Social Bot Detection Using a Hybrid Graph Approach

Abstract

The widespread presence of malicious bots on Online Social Networks (OSNs) has been identified as a critical threat affecting public discourse, social well-being, and the foundational trust in the digital ecosystem. We overcome the limitations of the existing graph-based detection approaches by (1) Capturing tweet relationships between user and tweet nodes and (2) Utilising user metadata within Graph Neural Networks (GNN). A comparison-based empirical evaluation between classical machine learning and graph-based methods has shown that this hybrid approach outperforms most of the existing methodologies. In response to recent concerns regarding the reliability of published bot detector models, our study revealed that Twitter accounts discussing political topics are the most prone to misclassification. We highlight the effectiveness of a hybrid graph approach for accurate identification of bots engaged in research, business, social issues, and common expressions. However, greater caution should be exercised when attempting to identify bots involved in political discourse.

Keywords: Graph Neural Network, Twitter bots, social network, misclassification analysis, feature engineering

1. Introduction

Over the last decade, social media has surged to the forefront of global communication and information exchange, acting as a pivotal platform for individuals to disseminate information and shape their perspectives. This phenomenon offers numerous benefits, but its prominence has also presented an avenue for malicious actors to exploit through the use of bots, enabling the manipulation of public opinion and dissemination of misinformation.

Amongst the numerous social media platforms, Twitter has emerged as a dominant force, renowned for its concise yet engaging format, characterised by tweets and retweets that foster continuous interaction and interconnectedness within the community. While this characteristic tweeting feature undoubtedly fosters dynamic engagement, it has also inadvertently facilitated a pressing issue: the rapid propagation of an echo chamber of misinformation.

In light of these challenges, past scholarships have pivoted to traditional machine learning techniques, such as Random Forests and Support Vector Machines (SVMs), to distinguish between bots and human users by analysing distinct account features like location and IP address (Yang et al., 2020). Traditional methods have also analysed content via Natural Language Processing (NLP) techniques and activity patterns such as measuring posting frequencies to identify patterns between bots (N. Chavoshi et al., 2016; Wei et al, 2020).

However, recent studies have revealed that these methods are vulnerable to adversarial manipulation and overfitting (Cresci et al., 2021). Adversarial manipulation occurs when bot operators identify characteristics within existing detection systems, allowing them to adapt and refine their bots to exhibit more human-like behaviours. This renders traditional methods less resilient over time and less effective against sophisticated bot operators. (Feng et al., 2021).

To address these challenges, our paper proposes an approach utilising GNNs to analyse the complex interconnections within Twitter's social network.

GNNs are a type of neural network suited for tasks involving graph-structured data. It offers a powerful framework for learning the representation of nodes in a graph by iteratively aggregating information from neighbouring nodes. This effectively encodes structural properties of the graph into lower dimensional embeddings and enables the network to benefit from the rich contextual information. By leveraging this, GNNs are able to capture complex relational dependencies amongst nodes (Hamilton et al., 2018).

In theory, GNNs demonstrate resilience against adversarial manipulation. Despite attempts by bot operators to alter account behaviours, the fundamental nature of bots as mass-created entities and their malicious intent of spreading misinformation through retweets remains unchanged. Consequently, bot accounts exhibit identifiable patterns distinct from those of genuine Twitter users. By representing each account as a node within the GNN, we can effectively capture these patterns and clusters of bot activity regardless of temporal changes or manipulation attempts.

Building on this insight, recent endeavours have explored employing GNNs for bot detection. However, the lack of comprehensive datasets suitable for model training and the predominant focus on analysing solely followers / following relationships between accounts have shown headwinds (Refer to Section 2). Consequently, the outcomes are limited when compared to those of traditional methods.

Various existing architectural designs also diverge based on their approaches to node aggregation. Therefore, we aim to dive deeper into these diverse methods while utilising more robust datasets and discerning optimal strategies for effective graph representation learning in the realm of social bot detection.

Lastly, in social bot detection, a fundamental challenge also lies in the evasion tactics employed by certain bots to dodge detection. Therefore, our research will also investigate the characteristics of misclassified bots that have successfully evaded detection in the context of GNN models. By analysing these instances, our study aims to yield insights into the limitations of existing detection methods and propose potential enhancements for more robust bot detection strategies.

With these goals in mind, our paper builds on existing works and proposes an innovative GNN-based machine learning framework that integrates several key enhancements.

- (1) Incorporating key relationships, retweets and replies, as edges between account nodes to provide an alternative graph structure to SEGCN and BotRGCN (Feng et al., 2021; Liu et al., 2024). (2) Addressing limitations to Kulkarni's and BotRGCN graph structure by enriching account nodes with metadata, including attributes such as demographics and activity patterns (Feng et al., 2021; Kulkarni, 2023). (3) Experimenting various aggregation methods within the GNN framework, with GraphSage Mean Aggregation emerging as the most suitable method.
- Through topic clustering and TF-IDF based records ranking, we further identify the characteristics of misclassified bots in the domain of GNN architectures, which can hopefully provide foundations for future works in this area.

