A New Frontier: Exploring the Use of Large Language Models in Social Media Bot Detection

Rachit Agarwal
rachita@usc.edu

Balaj Khalid
bkhalid@usc.edu

Abdullah Siddiqui
ms74647@usc.edu

Akash Singh
akashsin@usc.edu

Veer Pandey
veeranil@usc.edu

University of Southern California

Abstract

The challenge of distinguishing bots from human users on social media platforms is an ongoing arms race between advancements in bot detection being offset by adversarial detection avoidance techniques. Our research aims to develop a robust multi-modal classification framework to detect bots on social media, specifically Twitter, by leveraging user metadata, tweet content, and follower/following graph structures. We propose a mixture-of-experts approach, where each modality is processed by a specialized model, and the final classification is determined through majority voting. Extensive experiments led us to conclude that using LLMs like LLaMA 3.1 with in-context learning and supervised fine-tuning show markedly improved results over traditional approaches. This work cements the fact that LLMs are the new frontier of social media bot detection and the only effective tool that can adapt to evolving adversarial bot strategies.

1 Introduction

Bot accounts on social media platforms, in particular on Twitter, have seen a rapid increase in recent years. These accounts are responsible for spreading fake news, misinformation, radical or extremist ideologies, conspiracy theories, and even blatant election interference (Lu and Li, 2020; Ng et al., 2022). Despite the magnitude of this problem and its impact on modern society, research on detecting social media bots effectively has always been an arms race, with any advances in bot detection being offset by rogue agents working to hinder the progress.

Recently, we have seen the advent of LLMs, which have been proven to excel in various text generation, classification, and related tasks (Naveed et al., 2024). LLMs are capable of following user instructions in order to achieve a desired result. In this paper, our aim is to utilize their capabilities to examine how state-of-the-art LLMs together with robust feature engineering could further the frontier of social bot detection.

2 Related Work

Bot detection on social media platforms has been extensively studied and can be categorized into different categories as follows:

Feature-based methods: Early bot detection methods relied on extensive feature engineering and extraction of hand-crafted features from user metadata to feed to classical algorithms in an attempt to distinguish bots and humans. The idea behind these methods was that bot metadata information is inherently different from legitimate users (Efthimion et al., 2018). These methods operated under assumptions which are no longer valid, and render most of these techniques obsolete.

Text-Based methods: These methods use modern NLP techniques such as word embedding, LSTM and attention mechanisms to classify user tweets as bot or human (Kudugunta and Ferrara, 2018). They can have high accuracy for specialized test sets. However, these methods fail to identify stolen tweets from genuine users and subsequently suffer from a high degree of false negatives.

Graph-Based methods: With the advent of Graph Neural Networks, graph-based algorithms quickly gained traction as they were able to analyze the inherent graph based structure of online social media platforms like Twitter and leverage this information to distinguish bots from humans (Bui and Potika, 2022; Cai et al., 2024). Graph-based methods lean on the fact that a bot account is likely to be followed by or following other bot accounts, and interact with other accounts in a markedly different manner to a human user.

LLM based methods: There has been very limited research into employing specific LLMs for bot detection and this area is still an active field of research. The recent work of Feng et al. (2024) addresses this challenge by employing off the shelf LLM with instruction tuning in an attempt to distinguish the bots from humans. However, with little feature engineering, limited experimental hyperparameter tuning and measures against adversarial attack, the work still leaves room for improvement.

3 Hypothesis

We propose that augmenting large language models with multimodal data along with robust feature engineering will outperform traditional and deep learning approaches in bot detection on the Twitter platform.

4 Methodology

4.1 Baseline Model

To evaluate the effectiveness of our model, we will be comparing our results against the baseline leaderboard for twibot-22 dataset which lists the performance of various algorithms on the dataset (Feng et al., 2023). To be even more specific, we will measure our results against three different baseline models from Yang et al. (2013); Guo et al. (2021); Hu et al. (2020) which implement random forest, transformer (BERT), and graph neural network respectively. We will also do our own implementation of four traditional approaches i.e. random forest, light gradient boosting, graph neural network, and transformer (BERT), and compare our LLM models with the conventional models.

