Bot Detective - Technical Report

Stav Ben-Tov
Tel-Aviv University
stavbentov5@gmail.com

Tamir Davidi
Tel-Aviv University
tdavidi10@gmail.com

ABSTRACT

Social bots pose security, political, economic, and social risks to our lives. To effectively address these risks, it's essential for users exposed to their influence to acknowledge them and have the ability to detect them. *Bot Detective*[1] is a Chrome extension designed to increase awareness about social bots, protect user privacy, and mitigate the effects and risks associated with them. The extension combines the Twitter API with a Random Forest model we developed based on a combination of classified datasets. Our goal was to provide a practical and valuable tool for Twitter users by making the model accessible to them.

In this paper, we discuss the risks associated with social bots and the importance of raising awareness about them, which motivated the development of this plugin. Additionally, we outline the development process of *Bot Detective*, detailing the tools we utilized and the design choices we made to maximize efficiency and effectiveness. While acknowledging the limitations we faced, we also highlight potential avenues for future improvements and expansions of *Bot Detective*.

1 INTRODUCTION

In recent years, the utilization of social media among users around the world has seen a marked increase, along with its influence on them. To quantify this, Twitter has 166 million daily active users as of 2020, a 24% increase from 2019 [7].

With this increase, there has also been a growth in prevalence of social bots on social media platforms. However, not all users are aware of their existence and the potential impact they can have. A social bot is a computer algorithm that automatically produces content and interacts with humans on social media, trying to emulate and possibly alter their behavior [5]. According to recent studies, between 9% and 15% of active Twitter accounts are bot accounts [12], a report showed that in 2022, nearly half (47.4%) of all internet traffic originated from bots [14]. Another study states that social bots are responsible for generating 35% of the content that is posted on Twitter [4]. This is a substantial proportion compared to the fact that there might be users that assume the accounts they encounter on social medias are human being.

It's important to recognize that there are various types of bots, and not all of them have malicious intentions. For example, there are bots that post information, spread news or academic papers.

Social bots have various impacts on us; among them are security, economic, political and social risks. From a political perspective, we can observe their influence, as seen in the case of the 2016 U.S. elections. It is plausible that entities like the Internet Research Agency exploited existing ideological differences to exacerbate divisions and promote specific political narratives, favoring Donald Trump while undermining support for Hillary Clinton using bots. These bots disseminated false information and divisive content and might have influenced the election results [3, 9]. Turning to the economic impact, in 2014, bots were found to be responsible for spreading identical information about the Cynk company, which caused automatic trading algorithms to notice this, leading to a staggering 200-fold surge in market value, propelling the company's valuation to $5 billion. However, trading was halted once analysts recognized this phenomenon, resulting in financial losses [5, 15]. From a security and privacy perspective, recent study has revealed the existence of a network of bots that generate content with the aim of promoting suspicious websites and spreading harmful content. These bots also engage in the steal of selfies to create fake personas. They attempt to persuade people to invest in fraudulent cryptocurrencies and have even been suspected of stealing from existing crypto wallets [13].

All these factors have the potential to negatively affect our lives. For instance, there's a risk of foreign interference in our government's selection, where decisions may align more with external desires than the needs of our people. Furthermore, financial losses can result from a variety of causes by bots, such as becoming victim to bot-spread disinformation or disclosing personal information to scam bots. From another perspective, the propagation of misinformation can also contribute to exacerbating divisions within a country. This includes the dissemination of extreme opinions, fostering deep dichotomies, and promoting discord among our citizens.

Users are exposed to the influence of bots, effectively becoming tools in the hands of those who exploit bots to achieve their goals, manipulating users and affecting their behavior and opinions. Despite this, awareness does not necessarily align with the extent of impact that bots have on our lives.

A research study [10] involving 297 users highlighted the correlation between the perception of bots and classification accuracy. The study indicated that while some participants were aware of bots' main attributes, others lacked knowledge about bots, had an abstract understanding of the concept, or confused bots with other phenomena. Participants also faced challenges in accurately classifying accounts and were more prone to misclassifying bots compared to non-bots.

We are convinced that as users become capable of identifying bot accounts, the adverse effects they bring on our lives will diminish and the potential to spread misinformation and malicious content will be reduced. Our objective is to raise awareness about bots on Twitter and help people identify them. As indicated in a study [10], participants who lacked knowledge about bot intentions and held vague perceptions of them were more prone to misclassifying accounts. By ensuring users are informed about the presence and potential risks posed by bots, we can get a more secure and reliable

[1] Please refer to our project repository: https://github.com/stav-bentov/Twitter-Bot-Detector.git

online environment. Ultimately, this contributes to the overall safety of our lives.

We have replicated a model presented in the article titled "Scalable and Generalizable Social Bot Detection through Data Selection" [8] , given a username and his metadata return classification of bot or human, and developed a Chrome extension named *Bot Detective*, that operates in conjunction with the model and Twitter API.

Bot Detective features two primary functionalities. The primary functionality involves displaying a "Bot" or "Human" indicator beside each username in the Twitter feed. As users browse through their Twitter feed, they will seamlessly come across visual indicators displaying "Bot" or "Human" sign—generated by the ongoing classification process of the model. This real-time classification occurs effortlessly as the user scrolls through their feed, providing an interactive experience that assists users in determining the nature of encountered accounts. Each indicator is accompanied by a popup with warning and classification accuracy, in addition clicking can lead to a popup that provides educational content about the impact of bots. This popup contains articles that highlight the risks posed by bots.