Our GNN model has shown promise, demonstrating competitive performance against various benchmark models by achieving 93.48%, 89.78% and 93.08% in accuracy, precision and F1-score respectively. As our model bears the scalability and flexibility in deploying graph-based training with less dependency on solely preprocessed features, the strong performance of our model underscores its robustness and its potential in the field of social bot detection.

The organisation of the paper is as follows. Section 2 will cover the background research and datasets used for the paper. Section 3 will describe our methodology and processes while Section 4 will highlight our experimental results. Finally, the remaining sections will provide the limitations of the paper, an overall conclusion and areas for future work.

2. Background

2.1. Conventional Machine Learning Methods

Conventional Machine Learning techniques have played a significant role in the ongoing battle against Bot Detection. Models like XGBoost, Naive Bayes, and Decision Trees, coupled with sophisticated Feature Engineering strategies, have been pivotal in combating this challenge (Ramalingaiah et al., 2021; Shevtsov et al., 2022). Notably, continuous advancements in feature engineering have consistently bolstered the accuracy of these traditional methods (Yang et al., 2020).

Deep Learning approaches have emerged as formidable contenders in this domain. Particularly, NLP technologies, including Long Short-Term Memory (LSTM) networks and BERT Models, have garnered attention for their ability to discern semantic nuances in tweets, thereby aiding researchers in distinguishing between authentic users and bots (Dukić et al, 2020; Wei et al, 2020).

However, it has been observed that the efficacy of such models diminishes over time, especially in light of the evolving nature of bots (Cresci, 2020; Feng et al., 2021). To address this issue, researchers have delved into extracting more comprehensive account metadata features. Yet, the resilience of these approaches against bot operators remains an open question.

Table 1. GNN aggregation methods

Graph Convolutional Networks (GCNs)	GraphSAGE (Sample & Aggregate)	Graph Attention Networks (GAT)
$H^{(l+1)} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l}$	$h_v^{(l+1)} = W^{(l)} \cdot \mathrm{AGG}\left(\{h_v^{(l)}\} \cup \{h_u^{(l)}\}\right)$ $, \forall u \in \mathcal{N}(v)$	$h'_i = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j$
Kpif & Weilling, 2016	Hamliton, 2017	Veličković, 2018

2.2. Emerging GNN Architectures

The growing popularity of GNNs for node prediction has led researchers to explore their potential in the domain of bot detection. Noteworthy contributions in this area include the models SEGCN and BotRGCN, as well as a publication by Kulkarni in Spring 2023 (Feng et al., 2021; Kulkarni, 2023; Liu et al., 2024).

However, these studies are not without limitations. For instance, Kulkarni's work relies on the Twibot-20 dataset, which is characterised by limited data scale and lower annotation quality (as data was collected via crowdsourcing), resulting in decreased accuracy compared to traditional models (Feng et al., 2024; Graells-Garrido et al., 2022). Furthermore, Kulkarni's proposed graph structure focuses on building a 2-level heterogeneous graph composed of accounts and tweets as nodes and edges defined by the accounts linked to these tweet nodes. This restricts the account activity network to the form of a subgraph of the entire network, potentially missing out on information found beyond this subgraph (Kulkarni, 2023).

The SEGCN and BotRGCN propose a more suitable homogenous graph structure to capture the full social media network, where nodes are defined as accounts and edges between nodes are characterised by the account's followers and following (Feng et al., 2021; Liu et al., 2024). However, with most Twitter account activities being predominantly retweets and replies, modelling a graph based on retweets/replies could potentially capture greater information compared to one that analyses following/follower relationships, with the latter being unable to capture the complexity of graphical relationships among accounts fully (Pastor-Galindo et al., 2022).

There are multiple GNN aggregation methods that are commonly used. Primarily, Graph Convolutional Networks (GCNs) were used to aggregate the graph structures. Similar to Convolutional Networks, it represents a node by aggregating the feature vectors of its neighbours with fixed weights inversely proportional to the central and neighbour node degrees (seen in Table 1).

Another alternative method includes the use of GraphSage, which supports mean, max and LSTM aggregation methods. This is a scalable method which aggregates information from a node's fixed-size neighbourhood (seen in Table 1).

Lastly, Graph Attention Networks (GAT) are also an alternative form of aggregation. It introduces an attention mechanism within message passing. During message-passing, each node calculates an attention score for each neighbour, and the messages are weighted based on these attention scores before being aggregated to update the central node's representation (seen in Table 1).

2.3. Misclassified Bots

Misclassified bots are a persistent challenge in social bot detection, with established tools like Botometer showing varying accuracies across different contexts (Kaiser et al., 2020). Traditional machine learning techniques have proven insufficient in overcoming this issue (Hays et al., 2023).

In the context of GNNs, it's clear that bots exhibit different behaviours depending on the context, leading to varying connectivity patterns within graphs. Certain types of bots may be more likely to slip through detection mechanisms due to these differences.

Therefore, our paper will investigate the efficacy of GNNs in these various contexts and their implications on graph connectivity.

2.4. Dataset

Dataset selection is critical to train an effective network. Out of the 18 listed datasets on the Bot

Repository, only 2 explicitly provide the graph structure amongst Twitter accounts. However, both datasets have limited scale, incomplete graph structure and low annotation quality (Feng et al., 2024). This leads to inaccurate results and resulting models may be susceptible to the false positive problem (Kaiser et al., 2020).

