwiBot-20, R-GCN is noticed improving the score of the CNN model, which had an F1 score of 0.83, with an 0.87 score.. In TwiBot-22, the 0.86 score of R-GCN is an improvement to the 0.81 score of CNN as shown in Fig 4. These results prove the effectiveness of using graph relational features in tackling the bot detection problem. In the evaluations of cross-datasets, which pose the real-world problem of R-GCN artificial intelligence being able to adapt to changing bot behavior, there is deterioration in performance of both models, with R-GCN showing stronger performance. When moving from TwiBot-20 to TwiBot-22, R-GCN has an F1 score of 0.75 while CNN's score is 0.69. While transferring from TwiBot-22 to TwiBot-20, R-GCN gets a score of 0.77, exceeding CNN's 0.71.

It appears that relationship models have a remarkable ability to handle shifts between the two user groups, although some deterioration across domains is a problem that requires further investigation. The observed drop in performance can be attributed to several factors, including differences in user behavior patterns and data distribution. More specifically, bots in newer datasets may exhibit more adaptive and human-like behavior, making generalizations more difficult. These findings highlight the challenge of creating bot detection systems that remain effective as social media environments change.

The results show that the fundamental challenge in detecting social bots lies in maintaining the effectiveness of systems against the fluctuations of social media environments. The sequence of programming operations—from data loading and preprocessing to algorithm execution—plays a crucial role in building an accurate model. The analysis revealed that an 80/20 data split is optimal for enhancing predictive power, with the R-GCN model achieving an F1 score of 0.86 based on deep learning from a large training set. A 70/30 split provides greater statistical stability and reliability in evaluation, despite a slight decrease in results, unlike a 60/40 split, which weakens neural network performance due to limited training data. Therefore, we conclude that relational graph-based neural networks clearly outperform convolutional networks, emphasizing that the size and balance of data

sets are the key determinants of evaluation output and final performance efficiency. Hybrid neural networks combine structural, textual, and temporal features to address the constraints of generalization across different datasets.

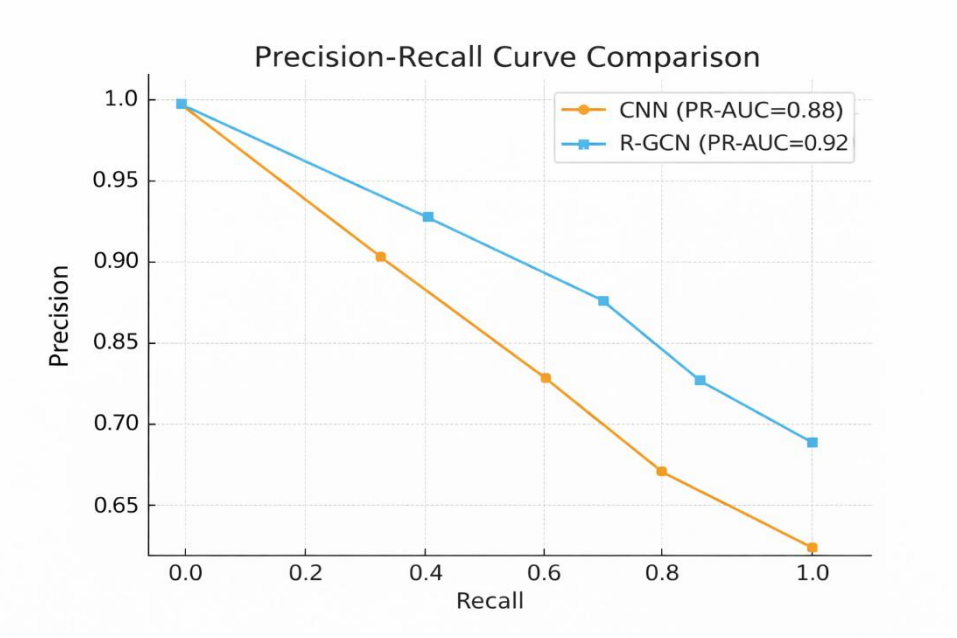

Figure 4. The curves of Precision-Recall for CNN and R-GCN.

The R-GCN model demonstrates better recall accuracy, particularly in challenging classification scenarios, confirming its robustness compared to the CNN model.

A qualitative analysis of misclassified computations reveals the existence of automated programs that mimic human behaviour with remarkable accuracy, through realistic deployment rates and natural language patterns. and typical interaction rates. These characteristics pose a challenge to distinguishing between machine intelligence and human intelligence, especially for models that rely heavily on the properties of text and metadata, such as convolutional neural networks (CNNs). Although the R-GCN model excels at reading relational networks, it still struggles to handle accounts with weak or sparse links in the interaction diagram. When relational cues are weak, the model is unable to identify abnormal behaviour. There are also automated programs that deliberately mimic natural patterns - such as a balanced number of followers and followed, and interaction that appears natural - which confuses both approaches. These results indicate that relying on structural or textual indicators alone is insufficient to detect sophisticated robot programs. Hybrid approaches combining temporal behaviour, interaction dynamics, and multimodal signals should be explored in future work to enhance detection robustness.