The second feature we have developed involves analyzing a random selection of 100 followers for a user and presenting the percentage of bots among them. This feature aims to demonstrate that sometimes not all followers are authentic humans, and a user's popularity, often measured by follower count, may not accurately reflect their true influence.

The purpose of *Bot Detective* is to enhance user awareness regarding the existence and potential hazards posed by bots while minimize their harmful effects and risks. By expediting the capability to classify users as bots or not, we aim for users to adopt a discerning and cautious mindset, thoughtfully assessing the opinions they come across, carefully evaluating the data they share, and maintaining vigilance to protect personal information.

2 RELATED WORK

Considerable efforts have been dedicated to the development of tools for bot detection on social media platforms. Bots are getting better at imitating humans; they are trying to avoid detection which makes it much harder to detect.

2.1 Awareness and Classification Tools

A study on user awareness of bots [10] revealed that individuals with abstract perceptions and poor understanding of bot intentions were more likely to misclassify accounts as bot/human. Thus, by enhancing awareness and adding a classification tool we can get users to improve their ability to independently identify bots.

The evolving field of bot identification technologies includes important contributions. Botometer [1], for example, is a well-known bot detection tool that assesses Twitter user activity and gives a score through its website by entering a username. Unfortunately, as for today this tool is not available due to some decryptions of Twitter API endpoint. Another known tool is BotSight [2], a browser plugin that enhances the user experience by including signs that indicate the chance of an account being a bot or human. This product also as for these days is not available.

2.2 Machine Learning Approach

The ongoing efforts involve the creation of detection models and the assembly of datasets that categorize user information to bot/human. Unsupervised models have been developed, such as the one presented in [11], which utilizes a hashing technique to cluster user accounts into correlated sets in near real-time. This method detects thousands of bots daily with a 94% precision rate and generates online reports regularly. An advancement also lies in the application of supervised learning, relying on classified datasets. There are popular techniques that leverage user, temporal, content, and social network features such as Random Forest classifiers and Logistic regression. There are also developments that are based on Neural Networks or Natural Language Processing (NLP) approaches [11, 16].

In the article titled "Scalable and Generalizable Social Bot Detection through Data Selection" [8], which served as the basis for our work, a Random Forest model was suggested. This model proposes the utilization of minimal account metadata to enable efficient analysis while addressing the challenges of scalability and generalization. In this study the researchers aimed to identify the optimal model by investigating 11 datasets.

The article gives a thorough overview of the research that was conducted to find the best bot classification model, M196 according to the article.

First, the researchers chose 20 features. They were interested in how separability and generalizability related to one another. In order to investigate separability, they evaluated the scores and homogeneity of each dataset, finding that 5 out of 11 showed a clear clustered structure, suggesting that the selected characteristics effectively distinguish between bots and humans. To understand aspect of generalizability, the researchers used Random Forest models that were trained on one dataset and tested on another to examine the consistency of human and bot categorization across other datasets. This analysis revealed that no single dataset could generalize well to all others.

They found no clear correlation between separability and generalizability. Even while bots can be quickly identified in one dataset, that does not guarantee that a classifier trained on that dataset will be successful in identifying bots in other datasets.

The researchers aimed to enhance three key parameters: cross-validation accuracy on training data, generalization to unseen data, and consistency with a more feature-rich classifier on unlabeled data. By creating six tests they ranked various models using different datasets and identified the top three performers. Among these, the M196 model, which utilizes five datasets, emerged as the best fit according to the specified criteria. This model will be further elaborated in subsequent sections of the article, as we have chosen to develop and employ it for our *Bot Detective* extension.

We aimed to make the model accessible and valuable for Twitter users. To achieve this, we integrated the model's output directly into the user's browsing experience by adding bot/human indicators. This use of the model allows users to discern the nature of accounts they encounter while browsing the platform.

3 METHODOLOGY

Bot Detective is a Chrome extension that works with Random Forest ML model and Twitter API. In terms of optimizing usability metrics, it was important for us to ensure that our extension is as efficient and effective as possible. Our goal was dual: to provide a quick response without interfering with the browsing experience and to achieve high accuracy of our model. With respect to security and privacy (S&P), we wanted to enhance the bot's threat awareness while minimizing intrusion into users' privacy.

The design, techniques, and tools we used were important in attaining these objectives.

3.1 Machine Learning Approach

The development of *Bot Detective* began with the construction of the model. We replicated the recommended M196 model from [8], which utilizes five classified datasets of bots and human account on Twitter and their corresponding metadata. In the initial stages, we collected these datasets, managing to obtain and employ four of them[2]: cresci-17, celebrity, bototmeter-feedback, and political-bots. From these datasets, we extracted 20 features which consist of two types of features: metadata features that we use directly, and derived features calculated from metadata. The metadata features include the number of tweets (including retweets) issued by the user, the number of current followers, the number of users being followed, the count of liked Tweets over the account's lifetime, the number of public lists the user is a member of, whether the user has kept the default theme or background for their user profile, whether the user has retained the default profile image, whether the user has a verified account. The derived features encompass evaluating the correlation of each metric: the number of tweets, followers, following, and public lists liked, with respect to the time difference between the probe time and account creation, length of username, number of digits in username, description's length and screen name likelihood which is the geometric-mean likelihood of all bigrams in it. For calculating derived features, additional metadata such as name, username, time of user creation, and description were required. The calculations required for the features needed to be handled carefully, with precautions such as avoiding division by zero and addressing empty values. By extracting these relevant features, we constructed a dataset containing 11,094 users classified as human and 9,772 as bots, resulting in a relatively balanced dataset.