4.2 Dataset

We have used the twibot-22 dataset for both training and testing large language models (Feng et al., 2023). Twibot-22 is a comprehensive graph-based Twitter bot detection benchmark that presents the largest Twitter dataset to date, provides diversified entities and relations on the Twitter network, and has considerably better annotation quality than other existing datasets in the domain. Twibot-22 provides 4 types of entities and 14 types of relations present in the twitter network. See Table 1 for more details about the dataset composition. This is an extremely rich dataset and a variety of research papers have worked upon this which provides us a good baseline to compare our model with (Feng et al., 2024).

For our purposes, the dataset can be divided into three main categories. It consists of user data (meta data information about user profile), tweet data (individual tweets of users and its related information), and edge data (specifies type of relation between two users in the dataset). The dataset also provides information about the train-val-test data split (see breakdown in Table 3), which is useful as it allows us to standardize our work so that we can accurately evaluate our model with the leaderboard for twibot-22 dataset. There is a separate file for labels that assigns a label to each user that we can use for evaluating our model with the ground truth. Please refer to the appendix for detailed visual and categorical analysis of the dataset.

Category	TwiBot-22
Humans	860,057
Bots	139,943
Users	1,000,000
Tweets	88,217,457
Human Tweets	81,250,102
Bot Tweets	6,967,355
Edges	170,185,937

Table 1: TwiBot-22 dataset statistics

4.3 Architecture Design

We are developing our model for a classification task where the input for our Large Language model will be the multimodal data of the user consisting of the profile information (metadata), the tweets of the user (text-based data) and a graphical representation of each user's follower/following information. The output will be a label defining the user as Human or Bot based on these input features.

We are employing a mixture of expert model frameworks where each specific modality in the user data will be analyzed separately (Liu et al., 2023). The output label corresponding to each model is combined through majority voting to provide the final output. For each modality, we have first employed a traditional machine learning method to benchmark its performance and then translated it into a large language model framework to compare and assess the effectiveness of LLMs for bot classification.

We also set up Apache Spark in order to utilize the extensive computation capability of CARC to process the 88 Million tweets contained in our dataset. We also created feature representations for our user's data. These features were then directly added to the list of metadata features. Additionally, we preprocessed the tweets of the user and developed a pipeline to sample tweets from the training data which were added to the large language model as in-context examples.

We then evaluated each modality of our dataset separately using LLMs to ensure their functionality, as well as to discover if any single modality was able to outperform the ensemble model. The approach taken for each particular modality is broken down in the next section. Once we successfully employed large language models across all modalities, we used an aggregation method to present the final classification output and prepare the evaluation report.

We also worked on creating a subset of our data for supervised-finetuning purpose. This subset was constructed by sampling instances from various modalities in the dataset and passing them through a processing algorithm. The algorithm generated a structured message object for each instance, encompassing four key elements: an instruction i.e. "The following task serves to distinguish between humans and bots depending on modality data provided", in-context examples relevant to the instance, the instance itself, and the ground truth label corresponding to the instance. This comprehensive approach allowed us to create a robust training set for supervised fine-tuning.

For the tweets and structure modalities, we employed a similarity mechanism to enhance the quality of samples selected for supervised finetuning. Specifically, the tweets were tokenized using the RoBERTa model, and cosine similarity scores were computed based on user metadata. The top-K samples with the highest similarity scores were then incorporated into the dataset. This ensured the inclusion of contextually relevant examples, thereby improving model learning.

The test set was generated in a similar manner, with the exception that the ground truth labels were omitted to maintain the integrity of the evaluation process.

To fine-tune a large language model (LLM) for our task, we utilized the aforementioned dataset in combination with the QLoRA technique. We selected LLaMA 3.1 as the base model and loaded it in a quantized format to enable computational efficiency. QLoRA employs low-rank adaptation (LoRA) (Hu et al., 2021) layers, which introduce a limited number of trainable parameters into specific components of the model, such as the attention mechanism, while keeping the rest of the model parameters frozen. Key hyperparameters for the LoRA configuration, including the rank of the low-rank layers, scaling factors, and dropout rates, were optimized to align with the task requirements.

Following model preparation, the training process was initiated by defining hyperparameters such as learning rate, batch size, number of epochs, and gradient accumulation steps to ensure efficient and effective fine-tuning. During this phase, only the parameters introduced by the LoRA layers were updated, significantly reducing the computational overhead compared to full fine-tuning.

