A Large-Scale Study of Telegram Bots

Taro Tsuchiya[1], Haoxiang Yu[2], Tina Marjanov[3], Alice Hutchings[3], Nicolas Christin[1], Alejandro Cuevas[4]

[1]Carnegie Mellon University, [2]Tsinghua University,
[3]University of Cambridge, [4]Princeton University
ttsuchiy@andrew.cmu.edu, yu-hx21@mails.tsinghua.edu.cn, tm794@cam.ac.uk,
ah793@cam.ac.uk, nicolasc@andrew.cmu.edu, aedcv@princeton.edu

Abstract

Telegram, initially a messaging app, has evolved into a platform where users can interact with various services through programmable applications, *bots*. Bots provide a wide range of uses, from moderating groups, helping with online shopping, to even executing trades in financial markets. However, Telegram has been increasingly associated with various illicit activities—financial scams, stolen data, non-consensual image sharing, among others, raising concerns bots may be facilitating these operations. This paper is the first to characterize Telegram bots at scale, through the following contributions. First, we offer the largest *general-purpose* message dataset and the first bot dataset. Through snowball sampling from two published datasets, we uncover over 67,000 additional channels, 492 million messages, and 32,000 bots. Second, we develop a system to automatically interact with bots in order to extract their functionality. Third, based on their description, chat responses, and the associated channels, we classify bots into several domains. Fourth, we investigate the communities each bot serves, by analyzing supported languages, usage patterns (e.g., duration, reuse), and network topology. While our analysis discovers useful applications such as crowdsourcing, we also identify malicious bots (e.g., used for financial scams, illicit underground services) serving as payment gateways, referral systems, and malicious AI endpoints. By exhorting the research community to look at bots as software infrastructure, this work hopes to foster further research useful to content moderators, and to help interventions against illicit activities.

1 Introduction

Telegram is one of the largest social apps in the world (1 billion users (Telegram 2025c)) and an application through which people interact with the web (e.g., get news, buy goods online). It enables individual or group chats, as well as broadcasting within channels. In recent years, Telegram has evolved beyond a chat platform, and now offers comprehensive software infrastructure: user authentication, static domains, web browsing, applications (*mini apps*), and advertisements (Telegram 2025b,d). Most importantly, and key to this paper, Telegram allows developers to offer various services through *bots*, programmable applications running within Telegram that facilitate interaction with users through commands, messages, and inline queries. Telegram users can directly message bots or ask them to perform tasks (e.g., moderating messages) in a channel or group. Bots can also collect payments from users for goods and services, and through traditional payment methods (e.g., Google or Apple Store), cryptocurrencies, Telegram's own cryptocurrency (TON) (Telegram 2025g), and Telegram's in-app currency (Telegram Stars) (Telegram 2025f).

Telegram has attracted a diverse user base, ranging from TV show fans, students preparing for exams, to cryptocurrency traders (Mozur et al. 2024). Recent years have seen growing concerns about Telegram's association with a variety of illicit activities, leading to increased scrutiny of Telegram's lenient content moderation policy. Recent evidence suggests that Telegram has become a new hub for data leaks and pirated software (Roy et al. 2025; Marjanov and Hutchings 2025; Marjanov et al. 2026; Lieber 2023), fraud (Gao et al. 2020; Cernera et al. 2023; Bijmans et al. 2021), non-consensual image abuse distribution (Burgess 2020; Semenzin and Bainotti 2020), money laundering (Gebrekidan and Dong 2025), propaganda (Kireev et al. 2025a; Hanley and Durumeric 2024), and extremist groups (Mozur et al. 2024). Anecdotal evidence suggests that bots are crucial in scaling these illicit operations because they allow operators to automate interactions with users and content.

However, little is known about the services bots provide since past work has largely focused on messages, channels, and groups (Baumgartner et al. 2020; La Morgia, Mei, and Mongardini 2025; Guo et al. 2024; Blas, Luceri, and Ferrara 2025; Kireev et al. 2025b; Gangopadhyay et al. 2025). In addition, Telegram does not have a centralized bot repository, making it difficult to enumerate these bots. To fill this gap, we offer the following contributions.