Using this united dataset we created a Random Forest model with 200 decision trees, each with a maximum depth of 5 levels, and the random seed for generating randomness within the model is set to 1. These parameters were chosen via a grid search with a range of values and achieved the best score of about 0.96 according to 5 cross validation tests. For comparison, the model in the article reached an accuracy percentage of 0.98.

3.2 Technical Approach

The implementation of the *Bot Detective* extension involved the development of client and server sides communicate with secured https requests. The client side was developed using JavaScript, leveraging the efficient memory capabilities of local storage. On the server side, we deployed a Google Cloud server, implemented in Python using the FastAPI. This server functions as an intermediary between the extension and the model. To manage memory storage on this side, we adopted Redis as a cache mechanism. This approach enhances efficiency and responsiveness in serving user requests.

3.2.1 2 Two-Layer Caching Mechanism. *Bot Detective* works by obtaining information from the server for each user. Requesting data from the server and performing model calculations every time can be time-consuming. To address this, we've implemented a two-layer caching mechanism. This approach optimizes the overall efficiency of the extension while also mitigating the strain on the Twitter API, which has limitations on the frequency of requests that can be made.

The first layer is the local storage; a feature that enables JavaScript websites and applications to store key-value pairs in a web browser without an expiration date, it stores the results returned by the server. This layer operates on a per-user basis. It minimizes unnecessary server access for recently computed data, ensuring rapid data retrieval and manipulation at the user level, contributing to a seamless user experience.

The second layer is located on the server and utilizes Redis storage. This layer stores results across all users , allowing for the possibility that one user's calculation results can be used by another user, thus saving computation time, Twitter API accesses and model execution.

3.2.2 Data Refresh. Considering that user behavior, followers or following counts, and activity rate can change over time, impacting the metadata upon which the model's classification is based, we have implemented an "expiration date". Cached results are stored for 30/10 days (depends on saved information), after which new calculations are required. This approach ensures that the information remains as up to date as possible while still benefiting from the cache.

3.2.3 Request Aggregation. Another strategy we adopted involved consolidating server requests (Figure 1). Due to the limitations of the Twitter API for the academic user account we utilized, certain endpoints had restrictions that could potentially stall the extension if exceeded. To maximize the efficiency and utility of each request, we employ a method of merging requests at the server level across all users, utilizing requests queue and specific techniques. We then initiate requests at regular intervals. By employing this approach, we not only save requests but also minimize the risk of surpassing the API's request limitations and thus ensure a seamless user experience without causing disruptions and delays.

3.2.4 Bot/ Human Sign Feature. One of *Bot Detective*'s features is the addition of bot/human indicators alongside each @username instance within a user's feed while browsing (Figure 2). This implementation is designed to be as unobtrusive as possible, ensuring a seamless user experience. Additionally, we've implemented hover popups linked to each indicator, which serves a dual purpose: it offers users a cautionary message and provides insights into the accuracy and confidence of the classification. The caution popup was implemented to avoid overwhelming the user with excessive information on the page, which could potentially disrupt their browsing

[2] The Varol-icwsm dataset lacked metadata and only included classification, rendering it unsuitable for our purposes

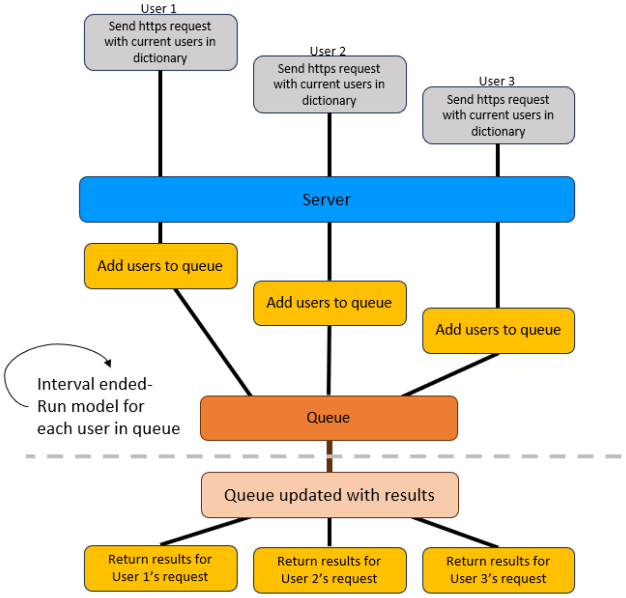
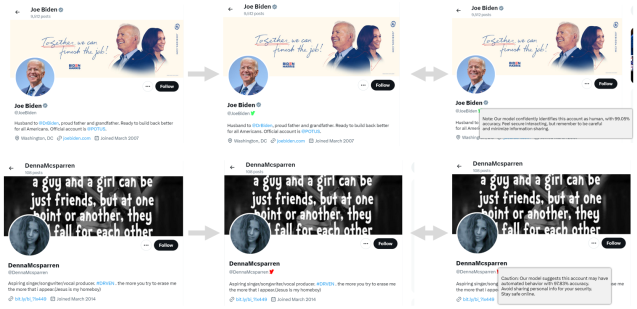

User 1
Send https request with current users in dictionary
User 2
Send https request with current users in dictionary
User 3
Send https request with current users in dictionary
Server
Add users to queue
Add users to queue
Add users to queue
Interval ended- Run model for each user in queue
Queue
Queue updated with results
Return results for User 1's request
Return results for User 2's request
Return results for User 3's request

Figure 1: Request Aggregation

Initially, User 1 sends a request, followed by requests from User 2 and User 3. The server aggregates all incoming requests, and at specified time intervals, it processes them collectively and dequeues the requests to provide responses to each of them.

experience. The alert serves as a reminder to users to exercise caution and make them understand the classification if needed. We recognize that achieving 100% accuracy in any classification system, including our model, is unattainable, hence, we are committed to ensuring our users are well-informed about this limitation. Additionally, the information on accuracy empowers users to make informed decisions. For instance, if a user sees that a user has been classified as bot with low accuracy, they can critically assess the classification and question its validity.