Upon completion of training, the fine-tuned model, along with its tokenizer, was saved for future use. The model was then rigorously evaluated using an unseen subset of the dataset prepared earlier. This evaluation involved comparing the model's performance metrics, such as accuracy and F1-score, against the baseline and expected outcomes. The results confirmed the model's effectiveness in addressing the target task, validating the utility of the QLoRA fine-tuning approach in this context.

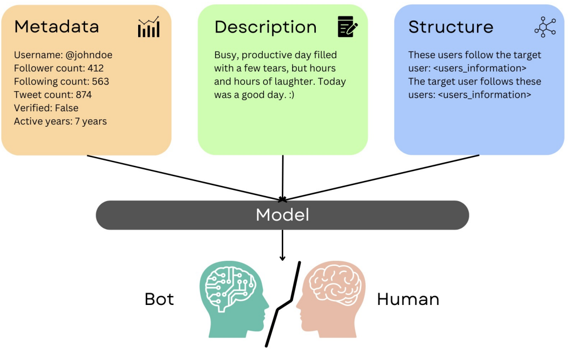

Metadata
Username: @johndoe
Follower count: 412
Following count: 563
Tweet count: 874
Verified: False
Active years: 7 years
Description
Busy, productive day filled with a few tears, but hours and hours of laughter. Today was a good day. :)
Structure
These users follow the target user: <users_information>
The target user follows these users: <users_information>
Model
Bot
Human

Figure 1: Sample Input and Output for our Model

4.4 Approach

4.4.1 User Data

To enhance the predictive capabilities of our model, we developed a robust data pipeline that transforms raw data into structured features. These engineered features are designed to capture nuanced behaviors that traditional bots struggle to replicate, especially focusing on temporal attributes like tweet rate and frequency. Our feature engineering efforts are summarized in Figure 3. Each feature is assessed for its relevance and importance in bot classification.

For the baseline models, we implemented a Random Forest Classifier and a Light Gradient Boosting Model (LightGBM). To evaluate the models under fair conditions, we performed k-fold cross-validation on the dataset, ensuring robust performance measurement across multiple runs. Both models were tuned using grid search to optimize the hyperparameters.

To handle the significant class imbalance in the dataset, we employed downsampling for the majority class to create a balanced subset. This allowed us to assess model performance in a way that mitigates skew, focusing on accurately capturing the minority class (bots) without overwhelming the models with majority-class examples.

Finally, we translated our settings from the traditional machine learning methods to a large language model setting where we used open source Models (primarily LLaMA 3.1) from Hugging Face to classify the data based on our extracted features and proceeded to compare its performance with Random Forest and LGBM. This analysis is detailed in later sections.

4.4.2 Tweet Data

In our initial approach, we only considered the `text` field i.e. the tweet. We preprocessed and tokenized the tweet using PorterStemmer and TweetTokenizer. We started off by analyzing our data using a baseline model. We elected to use BERT since that is a popular text classification model used in literature (Dukić et al., 2020). BERT performed poorly with small data size but performance improved as our dataset scaled. However, the model was overfit on training data and performed relatively poorly on test data, achieving a maximum test error of around 70 %.

Once we had enough computational resources, we processed the tweet data using pyspark pipelines. We extracted features related to the tweets. For each user we aggregated these features based on the mean. For the actual tweets of the user, we extracted 5 tweets for each user to subset the data in reasonable chunks. Next, we utilized all-MiniLM-L6-v2 and cosine similarity to create embeddings for each tweet and rank them based on similarity.

We then proceeded to apply the LLaMA 3.1 8B Instruct LLM model against our dataset. For in-context learning , for each test instance, we first obtain 10 most similar tweets and use them as in-context examples. Please note that since the number of tweets for humans were about 100 times more than bots in the dataset owing to its inherent imbalanced nature, we added checks to sample data so that a roughly equal number of human and bot instances would be selected as in-context examples. Subsequently, for supervised fine-tuning we create an instruction pair dataset with a similar prompt as the one used in our in-context learning experiment. The steps for supervised fine-tuning are explained in more detail in the previous section.