1. We provide the largest *general purpose* message dataset and the first bot dataset on Telegram. We update existing public datasets, extract additional channels and messages through snowball sampling, and extract the list of bots. Our dataset contains approximately 106,000 channels and 809 million messages, including 67,000 channels and 492 million messages that had not been reported before, 32,000 bots, and 9 million links between bots/channels/users.

2. We develop a novel system to autonomously send commands to each bot and record its responses. This is necessary to understand bot functionality, since the bot description is

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

often insufficient.

3. To understand what services bots offer, we classify the bots into several domains and functionalities based on our largest dataset above. We label all bots using a combination of manual, keyword, and Large Language Models (LLMs) analyses. While most bots are benign, we find an alarming number of bots with illicit operations such as fraud (4%) and underground (5%) (e.g., nudification apps, unauthorized access to paid content, and stolen data). Those bots often process payment, manage referrals, give access to ill-gotten digital goods, or host malicious AI endpoints.

4. We analyze the communities around bots to understand the bot usage. Despite descriptions being in English, many bots are used in non-English communities. While Russian is the most common language for most categories, English is predominant among finance bots. Finance bots are the most prevalent, but are often short-lived, reused, and frequently flagged by Telegram as scam. Ideology bots are often designed for a specific, close-knit community, whereas utility bots appeal to broader sparse communities.

Based on our analysis, we argue that bots have become a core infrastructure for facilitating illicit activities on Telegram, for instance, serving as payment processors, referral systems, or malicious AI endpoints. We suggest that Telegram and law enforcement focus on bot moderation as a potential point of disruption. We offer four practical recommendations: 1) looking at the bot command list and messages, 2) focusing on specific domains based on language, 3) grouping similar bots, and 4) identifying related channels through links. Our findings highlight the need to study Telegram ecosystem beyond messages and groups. Our dataset can serve as a valuable resource enabling further research. Following the ethical statement in §6.3, the dataset and the code are available at https://zenodo.org/records/17281308 and https://github.com/taro-tsuchiya/TelegramBots, respectively.

2 Background

We start with an overview of Telegram, discuss related work, and compare our data with previously released datasets.

2.1 Telegram Overview

Since its 2013 launch, the Telegram messaging app has positioned itself as a secure and privacy-focused alternative to its competitors. It has grown to over 1 billion active users (Telegram 2025c), and has become a key messaging channel for various communities, businesses, news organizations, and even heads of state. However, Telegram has also been criticized for its lax content moderation policies, which have allowed it to become a hub for various illicit activities (Mozur et al. 2024).

Today, Telegram is a platform which offers various features for users and developers, ranging from group messaging to advanced web services. Telegram supports two modes of group communication: *Channels* (one to many, unlimited number of members) and *groups* (many to many, up to 20,000 members). Furthermore, websites can authenticate users with the Telegram widget (Telegram 2025d), and even verify ID with the Telegram Passport program (Telegram 2025e). Telegram also supports web-based applications ("mini-apps," e.g., games, e-commerce) that can be launched within Telegram or a fully-fledged in-app browser (Telegram 2025b). Users do not need to leave Telegram to access mini apps or websites. Telegram further offers a variety of payment integration options: traditional credit cards, cryptocurrencies, the newly launched "Telegram Stars" (Telegram 2025f), an in-app currency to facilitate transactions between Telegram users, and even its own blockchain—The Open Network (TON) (Telegram 2025g)—originally developed for cryptocurrency payments into the Telegram ecosystem. Users can use TON and the associated Toncoin cryptocurrency to purchase goods, services, and even advertisement space within Telegram.