Note: Our model confidently identifies this account as human, with 99.05% accuracy. Feel secure interacting, but remember to be careful and minimize information sharing.

Caution: Our model suggests this account may have automated behavior with 97.83% accuracy. Avoid sharing personal info for your security. Stay safe online.

Figure 2: Indicator and popups in Bot/Human cases

At the beginning of our work, we initially considered solely adding a bot sign when the model classified an account as bot. However, following thinking and gathering feedback from 15 individuals aged 20 to 55 who used the extension, we presented them with the option of displaying both human and bot signs or just the bot sign. The consensus was in favor of displaying both signs. This approach improves the extension's responsiveness, as when only the bot sign was added, users felt that the extension might not be functioning properly when no bot signs were visible. Furthermore, we recognized the importance of addressing situations where a user is classified as human with lower accuracy. This setup allows users to approach interactions with these accounts with a sense of awareness, because now they know that a user is human with certain accuracy.

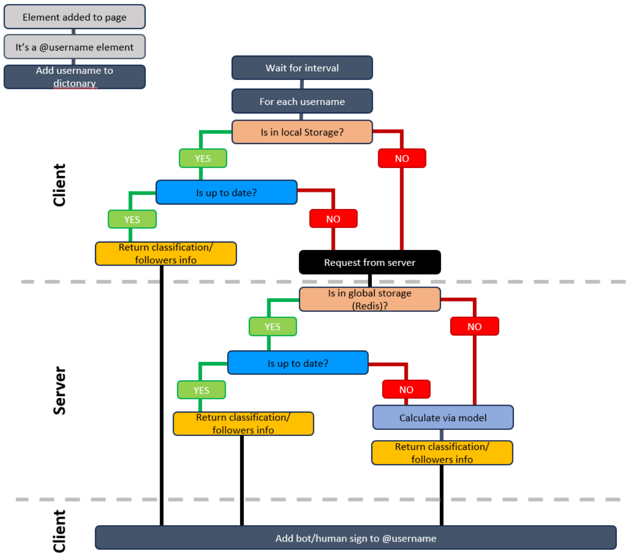


Figure 3: Data Flow

When browsing a Twitter page, a script actively monitors for changes and new elements being added. As described in high level in Figure 3, whenever an element containing a "@username" is detected, "username" is added to a dictionary. At intervals, this dictionary is examined. On client side we check whether the classification and accuracy for the given username are stored in the local storage and whether they are up to date. If so, a corresponding sign is added. If not, the server is consulted. Initially, on server side we check if the username's result is already stored in Redis. If it is, the result is sent back to the client. If not, the username is added to a queue along with other usernames from previous requests. After a set interval, the username is processed by our model. The resulting classification is then saved on Redis storage and sent to the client side, who saves the result in local storage and takes care of adding the appropriate sign and information to the user interface.

3.2.5 Bots in Followers Feature. As part of our goal to raise awareness and promote a more cautious and discerning approach, we have developed an additional feature to *Bot Detective*. This feature displays in each profile page the number of bots among a random sample of 100 followers with a popup that explain that these followers were taken randomly. Given that users frequently assess popularity based on follower counts, it's quite common for some accounts to buy followers or utilize bots to give the appearance of influence [6]. This feature exposes the potential prevalence of bots in accounts, encourages users to question the authenticity and reliability of online content. It prompts users to consider the possibility that not all accounts with high follower counts are genuinely influential and encourage them to engage with content more skeptically.

We opted to analyze 100 followers due to limitations in the Twitter API's capabilities. Examining all followers or a percentage of them might exceed the limit of the API and also result in slower processing, particularly for users with a substantial follower's amount. Moreover, this approach aligns well with the limitation of processing 100 users per endpoint for user metadata.

The process of incorporating this additional information follows a similar pattern to that of adding the bot/human sign, leveraging the two-layer storage approach. However, there is a one distinction, this action is initiated when there is a change in the URL, as opposed to being triggered by the examination of newly added elements on the page. This information is specifically integrated when navigating a profile page. Furthermore, we make use of the results of the analyzed followers by saving their classification data in Redis. This method optimizes the whole process by increasing the efficiency of their following requests.

3.2.6 Information Popup. Our focus extended beyond merely notifying users about the presence of bots and warning them. We also aimed to enrich their understanding by including a popup feature when clicking the bot or human sign that offers essential information and news about bots (Figure 12). This approach enables users to gain a more profound insight into the subject and access reliable sources of knowledge.

4 RESULTS

Our goal was to optimize our evaluation metrics: cross-validation accuracy on training data and generalization to unseen data. As we know, better cross-validation accuracy leads to enhanced generalization and predictive reliability on new data.

Using the united dataset we constructed, we conducted a grid search involving the linear Support Vector Classifier (SVC), Gaussian Naive Bayes and Random Forest Classifier. This involved exploring various parameter combinations. The highest accuracy scores achieved were as follows: Gaussian Naive Bayes reached 77.5%, Linear SVC achieved 90.8%, and the best performance of 96.2% was attained by the Random Forest Classifier with parameter settings of 200 decision trees, each limited to a maximum depth of 5 levels, and a random seed of 1 for internal model randomness generation. This optimal Random Forest model, incorporating the mentioned configuration, was subsequently selected as our final choice for the project.