Therefore, our paper leverages the **TwiBot-22 dataset**. Consolidated in 2022, it addresses key limitations of existing Twitter bot detection datasets, by providing a comprehensive and complete heterogeneous graph structure with high-quality annotations (Feng et al., 2022). Additionally, it provides extensive metadata for feature-based machine-learning methods. Given the limited existing research applying this dual-faceted approach while utilising the TwiBot-22 dataset, we believe that the utilisation of this new dataset would add valuable new insights and pave the way for further advancements in Twitter bot detection.

3. Methodology

Our GNN-based framework leverages SEGCN's encoding approach of incorporating user metadata in node representation (Liu et al., 2024). We provide an alternative graph structure to the one found in BotRGCN by utilising retweets/replies instead of follower/following and experimenting with different GNN aggregation functions, ultimately converging towards the best-performing one. To achieve our ML goal, we divided our methodology into three phases: sampling, feature engineering, and graph construction.

3.1. Sampling

Since we wanted to perform a combination of feature-based and graph-based approaches, we used Disproportionate Stratified Sampling (DSS) to achieve a balanced representation of human and bot accounts, with the aim of creating a class-balanced graph network. To ensure a well-connected graph as well as strong feature information within nodes, the order in which we extracted and sampled from our dataset was important. We first sampled the accounts based on the DSS strategy to establish a connected and class-balanced graph. Subsequently, we extracted the corresponding features for these accounts. In contrast to our dataset, we were able to obtain a well connected sample that maintained the balance between human and bots, thereby maintaining both graph connectivity and population representation.

3.2 Feature Engineering

3.2.1. Feature Extraction. While the TwiBot-22 provides extensive account metadata and temporal features, past research has shown that the bot detection algorithms benefitted from creating more robust features from the basic account data. These features were engineered based on domain knowledge and we adapted them in this project. We utilise three schools of thought to engineer new features to pick up potential bot behaviour (1) User Profile Features, (2) User Engagement Features & (3) User Activity Features (Feng et al., 2024).

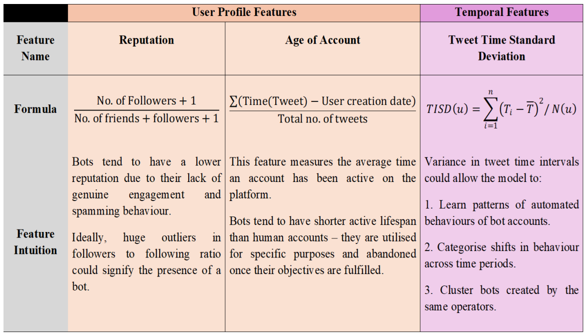
Unlike other graphical models, our objective was to combine insightful user-specific features, offering valuable insights into individual account characteristics, alongside user engagement and activity features, which characterise account behaviours (Kulkarni, 2023).

Table 2. Feature groupings.

User Profile Features	User Engagement Features	User Activity Features
Username length	Reputation	Tweet Time Standard Deviation
Name length	Age of the account	Max number of URLs in tweets
Description length	Retweet Ratio	Max number of mentions in tweets
Number of digits in username	URL Ratio	Max Number of Hashtags in tweet
Entropy of username	Mention Ratio	
Entropy of description	Hashtag Ratio	
Name and username similarity	Average Length of Tweets by user	
Ratio of length of username to length of name	Average number of tweets containing URLs	
Username length	Reputation	

With this approach, we successfully identified approximately 20 crucial features that could be extracted using the data available in TwiBot-22. Table 2 presents a few examples of these features, alongside their corresponding mathematical formulas and intuitive explanations in Table 3.

Table 3. Examples of formulas and intuitions of features selected.



	User Profile Features		Temporal Features
Feature Name	Reputation	Age of Account	Tweet Time Standard Deviation
Formula	$\frac{\text{No. of Followers} + 1}{\text{No. of friends} + \text{followers} + 1}$	$\frac{\sum(\text{Time(Tweet)} - \text{User creation date})}{\text{Total no. of tweets}}$	$TISD(u) = \sum_{i=1}^{n}\left(T_i - \overline{T}\right)^2 / N(u)$
Feature Intuition	Bots tend to have a lower reputation due to their lack of genuine engagement and spamming behaviour. Ideally, huge outliers in followers to following ratio could signify the presence of a bot.	This feature measures the average time an account has been active on the platform. Bots tend to have shorter active lifespan than human accounts – they are utilised for specific purposes and abandoned once their objectives are fulfilled.	Variance in tweet time intervals could allow the model to: 1. Learn patterns of automated behaviours of bot accounts. 2. Categorise shifts in behaviour across time periods. 3. Cluster bots created by the same operators.

3.2.2. Feature Selection and Analysis. In order to compute tweet data and information, we utilised GloVe embeddings for all our textual data. This approach enriches our feature space with semantic information, enabling more robust and nuanced representations of the data, which leads to an improvement in our model's performance.

For feature selection, we utilised a correlation analysis, along with Random Forest and XGBoost to quantitatively identify significant features for our model. We also analysed the means and variances of each feature for both the human and bot to guide us in the process of pruning unimportant features.