The sheer volume and imbalance within the tweets was one of the most challenging aspects of our project and we observe significantly less accuracy using this modality. This could be attributed to the fact that we made many simplifying assumptions as well as worked with subsets of the data which did not accurately reflect the actual data distribution.

4.4.3 Edge Data

For the third modality in our bot detection model, we utilized edge relations which are effective in capturing the graphical structure of Twitter platform. We subset our edge data based only on follower/following connections among users. This approach was chosen due to two main considerations: computational feasibility and alignment with existing research (Feng et al., 2021).

After defining the follower/following edges, we integrated the node metadata features derived in Section 4.4.1. Ideally we wanted to append these with the features extracted from the Tweet Data in Section 4.4.2, however, due to lack of computational resources we could only load a very small subset of Tweets and therefore a direct inclusion of those features is not possible. The metadata features were then combined with the edge structure to build a graph that represents both individual attributes and relational dynamics.

To take advantage of the information embedded in this graph structure, we implemented a Relational Graph Convolutional Network (RGCN). This model allowed us to encode relational information effectively, with RGCN layers designed to learn the complex dependencies between users based on both node attributes and their connections. The model was trained under controlled experimental conditions, where we split the data as specified by the train/validation/test splits provided in the twibot-22 dataset and determined a batch size of 32 and trained it for 100 epochs.

We explored several configurations to optimize the RGCN's performance, adjusting parameters like the number of convolution layers, learning rate, and dropout rates to optimize for its performance.

To switch from traditional approaches such as RGCN, we first created functions which could perform BFS search on our graph data structure to find the neighbors of each node and their attribute. These were then appended as in-context information for each test instance while running the inference on LLaMA 3.1. Please note that since the information appended for each neighbors was the metadata information, we utilized the same model we fine-tuned in Section 4.1.1 for running inference on the Graph network.

Of all modalities, the Edge data gave the highest performance boost compared to its traditional counterpart. This indicates the powerful capabilities of LLM on graph reasoning tasks.

4.5 Challenges

Limited Computational Access: The inability to install *bitsandbytes* library on CARC has restricted us from conducting supervised fine-tuning on this platform. As a result, we have had to explore alternative solutions, such as Kaggle. However, the computational resources available on Kaggle are limited, necessitating the use of a subset of the dataset for training.

Data Imbalance: The dataset itself is heavily imbalanced with the training data containing 92 percent human labels and only around 8 percent bot labels. This makes it challenging to be directly used with our models as they would not be able to generalize well to the minority class. This is problematic since our focus is identifying the minority class i.e. bot accounts. Moreover, this reduces the efficacy of using accuracy as evaluation metric and precision/recall and F-1 score metrics are needed to define our performance.

5 Results

Since our dataset is heavily imbalanced, we are evaluating our models using precision, recall and F1 score metrics in addition to accuracy. We do calculate the overall test accuracy of the model to compare with the existing benchmarks for Twibot 22 dataset. We aim to design a model to optimize recall as we want to flag as many malicious profiles as possible.

The experimental results obtained are outlined in Table 2. Please note that a direct comparison between the tweet data and other two modalities cannot be made since tweet data only uses a subset of actual data. For all our results we have evaluated on a test set of 10000 users except text data where we tested on a subset of 5% of all the tweets of a given user followed by a weighted sampling to obtain a 2:1 ratio of humans and bot tweets.

From Table 2, we observe that traditional and baseline approaches suffer due to the inherent imbalance of twibot-22 dataset providing average results with low recall, especially when utilizing the structural information. The limited capabilities of these models along with the intrinsic complexities of the dataset could be the source of these low scores.

Upon switching to large language models, we observe a significant increase in performance upon using in-context learning only with the LLaMA 3.1 8B model. We note significant improvement in the recall as well as accuracy of the model across all modalities. Furthermore, the overall ensemble scores suggest that each modality by itself only provides a partial information about the actual characteristics of the twitter profile, but the sum is stronger than the parts, with the ensemble model performing remarkably better than any individual modality.

After fine-tuning our models on an instruction pair dataset, we observe a slight increase in performance. Please note that we utilized a quantized model when performing supervised fine-tuning on Kaggle due to resource constraint issue and this could explain why the increase in performance is not significant as compared to the models with in-context examples. The resource constraint is further explained in section 4.5. This also indicates that with larger models and more computational resources we can expect a significant increase in the performance and establishes that LLMs have immense utility for bot detection.