Telegram *bots* are the primary way to offer goods and services programmatically within Telegram. Bots provide a text-based interface to users, allowing them to interact with them through commands and other inputs. For example, a user can provide an URL to initiate a download or provide an image for processing. Bots can also request payments and manage memberships, all without leaving Telegram. To create a bot, developers first register a new bot through Telegram's "BotFather," a special bot that helps users manage their bots. They *must* choose a username that ends with "`bot`," (case insensitive), and obtain an API token. Developers then write a script in any programming language (e.g., JavaScript, Python), run it on their own device/server or cloud, and connect it to Telegram through the Telegram Bot API and their API token. Telegram requires developers to support two global commands to ensure uniform user experience: `/start` (to launch the description and welcome message) and `/help` (to list all functions) (Telegram 2025a).

2.2 Related Work

Because of Telegram's lenient content moderation policy, Telegram has become a hub for problematic content such as propaganda (Kireev et al. 2025a; Hanley and Durumeric 2024), disinformation (Ng et al. 2024), conspiracies (Steffen 2025; Imperati et al. 2025), cybercrime (Guo et al. 2024; Roy et al. 2025; Marjanov et al. 2026), and phishing (Gao et al. 2020; Cernera et al. 2023; Bijmans et al. 2021). Cryptocurrency and blockchain communities are also active on Telegram (Baumgartner et al. 2020), and several studies document cryptocurrency price manipulation coordination attempts (Nizzoli et al. 2020; Xu and Livshits 2019; Mirtaheri et al. 2021).

Despite the critical role that bots play in the Telegram ecosystem, only a handful of studies investigate or even mention them. Roy et al. (2025) characterize the use of bots for cybercrime in their Appendix, discussing payment processing, content distribution, and channel expansion. Ricaldi et al. (2025) show that Telegram bots can be used for gaining customers' trust in underground markets (e.g., to automate the order process). Alrhmoun, Winter, and Kertész (2023) study Telegram bots in channels supporting the Islamic State, identifying two roles: content distribution and group management. Perlo et al. (2025) show the prevalent use of Telegram bots; nearly 90% of their groups in-

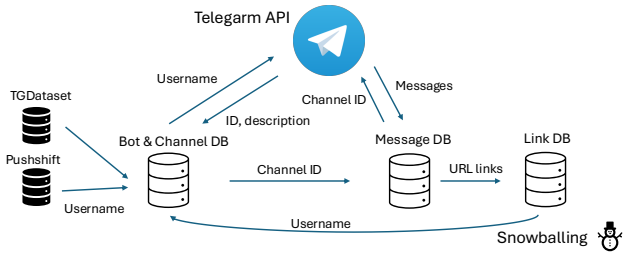


Figure 1: Data collection pipeline

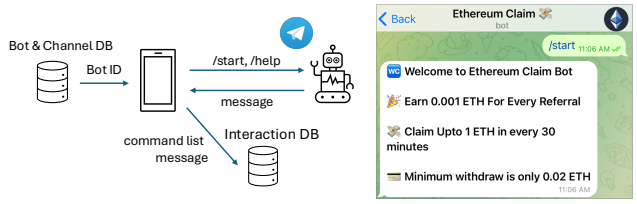


Figure 2: Bot interaction pipeline

clude bots. Other papers mention the use of Telegram bots for sharing non-consensual images (Semenzin and Bainotti 2020; Franco, Gaggi, and Palazzi 2024), controlling IoT devices (De Oliveira, Santos, and Neto 2016), or managing groups (Nikkhah, Miller, and Young 2018). Recent web articles describe the use of bots to help cybercriminals communicate with malware-infected devices (Nigam and Wilhoit 2018), support phishing operations (Büyükkaya 2024), non-consensual image abuse (Burgess 2020), or recruit members for extremist groups (Mozur et al. 2024). However, our paper is the first to characterize bot roles and functionality at scale, for both legitimate and malicious uses.