Typically, significant features exhibit a notable disparity in median and distribution between humans and bots, as illustrated in Figure 1. Consequently, we discarded features demonstrating similar distributions between humans and bots, as they proved to be less informative, as seen in Figure 2.

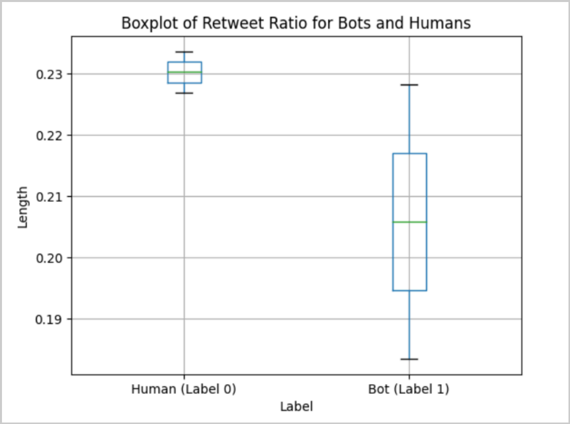


Figure 1. Ideal distribution within a feature.

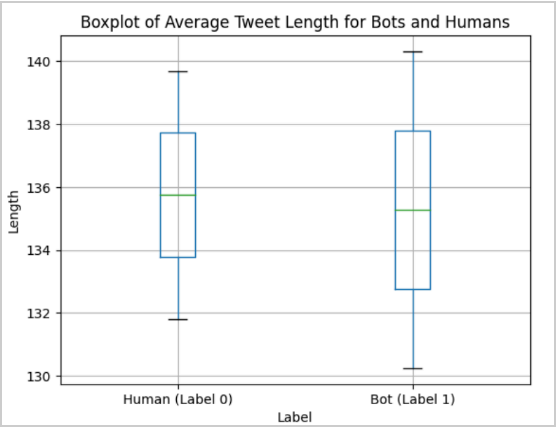
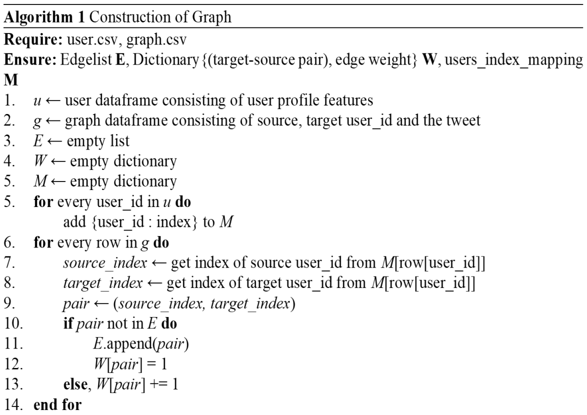


Figure 2. Distribution of a rejected feature.

3.3. Graph Construction

From the existing models we studied, a heterogenous graph is constructed where Twitter users are treated as nodes and their following-follower relationships are mapped as edges. However, many researches have shown that bots exhibit vastly different retweeting and replying habits from humans, and it could be a strong signal for distinguishing them in the dense social network (Gilani et al., 2019). Furthermore, there could be a case of inactivity between follower-following despite this relationship between accounts.

Considering retweets and replies form the bulk of social media activity and interaction between accounts, we have decided to adapt to utilising retweet/reply relationships as the edges to construct a homogenous graph where the edge weight represents the frequency of the interactions between the users, as shown in Figure 3 (Pastor-Galindo et al., 2022). Each node is also represented by a vector consisting of the user's aggregated tweet embedding to capture semantic properties as well as the profile features.



Algorithm 1 Construction of Graph

Require: user.csv, graph.csv

Ensure: Edgelist **E**, Dictionary {(target-source pair), edge weight} **W**, users_index_mapping **M**

```
$u$ ← user dataframe consisting of user profile features
$g$ ← graph dataframe consisting of source, target user_id and the tweet
$E$ ← empty list
$W$ ← empty dictionary
$M$ ← empty dictionary
for every user_id in $u$ do
    add {user_id : index} to $M$
for every row in $g$ do
    $source\_index$ ← get index of source user_id from $M$[row[user_id]]
    $target\_index$ ← get index of target user_id from $M$[row[user_id]]
    $pair$ ← ($source\_index$, $target\_index$)
    if $pair$ not in $E$ do
        $E$.append($pair$)
        $W$[$pair$] = 1
    else, $W$[$pair$] += 1
end for
```

Figure 3. Algorithm for graph construction

This edge definition in our method of graph construction is one key difference that separates us from recent works of the SEGCN and BotRGCN (Feng et al., 2021; Liu et al., 2024). At the time of writing, few if any thorough research has been done that uses reply/retweet to map social relationships between users, on top of capturing both semantic properties in tweets and profile properties in their node embeddings. The visual representation of the graph is illustrated in Figure 4.