We regard these results as significant achievements, as it closely aligns with the results presented in the article that guided us [8], which achieved a score of 98%.

4.1 Model Overview

The following visualizations showcase important features and give an overview of the model's overall performance.

The Importance score quantifies the impact of each feature on the model's decision-making process (Figure 4). A higher score indicates a stronger influence on the model's predictions. The presented results highlight specific features that play a significant position in the classification process. For instance, features such as the growth rate of likes given by user (favourites_growth_rate), the total number of likes given by user (favourites_count), and the total number of tweets posted (statuses_count) exhibit relatively higher importance scores. Conversely, features with lower scores, such as 'screen_name_length', 'num_digits_in_screen_name', and 'profile_use_background_image', hold less significant influence. Importantly, these results are in alignment with the Feature Importance findings from the referenced article [8].

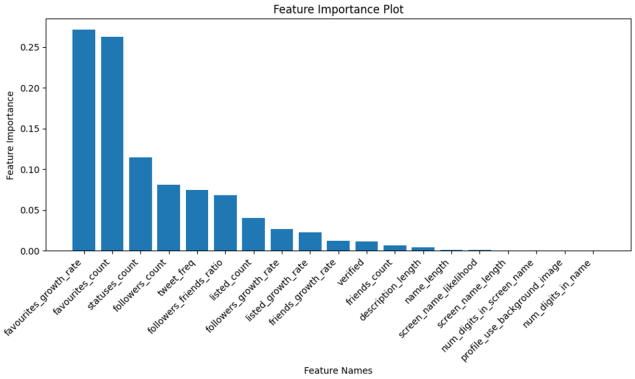


Figure 4: Feature Importance

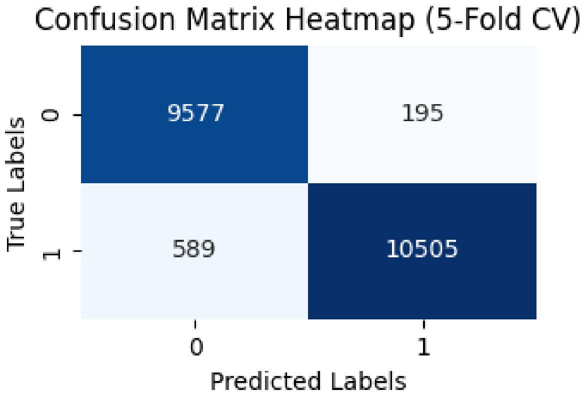


Figure 5: Confusion Matrix Heatmap

The obtained confusion matrix results (Figure 5) illustrate the strong performance of our model. With a total of 9577 true negatives and 10505 true positives, our model demonstrated its high accuracy ability to classify instances into their respective categories. The presence of only 195 false positives and 589 false negatives further underscores its effectiveness. These results reflect a well-balanced and reliable performance and highlight the model's proficiency in making correct predictions and minimizing misclassifications.

5 DISCUSSION

5.1 Limitations

A main limitation on our work was the limitation request on Twitter API.

It's important to note that the Twitter API is capable of a maximum of 900 User Lookup API requests within a 15-minute time frame, with each request can include up to 100 users. Consequently, we were able to evaluate the bot/human indication of up to 90000 users every 15 minutes. Exceeding this threshold triggers a rate limit exception, resulting in a mandatory waiting period before further API requests can be made. This can occur when many clients are using *Bot Detective*. Furthermore, for the number of bots in followers we used endpoint of Twitter API v1.1.[3] This endpoint also has its limitation requests and the need to take metadata of each follower also restricts it.

To mitigate the risk of encountering such rate limits, we have employed various approaches to enhance request utilization as described in the Methodology section.

Without this limitation, faster responses could have been achieved, and the feature that analyses 100 randomly chosen followers could have been adjusted to cover all followers. This broader approach of analyzing all followers would have offered a more accurate insight into the presence of bots and their influence on the user's social media environment. Additionally, we could have introduced new features without restrictions, such as analyzing the number of bots in likes, replies, and reposts. While the current limitations prevent us from adding these features, as the increased workload of getting large amount of user's metadata could trigger rate limits due to the substantial number of users requiring analysis.

5.2 Future Work

5.2.1 Improving Bot Detective. Twitter has officially announced the deprecation of certain v1.1 endpoints. As a result, the transition from Twitter API v1.1, which we now utilize for API calls, to v2 became essential for the development of the extension.

Adopting v2 opens opportunities to introduce new features, as previously mentioned, such as analyzing the presence of bots in likes, reposts, and replies of a tweet.

Another potential direction for future development that could extend *Bot Detective*'s capabilities is the integration of NLP-based toxicity assessment for tweets. Highly toxic comments containing offensive language or harmful content can be flagged. These assessments could be displayed alongside tweets and used to evaluate user's tweets, with the information being provided on their profile. Users may utilize this feature to make better decisions about interaction with certain accounts and reading their content. Understanding that certain individuals promote harmful and poisonous information that may harm other communities allows users to choose to avoid such accounts. Prevention of interaction with accounts that promote offensive content might help to improve reality and reduce the dissemination of negative and divisive opinions.

Additionally, we could enhance the model by incorporating features that assess the toxicity of tweets. Malicious social bots often spread misinformation and divisive content that could be categorized as toxic. This addition could be valuable to our model.