2.3 Comparison with Previous Datasets

We provide 1) the largest general-purpose Telegram message dataset, and 2) the first bot dataset, including bot/channel descriptions and linkage, and our interactions with bots. We review past Telegram datasets and compare them to our work. We limit our comparison to those 1) are publicly hosted and accessible, 2) documented, and 3) contain at least 1 million messages. Table 1 summarizes the seven existing such datasets and ours. Some datasets cater to specific topics such as politics (Baumgartner et al. 2020; Blas, Luceri, and Ferrara 2025), propaganda (Kireev et al. 2025b), war (Bawa et al. 2025), and underground markets (Guo et al. 2024; Marjanov et al. 2026), while others (including ours) are general purpose (La Morgia, Mei, and Mongardini 2025; Gangopadhyay et al. 2025). Most works start from seed channels (through a website, e.g., TGStat or keyword search) and use snowball sampling to discover more channels (as we do in §3.1). As Table 1 shows, our data include the largest general-purpose 492 million message dataset, and a unique bot dataset.

3 Data Collection

We explain how we collect Telegram channels[1], messages, and bots.

3.1 Collecting Channels and Messages

We show our data collection pipeline in Figure 1. We start our snowball sampling from the channels from the Pushshift (Baumgartner et al. 2020) and TGDataset (La Morgia, Mei, and Mongardini 2025) datasets. As Table 1 illustrates, Pushshift (Baumgartner et al. 2020) features over 30,000 channels and primarily focuses on right-wing or cryptocurrency-related channels, whereas TGDataset (La Morgia, Mei, and Mongardini 2025) includes nearly 120,000 channels and covers a broader variety of topics (e.g., US news, entertainment). Combining both datasets provides a general overview of the Telegram ecosystem. As both datasets are old (last collected in Oct. 2019 and Jul. 2022, respectively), we begin by updating messages from those channels.

Telegram does not have algorithms to recommend bot/channels, so their information is often passed through URL links in messages. We extract channel names from URLs using regular expressions that match patterns such as `telegram`, `tg`, `t.me/`, and others. After verifying the existence of those channels through the API, we conduct snowball sampling—iteratively collecting new channels through URL links from new messages. For each channel, we also collect its description. Besides discovering new channels, we also continuously collect new messages from the channels already collected. Notably, Telegram lets us get public channel metadata and messages passively, i.e., without joining the channel. We do not join any private channels or collect any media files (e.g., images, videos, documents).

Our collection ran continuously from May 19th until Aug. 18th, 2024, resulting in a dataset of 105,970 channels and 809,481,087 messages. We discovered 67,868 new channels (not in Pushshift and TGDataset) and 492,256,372 messages (not in Pushshift).[2] We extract 9,041,103 URL links to other channels/bots/users, far more than previous efforts (2.7 million links by Gangopadhyay et al. (2025)).

3.2 Bot Discovery

Bots Collected from Messages. Bot information is typically passed through user links (e.g., `t.me/USERNAME`) (Telegram 2025a). For bots, `USERNAME` *must* end with (case-insensitive) "`bot`" (see §2.1). We look at channel messages[3] and extract all user links and exclude channels/groups based on API response. For the remaining user links, we extract the usernames ending with "`bot`." Two of the authors checked

[1]For our analysis, we do not distinguish channels and groups and refer both as channels. 82.2% are channels and 18.8% are groups.

[2]TGDataset stores messages differently from Pushshift or us, so we could not directly compare against it.

[3]We might miss bots that are not mentioned in messages but only in channel descriptions.

Author	Year / Venue	Nr of channels	Nr of messages	Collection date	Topic	Snowball
Baumgartner et al. (2020)	2020, ICWSM	30K	317M	∼Oct 2019	Politics + Cryptocurrency	Yes
La Morgia, Mei, and Mongardini (2025)	2025, KDD	120K	400M	∼Jul 2022	General	Yes
Guo et al. (2024)	2025, SIGMETRICS	71K	200M	∼Feb 2024	Underground	Yes
Blas, Luceri, and Ferrara (2025)	2025, WWW (companion)	43K	1B	∼Feb 2025	Politics	Yes
Kireev et al. (2025b)	2025, ICWSM	13	13M	∼Oct 2023	Propaganda	No
Bawa et al. (2025)	2025, ICWSM	518	5M	∼Mar 2023	War	No
Gangopadhyay et al. (2025)	2025, ICWSM	71K	120M	∼Oct 2024	General	Yes
Marjanov et al. (2026)	2026, USENIX Security	1,521	14M	∼Aug 2025	Underground	Yes
This paper	*2026, ICWSM*	*106K (68K new)*	*809M (492M new)*	*∼Aug 2024*	*General*	*Yes*

Table 1: Comparison of publicly available Telegram datasets.