6 Analysis

We conducted experiments with a wide range of models and techniques across all three modalities. We bench-marked our results against both traditional approaches such as random forest/ gradient boosting machines and advanced methods such as relational graph convolutional networks and BERT which can capture complex patterns in the twitter data and its inherent structure. Apart from implementing these traditional models ourselves, we also elected to choose three baseline models from the twibot-22 dataset leaderboard. These models are

Method	**M**	**T**	**N**	**Acc**	**Prec**	**Recall**	**F-1 Score**
Baseline Models (From Literature)							
EvolveBot (Random Forest)	✓	✓	✓	0.71	0.56	0.08	0.14
BGSRD (BERT GAT)	✓			0.72	0.23	0.20	0.21
HGT (Graph Neural Network)	✓	✓	✓	0.75	0.68	0.28	0.40
Traditional Approaches (Our Implementation)							
Random Forest	✓			0.62	0.64	0.67	0.62
Light Gradient Boosting	✓	✓		0.69	0.66	0.68	0.66
Graph Neural Network	✓		✓	0.69	0.45	0.49	0.42
Transformer (BERT)		✓		0.72	0.68	0.67	0.67
LLM based Approaches (LLaMA 3.1 8b Instruct)							
META (Random sampls)	✓			0.77	0.59	0.77	0.67
TEXT (Similar samples)		✓		0.57	0.56	0.68	0.61
STRUCT (User relations)	✓		✓	0.76	0.65	0.76	0.67
Ensemble	✓	✓	✓	0.78	0.68	0.76	0.72
Fine Tuned LLM (LLaMA 3.1 8b Instruct QLORA)							
META (Random samples)	✓			0.79	0.65	0.72	0.69
TEXT (Similar samples)		✓		0.69	0.60	0.67	0.63
STRUCT (User relations)	✓		✓	0.81	0.69	0.71	0.70
Ensemble	✓	✓	✓	0.83	0.72	0.76	0.74

Table 2: Evaluation results for various methods employed. The M, T, N columns stand for the Metadata, Text, and Neighboring nodes respectively. The Acc, Prec, Recall, F-1 Score columns serve to show the Accuracy, Precision, Recall, and F-1 Score of the assembled model on the test data.

taken from Yang et al. (2013); Guo et al. (2021); Hu et al. (2020) which implement random forest, BERT, and graph neural networks respectively.

Our results show that the large language models even without fine-tuning and relying only on a small number of in-context examples can perform on par with the traditional methods which require more human intensive effort to carefully craft and engineer selected features. We also demonstrate how fine-tuned large language models prove to be much superior to conventional models and showcase clearly improved performance on the twitter dataset.

Few of the areas where large language models demonstrated their best performance are:

- *Dealing with imbalanced data.* Using large language models, we were able to drastically improve the overall recall, thus flagging more malicious accounts.

- *Leveraging relational data well.* We observed from our experiments that large language models outperformed GNN architectures. This is specially impressive since the latter are especially designed for handling structural data such as social media networks.

We also encountered some unexpected results:

- *Bias in textual data.* The amount of textual data in twibot-22 dataset scales to become a massive library with billions of rows with only a fraction being from bot accounts. This skewed the results we obtained for generating similar tweets to a test instance, since the sampling was biased towards human tweets, thereby degrading the performance. Pursuing measures to rectify the imbalance such as forcing a strict ratio of bot to human samples mitigated this to some extent, but this approach is impractical in a live system. Further investigation needs to be carried out for a scalable solution.

7 Conclusion

We conclude that the use of LLMs for identifying bots on social media is a promising domain, although more research is merited. LLMs have the potential to be the new frontier for bot detection, and we argue that they are the most promising candidate in further bot detection research since malicious agents are already employing LLMs to supercharge their bots and make them harder to detect (Feng et al., 2024).

The extensive experiments carried out in this work effectively demonstrate that LLMs can outstrip traditional bot detection techniques, and are specially performative when given a small subset of data where traditional methods produce exceedingly poor results.

Future Work

There are multiple ways in which this work can be carried forward, that we were not able to pursue due to time or computational constraints.