Such improvements could further enrich the project, enabling users to gain deeper insights into other users and encouraging more cautious and informed interactions.

5.2.2 User Study. We decided to conduct the user study protocol by an online survey to evaluate the effectiveness of our extension and examine our hypothesis.

Our hypothesis suggests that individuals using *Bot Detective* will exhibit increased awareness of security and the presence of bots. Additionally, they are likely to adopt a more cautious and critical approach to the content they read on Twitter and their detection ability will be improved.

To assess the effectiveness and impact of *Bot Detective*, we split the survey into two parts. One part will be asked before users begin using the extension, while the second will be conducted after two months of use. This approach enables us to compare changes in user behavior and perspectives. The survey includes questions to assess user familiarity with bots and their threats, their ability to detect bots, and their overall usability experience over the span of two months.

5.3 Exploring Bot Influence: Analysis of Bot Followers Among Knesset Members

The implications of our project extend beyond the development of a Chrome extension. In this section, we explore potential avenues for future research using our model.

Bots are well-known for pushing one-sided content, magnifying extremist viewpoints, and spreading misinformation. We occasionally see their influence, like with Sync stock and the 2016 US elections. Like every other country with social media exposure, Israeli users are also exposed to bot activities, which can have an impact on them, especially in this currently political tension.

The amount of followers can affect the promotion and spread of account's content. Since follower amount is often used as a popularity measure and purchasing bot followers is a common practice, we found it interesting exploring the bot percentage of the followers of Knesset members.[4]

Due to limitations of Twitter API, we explored only 10% of followers, randomly selected for each account of Knesset member. For analyzing the data, we decided to display it in several variations and comparisons, we will talk about one of them (Figure 6).

In figure 6 we can see that according to our analysis, both the Coalition and Opposition members have relatively similar bot percentages, with the Coalition at around 31.55% and the opposition at around 31.16%. This similarity indicates that the presence of bots is not significantly biased towards any political orientation, underscoring the possibility of bots being widespread across the entire political spectrum. Another interesting point is that it appears that bots have a high presence among Knesset followers, accounting for

[3] followers endpoint is now unavailable.

[4] Notably, several Knesset members and ministers, including Yair Levin, Avraham Bezalel, Erez Meloul, Yonatan Mishricki, Yinon Azoulay, Moshe Arbel, Simon Moshiashvili, Moshe Rot and Yasser Hujirat do not possess Twitter accounts. Additionally, Avi Maoz, who lacks a Twitter account, had their party's account analyzed, given that he is the sole representative of his party.

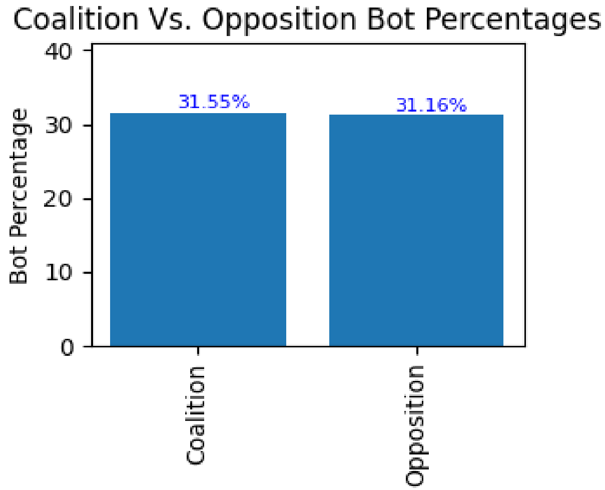


Figure 6: Coalition-Opposition Bot Percentage
These percentages were determined by calculating the average bot followers' proportion among all Coalition/Opposition Knesset members.

about 30%, which is higher than their typical frequency on Twitter, which is around 15%.

Another results are in the Appendices.

6 CONCLUSION

In conclusion, *Bot Detective* Chrome extension constitutes a step towards countering the challenge of bot influence on Twitter. Through a combination of machine learning and real-time data analysis, the extension empowers users with the ability to distinguish authentic accounts from bots.

In addition to its bot/human indicator feature, we've developed a feature that calculates the number of bots in 100 randomly selected followers of a Twitter user. The information is displayed on every profile page. These two features assist with achieving secure surfing and raise the awareness about the presence of bots on Twitter.

To amplify *Bot Detective*'s effectiveness, we've created an informative popup that educates users about the threats posed by bots. This feature directs users to reliable resources, enhancing their understanding of the matter.

We explained about the tools and approaches we employed to optimize the extension's effectiveness, efficiency, and user-friendliness. Moreover, we've considered the extension's limitations and discussed potential future upgrades.

7 CONTRIBUTIONS

Our collaboration was built on joint decision-making, task assignment, and equitable distribution of responsibilities. Both of us actively participated in the coding process, with each piece of code undergoing review and correction by the other team member as needed.

Both of us contributed to the construction of the unified database, the extraction and calculation of pertinent data, and the evaluation of the optimal classifier for our model by exploring various classifier options. Each of us was involved in both aspects: the server side and the client side.

On the server side, Tamir took charge of tasks encompassing request consolidation for enhancing performance, Google Cloud Server setup and configuration, and utilization of Redis. Stav, on the other hand, was responsible for calculating and generating responses to requests while considering the limitations imposed by the Twitter API.

On the client side, Stav was in charge of adding and designing appropriate indicators and popups, creating an informative popup, adding the feature of information about the number of bots in the followers, and handling local storage.