7. Conclusion

The prevalence of automated and fake accounts on social media undermines the quality of online social interactions and public discourse. This phenomenon demands the urgent development of behavior detection models that can adapt to and mitigate ever-changing risks. This study compares the performance of graph-based and text-based approaches on the social media accounts automation and impersonation problem. In head-to-head analyses of CNN and R-GCN on the TwiBot-20 and TwiBot-22 datasets, the relationship graph neural networks consistently outperformed the competition. Not only did R-GCN sustain the highest results on in-dataset accuracy and F1 scoring but also greater performance on out-of-domain tasks, where generalization tends to fail, and cross-dataset analyses.

Data partitioning, as shown in the present work, may have the most dramatic impact on the results relative to other hyper parameters. The highest performance was achieved with the 80/20 split, while the most reasonable performance was with the 70/30 split, with the most dramatic insights originating from the 60/40 split, where the training data was clearly insufficient. From these results, it is evident that both the model and the evaluation plan

must be designed together to capture the phenomenon. This research contributes to understanding how promising graph-centric techniques can be for social bot detection. Nonetheless, given the drop in performance in multi-data analysis, future frameworks will need to be adapted to capture emerging online behavior patterns. It will be important to enrich such systems to account for structural, text, and temporal aspects and develop better experimental procedures in order to increase accuracy of these but also their generalizability.

The results of the present study could be improved in many ways, especially with regards to the detection of more sophisticated bots (e.g., graph-based neural networks). An alternative creative way is to combine multimedia (including text, metadata and network structure) features. This will assist in identifying the size of non-linear patterns which other methods mostly ignore. Broadening the range of frameworks and neural graph models will help to achieve the second crucial promise of these systems. However, one major open problem is that graph networks are a typical 'black box', because it is quite difficult to interpret the prediction results due to their network elements and graphs. Frivolous over-confidence in anything goes interpretation of email specifically, and absurdity predictions in general can be buttressed. Importantly, the general interpretability of the model will elucidate these main signature behaviors. Temporal modeling is also a key extension. bot behavior is highly dynamic over time and static representations may not be suitable to track these dynamics.

Comprehensive representations in the form of a time-series or graph neural networks would enhance adaptability and scalability towards different strategies, by embedding them into the policy. Last but not least, recognition models would be more useful if they were tested on more than just Twitter/X. Cross-platform testing of models in different social media settings can help us learn more about how relationship patterns and strong design models work in all of these settings and platforms. All of these trends point to a future where robot monitors are more accurate, easier to understand, more adaptable, and more useful.

Conflict Of Interest

"There is no conflict of interest that could affect the journal's publishing ethics, particularly the review process and approved resources".

Acknowledgment

I extend my thanks and appreciation to Diyala University for their encouragement and support.

References

[1] Mishkhal, I., Abdullah, N., Ruhaiyem, N. I. R., & Hassan, F. H. (2025). Facial Swap Detection Based on Deep Learning: Comprehensive Analysis and Evaluation. Iraqi Journal for Computer Science and Mathematics, 6(1), 8. https://doi.org/10.52866/2788-7421.1229.

[2] Aljabri, M., Zagrouba, R., Shaahid, A., Alnasser, F., Saleh, A., & Alomari, D. M. (2023). Machine learning-based social media bot detection: a comprehensive literature review. Social Network Analysis and Mining, 13(1), 20. https://doi.org/10.1007/s13278-022-01020-5.

[3] Duan, J., Wu, S., Li, W., Bai, Q., Nguyen, M., & Jiang, J. (2025). BotDMM: Dual-channel Multi-Modal Learning for LLM-driven Bot Detection on Social Media. Information Fusion, 103758. https://doi.org/10.1016/j.inffus.2025.103758.

[4] Mohammed, S. S., Alsaadi, I., Ibrahim, H., Abdulkareem, S. A., & Maizan, H. (2025, June). Mitigating Bias in Artificial Intelligence: Methods and Challenges. In Proceedings of International Conference on Applied Innovation in IT (Vol. 13, No. 2, pp. 93-102). Anhalt University of Applied Sciences. [Online]. .Available: https://icaiit.org/proceedings/13th_ICAIIT_2/1-10-ICAIIT_2025_13 (2).pdf.

[5] Nsaif, W. S., Salih, H. M., Saleh, H. H., & Al-Nuaimi, B. T. (2024). Chatbot development: Framework, platform, and assessment metrics. The Eurasia Proceedings of Science Technology Engineering and Mathematics, 27, 50-62. https://doi.org/10.55549/epstem.1518314.

[6] Arazzi, M., Cotogni, M., Nocera, A., & Virgili, L. (2023, June). Predicting tweet engagement with graph neural networks. In Proceedings of the 2023 ACM international conference on multimedia retrieval (pp. 172-180). [Online]. .Available: https://arxiv.org/pdf/2305.10103.