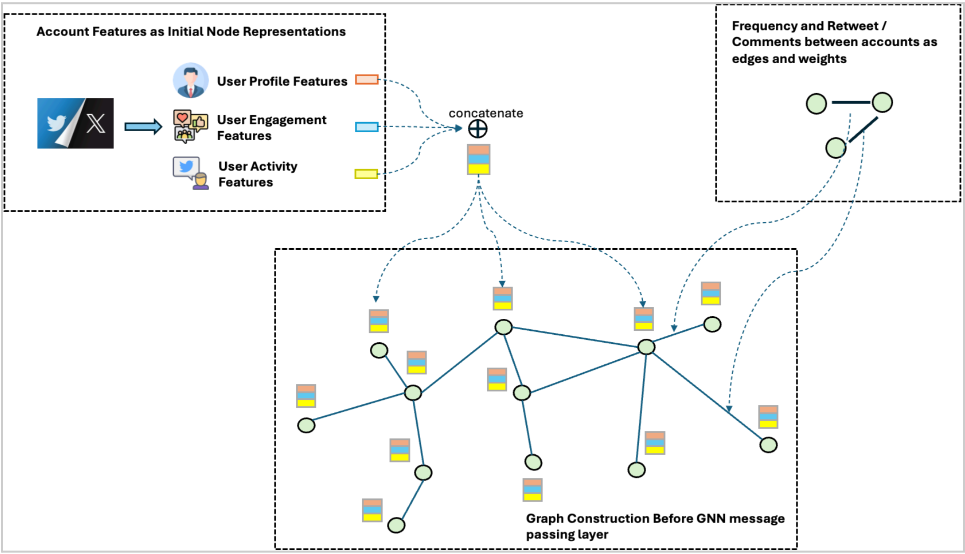

Account Features as Initial Node Representations
User Profile Features
User Engagement Features
User Activity Features
concatenate
Frequency and Retweet / Comments between accounts as edges and weights
Graph Construction Before GNN message passing layer

Figure 4. Visualisation of node embeddings & graph construction.

4. Results and Discussion

Our objective is to carry out node classification on the input graph and predict the label of each node as human or bot. This section compares the performance of different GNN models as well as other classical ML methods and models used in existing literature, with the aim of showing our model's suitability as a solution in the field of social bot detection

4.1. GNN Models

To find the most optimal model for this machine learning problem, we include a hyperparameter tuning step in the evaluation process. As the GraphSAGE and GAT have their distinct hyperparameters on top of the baseline GCN hyperparameters, we perform a two-step hyperparameter tuning process. First, we performed the manual task of searching for the most optimal hyperparameters using the GCN as a baseline model. This step includes evaluating 4 GCN models with different hyperparameters. Refer to Table 4 for an overview of the hyperparameter tuning results.

Surprisingly, we found that increasing the dimensions and number of Graph Convolution layers did not significantly improve model performance. In contrast, the incorporation of Batch Normalisation layers before the Graph Convolution layers appears to improve model performance significantly. Ultimately, we find GCN 4, with the above hyperparameters, performed the best and will proceed to use these hyperparameters for training the other GNN models. Secondly, we extended the hyper-parameter tuning step to find the best aggregation method for the GraphSAGE (mean, max and LSTM) and the most optimal number of attention heads (2 or 4) for the GAT. Refer to Table 5 for an overview of the hyperparameter tuning results.

Table 4. Hyper-parameter tuning results I (GCN as baseline).

Model	Hyper-parameters	Accuracy	Precision	Recall	F1-Score

GCN 1	GCNConv(110, 110)
	Dropout(0.3)
	BatchNorm1d(110)
		0.7926	0.8045	0.7173	0.7584

GCN 2	GCNConv(110, 110)
	GCNConv(110, 110)
	Dropout(0.3)
	BatchNorm1d(110)
		0.7178	0.6439	0.8458	0.7312

GCN 3	Linear(110, 128)
	GCNConv(110, 110)
	Dropout(0.2)
	BatchNorm1d(110)
	Feed-forward Network
		0.7685	0.6917	0.8838	0.7760

GCN 4	BatchNorm1d(110)
	Linear(110, 64)
	GCNConv(64, 64)
	Dropout(0.2)
	BatchNorm1d(64)
	Feed-forward Network
		0.9126	0.8819	0.9322	0.9063

Table 5. Hyper-parameter tuning results II (GraphSAGE and GAT).

Model	Hyper-parameters	Accuracy	Precision	Recall	F1-Score
GraphSAGE 1	SAGEConv(64,64, "max")	0.9120	0.8648	0.9555	0.9120
GraphSAGE 2	SAGEConv(64,64, "mean")	0.9348	0.8978	0.9662	0.9308
GraphSAGE 3	SAGEConv(64,64, "LSTM")	0.9261	0.8898	0.9555	0.9261
GAT 1	SAGEConv(64,64, heads = 2)	0.9308	0.9005	0.9529	0.9260
GAT 2	SAGEConv(64,64, Heads = 4)	0.9287	0.9066	0.9398	0.9229