Due to limited time and computational resources, not much effort could be put into hyperparameter tuning and experimentation under different settings. An in-depth analysis of the trade off between the training resource and performance could give insights into more optimized experimental setups.

Another promising avenue is the use of reinforcement learning using DPO in order to better genrealize the model and prepare it for adversarial attacks (Yang et al., 2024). This would be significant since developing a robust model is essential in order to ensure your solution is sustainable when faced by ever-evolving evasive strategies employed by bots.

Further, more work can be done on the supervised fine-tuning domain by generating an even larger or more sophisticated dataset, and using that to train the LLM. This would likely have the intended effect of boosting LLM performance.

Another domain that can be explored is investigating the fairness of LLM based bot detectors by carefully evaluating their results. In other words, exploring whether LLMs are more likely to mislabel certain kinds of users over others, possible based on inherent social biases. See Ethics Statement below for more details.

Lastly, while our research is focused on twitter due to the availability of dataset and abundant prior literature, the approach and experiments performed could be translated to other social media platforms as well. An interesting yet challenging area of research could be the development of automated pipelines to distinguish between bots and humans which are independent of the platform being tested.

Ethics Statement

We would like to acknowledge that large language models are widely understood to have inherent sociopolitical biases and such biases can have an unintended or adverse impact on the downstream tasks (Blodgett et al., 2020; Shaikh et al., 2023). It would follow reason that social media bot detection would also suffer from the same issues that plague LLMs in other tasks. LLMs are highly susceptible to making decisions based on outdated stereotypes, social biases, and even spurious correlations. As a result, we would like to state clearly that we do not believe that LLMs are a one-stop solution for content moderation on social media platforms. Despite their obvious prowess and clear potential, it is imperative that in actual applications at scale, LLM based bot detectors make decisions with human content moderators in the loop in order to mitigate the danger of any unintended biases affecting real human users.

Acknowledgments

We extend our heartfelt gratitude to TA Sayan Ghosh and Professor Swabha Swayamdipta for their invaluable guidance, support, and insightful feedback throughout the course of this work. Their mentorship greatly contributed to shaping the direction and rigor of our research.

We also acknowledge the inspiration drawn from the research provided by the team behind the paper "What Does the Bot Say? Opportunities and Risks of Large Language Models in Social Media Bot Detection" (Feng et al., 2024). Their work, as well as their publicly available dataset and repository, served as a foundation and motivation for our work.

References

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Thi Bui and Katherina Potika. 2022. Twitter bot detection using social network analysis. In *Proceedings of the 2022 Fourth International Conference on Transdisciplinary AI (TransAI)*.

Zijian Cai, Zhaoxuan Tan, Zhenyu Lei, Zifeng Zhu, Hongrui Wang, Qinghua Zheng, and Minnan Luo. 2024. Lmbot: Distilling graph knowledge into language model for graph-less deployment in twitter bot detection. In *arXiv preprint arXiv:2306.17408*.

David Dukić, Dominik Keča, and Dominik Stipić. 2020. Are you human? detecting bots on twitter using bert. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 631–636. IEEE.

Philip G. Efthimion, Sophia C. Payne, and Nicholas Proferes. 2018. Supervised machine learning bot detection techniques to identify social twitter bots. *SMU Data Science Review*, 1(2):5.

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, Xinshun Feng, Qingyue Zhang, Hongrui Wang, Yuhan Liu, Yuyang Bai, Heng Wang, Zijian Cai, Yanbo Wang, Lijing Zheng, Zihan Ma, Jundong Li, and Minnan Luo. 2023. Twibot-22: Towards graph-based twitter bot detection. *arXiv preprint arXiv:2206.04564*.

Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021. Twibot-20: A comprehensive twitter bot detection benchmark. ArXiv preprint arXiv:2106.13088.

Shangbin Feng, Herun Wan, Ningnan Wang, Zhaoxuan Tan, Minnan Luo, and Yulia Tsvetkov. 2024. What does the bot say? opportunities and risks of large language models in social media bot detection. *Preprint*, arXiv:2402.00371.

Qiang Guo, Hao Xie, Yifan Li, Wenjin Ma, and Chunyan Zhang. 2021. Social bots detection via fusing bert and graph convolutional networks. *Symmetry*, 14(1):30.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710.

Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *arXiv preprint arXiv:1802.04289*.