[7] Ng, L. H. X., & Carley, K. M. (2025). A global comparison of social media bot and human characteristics. Scientific Reports, 15(1), 10973. https://doi.org/10.1038/s41598-025-96372-1.

[8] Feng, S., Wan, H., Wang, N., Li, J., & Luo, M. (2021, October). Twibot-20: A comprehensive twitter bot detection benchmark. In Proceedings of the 30th ACM international conference on information & knowledge management (pp. 4485-4494). https://dl.acm.org/doi/epdf/10.1145/3459637.3482019

[9] Narayan, N. (2021, September). Twitter bot detection using machine learning algorithms. In 2021 fourth international conference on electrical, computer and communication technologies (ICECCT) (pp. 1-4). IEEE.

[10] Cresci, S. (2020). A decade of social bot detection. Communications of the ACM, 63(10), 72-83. https://doi.org/10.1145/3409116.

[11] Kudugunta, S., & Ferrara, E. (2018). Deep neural networks for bot detection. Information Sciences, 467, 312-322. https://doi.org/10.1016/j.ins.2018.08.019.

[12] Feng, S., Wan, H., Wang, N., & Luo, M. (2021, November). BotRGCN: Twitter bot detection with relational graph convolutional networks. In Proceedings of the 2021 IEEE/ACM international conference on advances in social networks analysis and mining (pp. 236-239). [Online]. .Available: https://arxiv.org/pdf/2106.13092.

[13] Gangireddy, S. C. R., P, D., Long, C., & Chakraborty, T. (2020, July). Unsupervised fake news detection: A graph-based approach. In Proceedings of the 31st ACM conference on hypertext and social media (pp. 75-83). [Online]. .Available: https://pureadmin.qub.ac.uk/ws/files/212663108/ht20_crc.pdf

[14] Chinnaiah, V., Dhayanithi, M., Patturaj, S., Ranganathan, R., & Mohan, V. B. (2023, November). Fake Trend Detection in Twitter Using Machine Learning. In International Conference on Computing and Communication Networks (pp. 1-11). Singapore: Springer Nature Singapore. [Online]. .Available: https://link.springer.com/chapter/10.1007/978-981-97-2671-4_1#citeas

[15] Arin, E., & Kutlu, M. (2023). Deep learning based social bot detection on twitter. IEEE Transactions on Information Forensics and Security, 18, 1763-1772.

[16] Fu, C., Shi, S., Zhang, Y., Zhang, Y., Chen, J., Yan, B., & Qiao, K. (2023). Squeezegcn: adaptive neighborhood aggregation with squeeze module for twitter bot detection based on gcn. Electronics, 13(1), 56. https://doi.org/10.3390/electronics13010056.

[17] Tzoumanekas, G., Chatzianastasis, M., Ilias, L., Kiokes, G., Psarras, J., & Askounis, D. (2024). A graph neural architecture search approach for identifying bots in social media. Frontiers in Artificial Intelligence, 7, 1509179. https://doi.org/10.3389/frai.2024.1509179.

[18] Wang, X., Chen, K., Wang, K., Wang, Z., Zheng, K., & Zhang, J. (2024). FedKG: A knowledge distillation-based federated graph method for social bot detection. Sensors, 24(11), 3481. https://doi.org/10.3390/s24113481.

[19] Zeng, K., Li, Z., & Wang, X. (2025). Emoji-driven sentiment analysis for social bot detection with relational graph convolutional networks. Sensors, 25(13), 4179. https://doi.org/10.3390/s25134179.

[20] Ilias, L., Kazelidis, I. M., & Askounis, D. (2024). Multimodal detection of bots on x (Twitter) using transformers. IEEE Transactions on Information Forensics and Security. [Online]. .Available: https://arxiv.org/pdf/2308.14484

[21] Jadhav, K., Potikas, P., Pollett, C., & Potika, K. (2025, July). Multirelational Twitter Bot Detection Using Graph Neural Networks. In 2025 IEEE 11th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService) (pp. 147-154). IEEE. [Online]. .Available: https://ieeexplore.ieee.org/document/11129504.

[22] Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. Computational Social Networks, 6(1), 1-23. https://doi.org/10.1186/s40649-019-0069-y.

[23] Jarrahi, A., Mousa, R., & Safari, L. (2023). SLCNN: Sentence-level convolutional neural network for text classification. arXiv preprint arXiv:2301.11696. https://doi.org/10.48550/arXiv.2301.11696.

[24] Soni, S., Chouhan, S. S., & Rathore, S. S. (2023). TextConvoNet: a convolutional neural network based architecture for text classification. Applied Intelligence, 53(11), 14249-14268. https://doi.org/10.1007/s10489-022-04221-9

[25] Anwer, S. (2025). Deep Neural Network and Transformer Models for Emotion Recognition. Bilad Alrafidain Journal for Engineering Science and Technology, 4(1), 100-112. https://doi.org/10.56990/bajest/2025.04.