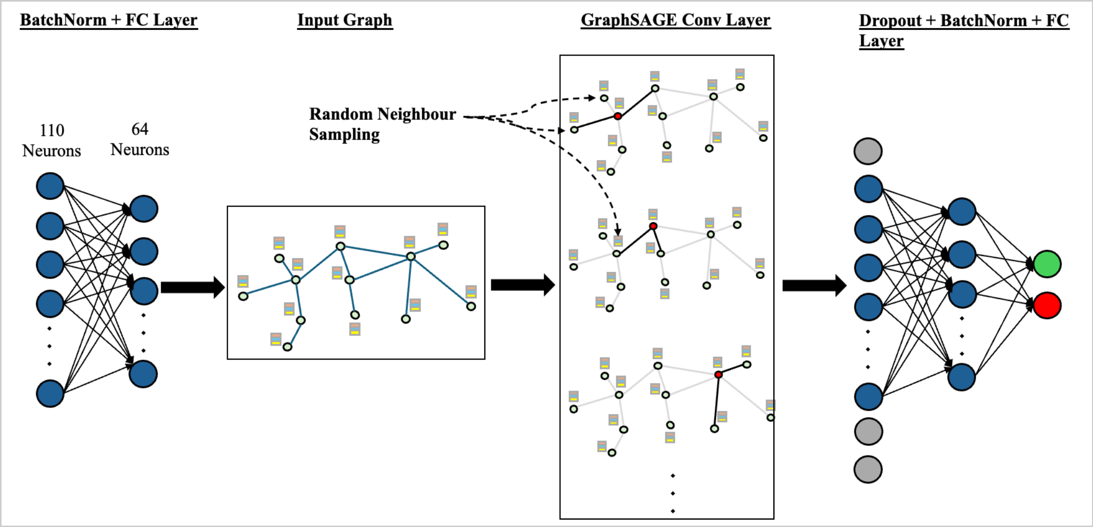


Figure 5. Our proposed GraphSAGE architecture.

We conclude that GraphSAGE, employing a mean aggregation function, outperformed other models and gave the best results with the highest accuracy, recall and F1-score across the 5 different permutations of GAT and GraphSAGE. Figure 5 above illustrates the architecture of our best-performing GNN model which utilises mean aggregation for message passing and neighbourhood sampling proposed in the GraphSAGE architecture (Hamilton et al., 2018). We utilised Adam optimiser with 0.001 learning rate and 0.05 decay, and cross entropy loss as our loss function. We also referenced existing papers in using leaky ReLu as our activation function to overcome the dying ReLu problem.

4.2. Performance Against Classical Methods

Our final evaluation layer for the GraphSAGE model involved a comparative analysis with classical machine learning methods, which are commonly utilised as benchmarks in the field of bot detection on social media. This comparative approach helps underline the relative effectiveness and suitability of advanced graph-based techniques in identifying bots compared to traditional methods (Kulkarni, 2023). The results are outlined in Table 6 below.

Overall, our proposed pipeline and model have shown significant success, outperforming most classical machine learning methods, except Random Forest and XGBoost. However, it holds great potential as theoretically, it would perform significantly better given the full use of the TwiBot-22 with a more complete and connected graph structure. Additionally, it has greater scalability for the ever-changing nature of bot evolution in the social media landscape given its inductive underlying methodology.

Table 6. Final model (GraphSAGE 2) performance against classical methods.

Model	Accuracy	Precision	Recall	F1-Score
GraphSAGE 2	0.9348	0.8978	0.9662	0.9308
Logistic Regression	0.7686	0.6924	0.8841	0.7766
K Nearest Neighbour	0.9296	0.8950	0.9576	0.9263
Support Vector Machine	0.7647	0.6671	0.9224	0.7810
Random Forest Classifier	0.9506	0.9190	0.9775	0.9473
XGBoost Classifier	0.9539	0.9288	0.9731	0.9504

4.3. Evaluation of Model Against Existing Benchmarks

Finally, we evaluate the performance of our model with the existing benchmark models that are tested on TwiBot-22 in Table 7. Though direct comparisons are not possible due to differences in datasets, the strong performance suggests that our model shows great promise in the field of social bot detection.

Table 7. Comparison with existing benchmarks

Model	Accuracy	Precision	F1-Score
Our Model	0.9348	0.8978	0.9308
BotRGCN (Feng et al., 2021)	0.7966	0.7481	0.5750
SEGCN (Liu et al., 2024)	0.8271	0.7723	0.5931

4.4. Analysis of Misclassified Bots

To further understand the potential limitations of our model, it is crucial to study the types of bots the model struggled with correctly classifying. We employed k-means clustering to analyse the aggregated tweet embeddings of all bots. As these embeddings are numerical representations that capture the semantic meaning of each bot's tweet corpus, we aimed to group bots with similar content in their tweets together. This approach allowed us to explore underlying patterns within the bot population that might have contributed to misclassification. We determined the optimal number of clusters to be 5 by using the Elbow method (Refer to Figure 6).

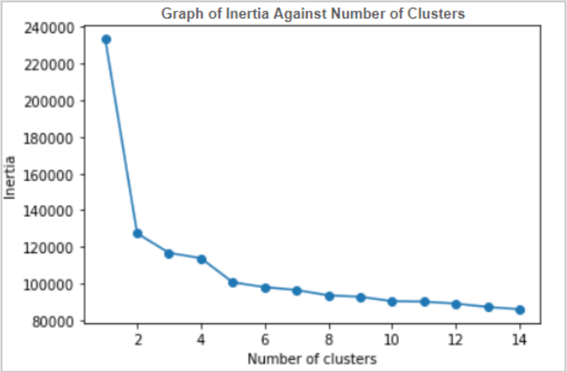


Figure 6. Graph of inertia against number of clusters.

After obtaining the clusters, we extracted all the bots' tweets and performed the steps described below.