We worked on implementing HTTPS communication between the extension and the server. Additionally, Tamir was responsible for generating the essential server.crt certificate file and the server.key private key file, enabling HTTPS through the TLS protocol, thus enhancing the system's security.

We also thought about ways to explore followers of Knesset members, and Tamir conducted the required analysis by creating the visualization. At the end of our work on *Bot Detective*, we built an online survey for the user study procedure.

REFERENCES

[1] [n. d.]. *Botometer.* https://botometer.osome.iu.edu/

[2] [n. d.]. *BotSight.* https://download.botsight.nlok-research.me/#about

[3] Will J. Grant Patrick L. Warren Darren L. Linvill, Brandon C. Boatwright. 2019. *THE RUSSIANS ARE HACKING MY BRAIN!" investigating Russia's internet research agency twitter tactics during the 2016 United States presidential campaign.* https://www.sciencedirect.com/science/article/abs/pii/S074756321930202X?fr=RR-2&ref=pdf_download&rr=7fe5be4be81b8e45

[4] Elfadil A. Mohamed Hany Alashwal Eiman Alothali, Nazar Zaki. 2019. *Detecting Social Bots on Twitter: A Literature Review.* https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8605995

[5] Calyton Devis Filoppo Menczer Alessandro Flammini Emilio Ferara, Onur Varol. 2016. *The Rise of Social Bots.* https://dl.acm.org/doi/pdf/10.1145/2818717

[6] Christopher Kruegel Giovanni Vigna Gianluca Stringhini, Manuel Egele. 2012. *Poultry Markets: On the Underground Economy of Twitter Followers.* https://dl.acm.org/doi/pdf/10.1145/2342549.2342551

[7] Allan Jay. 2023. *Number of Twitter Users 2022/2023: Demographics, Breakdowns Predictions.* https://financesonline.com/number-of-twitter-users/

[8] Pik-Mai Hui Filippo Menczer Kai-Cheng Yang, Onur Varol. 2019. *Scalable and Generalizable Social Bot Detection through Data Selection.* https://ojs.aaai.org/index.php/AAAI/article/view/5460

[9] Summer Lightfoot. 2017. *Political Propaganda Spread Through Social Bots.* https://www.researchgate.net/profile/Summer-Lightfoot-2/publication/324024528_Political_Propaganda_Spread_Through_Social_Bots/links/5ab98f1da6fdcc46d3b9d7c8/Political-Propaganda-Spread-Through-Social-Bots.pdf

[10] Daniel Kats Mahmood Sharif. 2022. *I Have No Idea What a Social Bot Is": On Users' Perceptions of Social Bots and Ability to Detect Them.* https://dl.acm.org/doi/fullHtml/10.1145/3527188.3561928

[11] James H Jones Maryam Heidari. 2020. *Using BERT to Extract Topic-Independent Sentiment Features for Social Media Bot Detection.* https://ieeexplore.ieee.org/document/9298158

[12] Clayton A. Davis Filippo Menczer Alessandro Flammini Onur Varol, Emilio Ferrara. 2017. *Online Human-Bot Interactions:Detection, Estimation, and Characterization.* https://ojs.aaai.org/index.php/ICWSM/article/view/14871/14721

[13] Shannon Thaler. 2023. *Scientists found over 1,000 AI bots on X stealing selfies to create fake accounts.* https://nypost.com/2023/08/24/scientists-found-1140-ai-bots-on-x-creating-fake-profiles/

[14] Security Today. 2023. *Report: 47 Percent of Internet Traffic is From Bots.* https://securitytoday.com/articles/2023/05/17/report-47-percent-of-internet-traffic-is-from-bots.aspx

[15] Marcus Wohlsen. 2011. *Dumb People Sent a Worthless Stock Soaring. Dumb Machines May Do It Next.* https://www.wired.com/2014/07/dumb-people-sent-a-worthless-stock-soaring-dumb-machines-may-do-it-next/

[16] Xiaoyun Han Binyang Li Menglong Lu Dongsheng Li Zhen Huang, Zhilong Lv. 2022. *Social Bot-Aware Graph Neural Network for Early Rumor Detection.* https://aclanthology.org/2022.coling-1.580.pdf

A APPENDICES

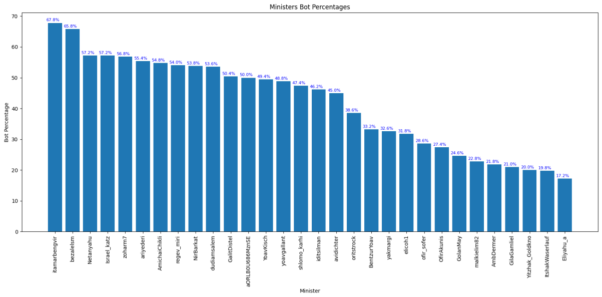


Figure 7: Ministers Bot Percentages

The analysis of Ministers Bot Percentages uncovered intriguing patterns in their online following. Ministers such as 'itamarbengvir' (Itamar Ben-Gvir) and 'bezalelsm' (Bezalel Smotrich) exhibit higher bot percentages, raising concerns about the authenticity of their followers. On the other hand, ministers like 'Eliyahu_a' (Amichai Eliyahu) and 'ItshakWaserlauf' (Yitzhak Wasserlauf) show lower bot percentages, suggesting more genuine online engagement.