Yuhan Liu, Zhaoxuan Tan, Heng Wang, Shangbin Feng, Qinghua Zheng, and Minnan Luo. 2023. Botmoe: Twitter bot detection with community-aware mixtures of modal-specific experts. *arXiv preprint arXiv:2304.06280*.

Yi-Ju Lu and Cheng-Te Li. 2020. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514. Association for Computational Linguistics.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Lynnette Hui Xian Ng, Iain J. Cruickshank, and Kathleen M. Carley. 2022. Cross-platform information spread during the january 6th capitol riots. *Social Network Analysis and Mining*, 12(1):133.

Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, and Diyi Yang. 2023. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4454–4470, Toronto, Canada. Association for Computational Linguistics.

Chao Yang, Robert Harkreader, and Guofei Gu. 2013. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 8(8):1280–1293.

Zeyu Yang, Zhao Meng, Xiaocheng Zheng, and Roger Wattenhofer. 2024. Assessing adversarial robustness of large language models: An empirical study. *arXiv preprint arXiv:2405.02764*.

A Appendix

A.1 Dataset Features

Split Category	Count of samples
Train Set	700,000
Test Set	100,000
Val Set	200,000

Table 3: Data Split Ratio for TwiBot-22 dataset

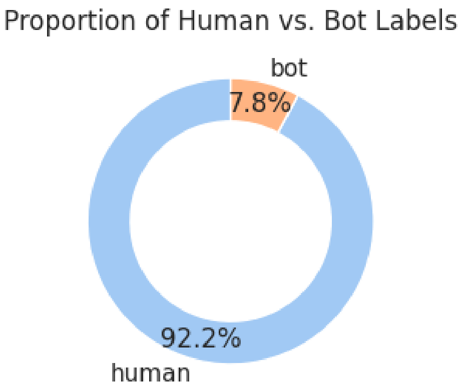
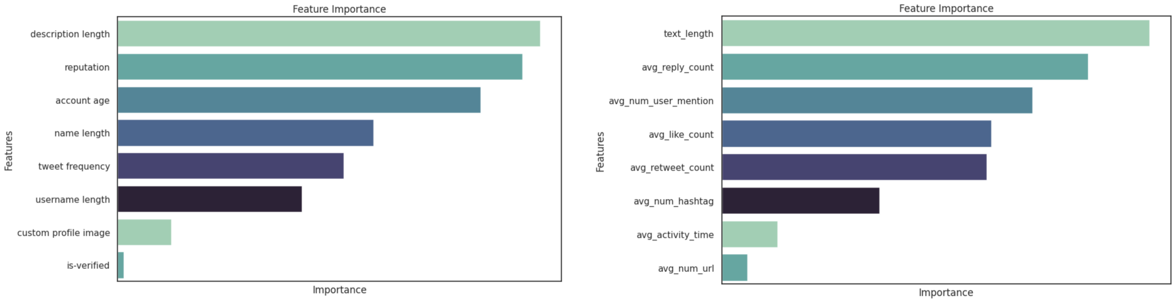