- Preprocessing: Text data from each cluster underwent preprocessing, including converting to lowercase, removing punctuation, tokenization, and stop word removal (using both built-in and a custom list of additional stop words). This custom list included words that were commonly observed but left out in the built-in list.

- TF-IDF Vectorisation: A TF-IDF vectorizer was created and fitted on the preprocessed sentences within each cluster. TF-IDF helps identify the most important words within a document collection, considering both a word's frequency within a cluster and its overall infrequency across all clusters.

- Extracting Top Words: By analysing the TF-IDF vectors, we were able to extract the top 30 words with the highest average TF-IDF scores within each cluster. These words likely represent the most prominent themes and topics discussed by the bots in that particular cluster (Refer to Table 8).

Table 8. Top words and thematic interpretation from clusters.

Cluster	Top Words	Theme
1	ukraine, russian, putin, nato, trump, biden, president	Politics
2	think, make, know, good, need, want, say, right	Common Expressions
3	market, startup, company, digit, develop, build, innovation, technology	Business
4	like, follow, happiness, come, know, giveaway, want	Social
5	paper, research, learn, science, model, study, robot	Research

After training and testing our model, we identified the bots that had been misclassified as humans and calculated the percentage of these bots within each cluster (Refer to Figure 7).

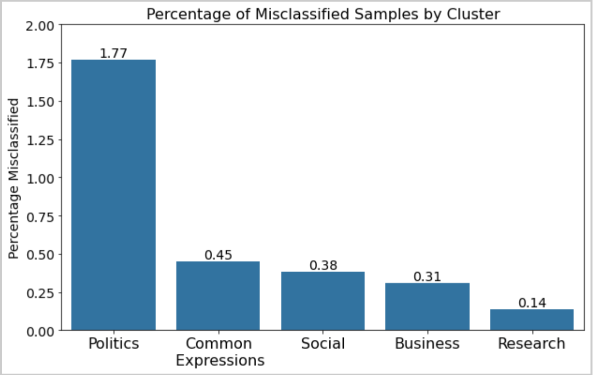


Figure 7. Proportion of misclassified bots by clusters.

The bots most frequently misclassified belong to the cluster involved in political activities. This cluster's misclassification rate of 1.77% represents a substantial increase from the 0.45% rate of the second-highest misclassified cluster, which deals with common expressions. This discrepancy indicates that the bots engaged in political topics pose a remarkable challenge to the accuracy of our model. Bots involved in the topic of politics often engage in a wide range of activities, from posting original content to retweeting and replying to various political opinions. This diversity makes them harder to classify because they are designed to blend in with genuine political discourse. They also use nuanced language, emotional appeals, and complex rhetoric designed to influence public opinion. This complexity in language use can make it more challenging for detection as they mirror human users' language.

The least commonly misclassified bots are those in Cluster 5 which is involved in the field of research. The misclassification rate is 0.14%, which is a considerable reduction from the second-lowest cluster at 0.31%. While this might seem minor, this might play a role in helping us better understand the biases of our model. Bots involved in the topic of research typically exhibit more specific and predictable behaviours, such as sharing links to academic papers or scientific news. Their interactions are usually less varied and more focused on particular topics, making their patterns easier to detect and classify accurately. They also tend to use more formal and specialised language related to specific fields of study which makes them easier to identify because they stand out from the more varied and informal language used by genuine users in general conversations.

5. Conclusion

In this study, we investigated the increasing prevalence of malicious bots and their contribution to the rapid spread of misinformation. This is further propagated by the nature of social media, with Twitter being one of the largest platforms. This ultimately causes huge harm in our society.

Traditional machine learning techniques have been previously employed in bot detection, but these methods have shown vulnerabilities to adversarial manipulation and overfitting. Bot operators can refine their bots to mimic human behaviours, thus undermining the reliability of traditional detection methods over time.

To address these challenges, we explored the use of GNNs to analyse the complex interconnections within Twitter's social network. GNNs are able to capture the structural properties and relational dependencies among nodes, providing resilience against manipulation from bot operators. Our approach leverages the intrinsic patterns exhibited by bot accounts, which remain detectable despite attempts to mimic human behaviour. Furthermore, our paper addresses the limitations of previous studies by incorporating retweets, replies, and selected activity features along with account metadata, providing a more comprehensive depiction of account interactions.

Our experimental results show the effectiveness of our GNN-based approach in bot detection, outperforming traditional methods. This highlights the potential of GNNs to serve as a robust and resilient tool in mitigating the spread of misinformation on social media platforms.

However, our study is not without limitations. While our goal is to build a model that can precisely classify and predict which users are bots or humans, the goal of social media platforms is to prioritise user experience and user retention. It is more costly to misclassify a human as a bot compared to misidentifying a bot, as it can lead to reduced user satisfaction and an erosion of trust, which ultimately damages the company's reputation and discourages new users from joining the platform.

6. Future Work

We propose that future works could look into minimising false positives. This means adopting a cautious approach where we err on the side of classifying an account as human unless there's compelling evidence suggesting otherwise. A weighted loss function could be used to assign a higher weight to the loss associated with false positives compared to the loss of false negatives.