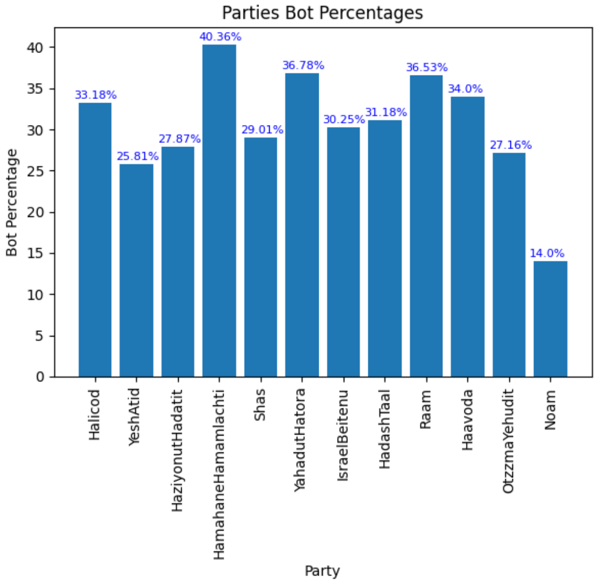


Figure 8: Parties Bot Percentages

As we observe the results in descending order of the number of mandates, it's evident that certain parties have more bot presence among their followers. Parties like 'HamahaneHamamlachti' and 'Raam' exhibit higher bot percentages, possibly impacting the authenticity of their online engagement. Notably, the party 'Noam', represented by only one Knesset member, shows a lower bot percentage. It's worth mentioning that this data might appear unusual due to the party's limited representation.

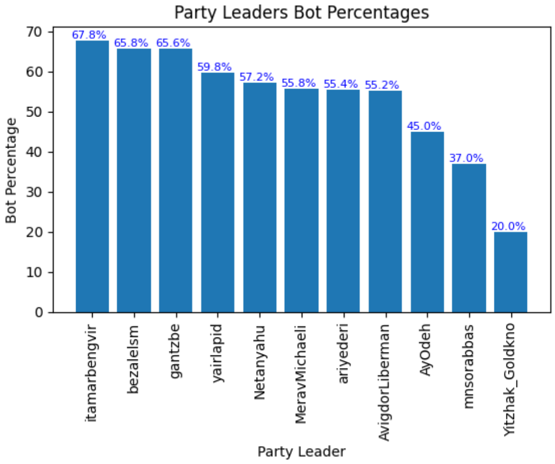
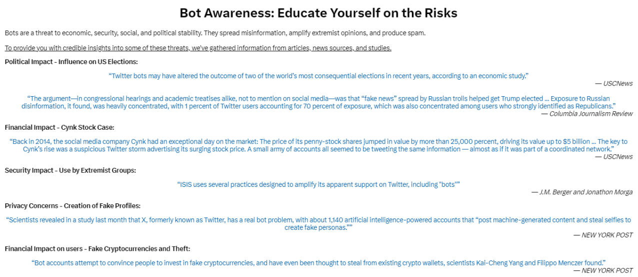


Figure 9: Party Leaders Bot Percentages

The data reveals intriguing insights into the bot percentages among party leaders. Among them, 'itamarbengvir' (Itamar Ben-Gvir) and 'bezalelsm' (Bezalel Smotrich) stand out with the highest bot percentages, while 'gantzbe' (Benny Gantz) and 'yairlapid' (Yair Lapid) also exhibit notable levels. On the other hand, party leaders such as 'AyOdeh' (Ayman Odeh) and 'mnsorabbas' (Mansour Abbas) demonstrate comparatively lower bot percentages. An interesting exception is the remarkably lower bot percentage for 'Yitzhak_Goldkno' (Yitzhak Goldknopf).

Figure 10: ***Bot Detective*** **Logo**

as seen on browser's toolbar



Bot Awareness: Educate Yourself on the Risks

Bots are a threat to economic, security, social, and political stability. They spread misinformation, amplify extremist opinions, and produce spam.

To provide you with credible insights into some of these threats, we've gathered information from articles, news sources, and studies.

Political Impact - Influence on US Elections:

"Twitter bots may have altered the outcome of two of the world's most consequential elections in recent years, according to an economic study."

— USCNews

"The argument—in congressional hearings and academic treatises alike, not to mention on social media—was that "fake news" spread by Russian trolls helped get Trump elected ... Exposure to Russian disinformation, it found, was heavily concentrated, with 1 percent of Twitter users accounting for 70 percent of exposure, which was also concentrated among users who strongly identified as Republicans."

— Columbia Journalism Review

Financial Impact - Cynk Stock Case:

"Back in 2014, the social media company Cynk had an exceptional day on the market: The price of its penny-stock shares jumped in value by more than 25,000 percent, driving its value up to $5 billion ... The key to Cynk's rise was a suspicious Twitter storm advertising its surging stock price. A small army of accounts all seemed to be tweeting the same information — almost as if it was part of a coordinated network."

— USCNews

Security Impact - Use by Extremist Groups:

"ISIS uses several practices designed to amplify its apparent support on Twitter, including "bots""

— J.M. Berger and Jonathon Morga

Privacy Concerns - Creation of Fake Profiles:

"Scientists revealed in a study last month that X, formerly known as Twitter, has a real bot problem, with about 1,140 artificial intelligence-powered accounts that "post machine-generated content and steal selfies to create fake personas.""

— NEW YORK POST

Financial Impact on users - Fake Cryptocurrencies and Theft:

"Bot accounts attempt to convince people to invest in fake cryptocurrencies, and have even been thought to steal from existing crypto wallets, scientists Kai-Cheng Yang and Filippo Menczer found."

— NEW YORK POST

Figure 11: informative pop-up