Figure 2: Distribution of Labels in Training Data

A.2 Engineered Features

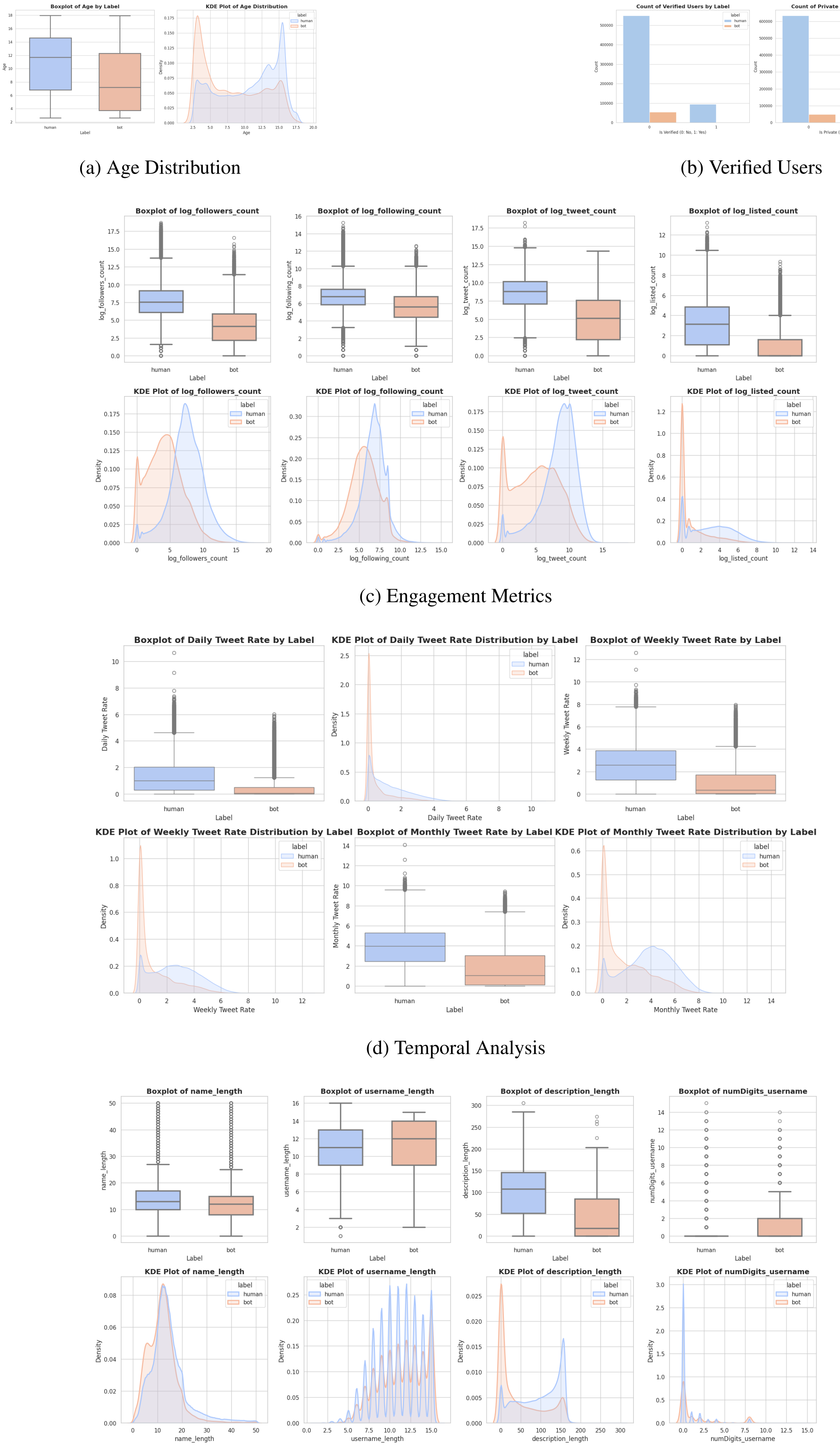


(a) Meta-Data Features

(b) Tweet-Data Features

Figure 3: Relative Importance of Engineered Features

A.3 Feature Visualizations



(a) Age Distribution

(b) Verified Users

(c) Engagement Metrics

(d) Temporal Analysis

(e) Metadata Overview

Figure 4: Distribution of User Meta Data

A.4 Tweet Dataset Breakdown

Field Name	Format	Example
attachments	JSON / NaN	NaN
author_id	String	"30977232"
context_annotations	JSON / NaN	NaN
conversation_id	String	"1493829586240610305"
created_at	Datetime (ISO 8601, UTC)	"2022-02-16 06:08:22+00:00"
entities	JSON	{’hashtags’: [], ’user_mentions’: [...] }
geo	JSON / NaN	NaN
id	String	"t1493829586240610305"
in_reply_to_user_id	String / NaN	NaN
lang	String (ISO 639-1 code)	"en"
possibly_sensitive	Boolean (True/False)	False
public_metrics	JSON	{’retweet_count’: 1, ’reply_count’: None, ... }
referenced_tweets	JSON / NaN	NaN
reply_settings	String	"everyone"
source	String (HTML)	"<a href=ḧttp://twitter.com/download/iphone ..."
text	String	"RT @SimmyP1: @young_urbanists @ofuturecities ..."
withheld	JSON / NaN	NaN

Table 4: Description of the (raw) fields in the TwiBot-22 tweet dataset.