Additionally, a deeper study could be done to further extend our analysis of the impact of different bot types on our model's ability to distinguish between human and bot accounts. Our analysis on misclassified bots was able to identify specific clusters of bots, and we foresee several avenues for future research.

One promising direction is to replicate our current machine-learning framework by sampling accounts that exhibit specific words or themes in their posts, tweets, or replies. This approach will enable us to explore whether thematic content impacts the model's ability to identify specific types of bots, such as political bots, spam bots, or social bots. By focusing on the linguistic and thematic characteristics of the account content, we can investigate if certain themes are more challenging for the model to detect and whether this varies across different bot categories.

A broader and more detailed examination of how different types of bots influence the effectiveness of social bot detection models is warranted. Understanding these nuances will not only refine our detection algorithms but also provide deeper insights into the behaviours and strategies of various bot types. This comprehensive analysis will ultimately enhance the robustness and accuracy of social bot detection methods, contributing to more secure and reliable online platforms.

7. References

Alarfaj, F. K., Ahmad, H., Khan, H. U., Alomair, A. M., Almusallam, N., & Ahmed, M. (2023). Twitter bot detection using diverse content features and applying machine learning algorithms. *Sustainability*, *15*(8), 6662.

Cresci, S. (2020). A decade of social bot detection. *Communications of the ACM*, *63*(10), 72–83.

Cresci, S., Petrocchi, M., Spognardi, A., & Tognazzi, S. (2021). Adversarial machine learning for protecting against online manipulation. *IEEE Internet Computing*. 47-52.

Dukić, D., Keča, D. and Stipić, D. (2020). Are you human? Detecting bots on twitter using BERT. *IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, Sydney, NSW, Australia, 2020, 631-636.

Feng, S., Tan, Z., Wan, H., Wang, N., Chen, Z., Zhang, B., Zheng, Q., Zhang, W., Lei, Z., Yang, S., Feng, X., Zhang, Q., Wang, H., Liu, Y., Bai, Y., Wang, H., Cai, Z., Wang, Y., Zheng, L., . . . Luo, M. (2022). TwiBot-22: Towards graph-based twitter bot detection. *arXiv (Cornell University)*.

Feng, S., Wan, H., Wang, N., & Luo, M. (2021). BotRGCN: Twitter bot detection with relational graph convolutional networks. *ASONAM '21: Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 236–239.

Ferrara, E. (2023). Social bot detection in the age of ChatGPT: challenges and opportunities. *First Monday, 28(6)*.

Gilani, Z., Farahbakhsh, R., Tyson, G., & Crowcroft, J. (2019). A large-scale behavioural analysis of bots and humans on twitter. *ACM Transactions on the Web*, *13*(1), 1–23.

Graells-Garrido, E., & Baeza-Yates, R. (2022). Bots don't vote, but they surely bother! A study of anomalous accounts in a national referendum. *WebSci '22: Proceedings of the 14th ACM Web Science Conference 2022*, 302–306.

Hamilton, W.L., Ying, R., & Leskovec, J. (2017). Representation learning on graphs: methods and applications. *IEEE Data Eng. Bull. 40*. 52-74.

Hays, C., Schutzman, Z., Raghavan, M., Walk, E., & Zimmer, P. (2023). Simplistic collection and labelling practices limit the utility of benchmark datasets for twitter bot detection. *WWW '23: Proceedings of the ACM Web Conference 2023*, 3660–3669.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Kulkarni, W. J. (2023). Twitter bot detection using NLP and graph classification. *Master's Projects*. 1263.

Lerner, E. (2021, November 17). *Social media bots may appear human, but their similar personalities give them away*. Penn Engineering Blog. https://blog.seas.upenn.edu/social-media-bots-may-appear-human-but-their-similar-personalities-give-them-away/

Liu, F., Li, Z., Yang, C., Gong, D., Lu, H., & Liu, F. (2024). SEGCN: a subgraph encoding based graph convolutional network model for social bot detection. *Scientific Reports*, *14*(1), 4122.

N. Chavoshi, H. Hamooni and A. Mueen, "DeBot: twitter bot detection via warped correlation," *2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 2016*, 817-822,

Pastor-Galindo, J., Mármol, F. G., & Pérez, G. M. (2022). Profiling users and bots in twitter through social media analysis. *Information Sciences*, *613*, 161–183.

Ramalingaiah, A., Hussaini, S., & Chaudhari, S. (2021). Twitter bot detection using supervised machine learning. *Journal of Physics. Conference Series*, *1950*(1), 012006.

Rauchfleisch, A., & Kaiser, J. (2020). The false positive problem of automatic bot detection in social science research. *PloS One*, *15*(10), e0241045.

Shevtsov, A., Tzagkarakis, C., Antonakaki, D., & Ioannidis, S. (2022). Explainable machine learning pipeline for twitter bot detection during the 2020 US Presidential Elections. *Software Impacts, 13*, 100333.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017, October 30). *Graph attention networks*. arXiv, abs/1710.10903

Wei, F., & Nguyen, U.T. (2019). Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 101-109.

Yang, K.-C., Varol, O., Hui, P.-M., & Menczer, F. (2020). Scalable and generalizable social bot detection through data selection. *Proceedings of the AAAI Conference on Artificial Intelligence, 34(01)*, 1096-1103.