100 random samples and found no obvious false positives—i.e., regular user handles serendipitously ending with "`bot`."

Bots Collected from Third-Party Directories. Because Telegram does not have a centralized bot directory, users also rely on third-party websites that list bots and channels. To increase our coverage, we also extract the bots from telegramic.org/bots, tgbots.io/, telegrambotlist.com, tgdr.io, and telega.io.

Combining both sources yields a total of 32,071 bots (23,886 (74.5%) through 1,147,674 URL links across 17,455 channels; and 8,185 (25.5%) through third-party directories). We verify that each bot is active and not deleted, and collect its description through the Telegram APIs: `userFull` and `botInfo`. The description includes the "`about`" field in the bot profile, the description of the bot (at start time), and the list of URLs displayed in the bot profile. 20,416 (63.7%) bots have non-empty descriptions.

3.3 Extracting Bot Functionality

While some bots may use their description to detail their functions in the description, many bots instead rely on 1) external documentation or 2) interactions with users. The former is more common for bots that carry out routine tasks (e.g., moderation) within groups—we can extract this information from messages. To cover the latter case, we interact with bots. Figure 2 illustrates the interaction pipeline, along with one example of a bot interaction. We launch the Telegram client (using the official API) and issue two required commands (`/start`, `/help` (see §2.1)) to all bots in our dataset, and record responses within a $5s$ timeout. Bots whose script is not running return no response. In addition, not all bots adhere to this standard. In total, 7,375 bots responded to either `/start` or `/help` commands. (4,959 bots responded to `/start` and 5,502 bots responded to `/help`.) When the bot is flagged by Telegram as a scam or fake account, it will respond with a warning message;[4] we found 1,331 such bots. This number is a lower bound, as scam bots could have been deleted prior to our data collection. Last, 9,197 bots returned a command list upon request. 67% support `start`; 28% , `help`, 7%, `balance`; 7%, `settings`; and 6% feature `menu` commands.

[4]"Warning: Many users reported this account as a scam or a fake account. Please be careful, especially if it asks you for money."

4 Characterizing Bots

To understand the bots' services, we classify all 32,071 bots into the domains they operate in and the functionalities they provide.

4.1 Usage Domains

We first illustrate how we categorize the domains, and then show the classification results.

Methodology One author, who is an active Telegram bot user, initially came up with 21 domains based on the Telegram channel topics (TGDataset) and the websites with the list of bots (§3). As a test run, two authors independently annotated 100 test samples, discussed disagreements, and merged less common categories.

We ended up with nine categories:

1. **Admin Tools** (AT): Bots that manage groups on behalf of owners, such as membership management, question answering, content moderation, and group statistics.
2. **Content & Media** (CM): Bots that help distribute or collect educational and training materials, streaming (music, movies, TV series), and news media.
3. **Ideology** (ID): Bots used for political campaigns, social movements, or religion-related purposes.
4. **Finance** (FN): Bots that provide access to financial services, such as online wallets, trading, airdrops, mining, or providing financial information.
5. **Shopping** (SP): Bots that facilitate online shopping, including selling and buying products, collecting reviews, providing customer service, and product search.
6. **Social & Gaming** (SG): Bots that facilitate online interactions (e.g., chatting for fun, dating, games, and gambling).
7. **Underground** (UG): Bots that support underground operations, such as cybercrime (e.g., hacking, stolen data, phishing) and adult content.
8. **Utility** (UT): Bots that provide tools or functions to individual users (not groups), such as using LLM endpoints, developer tools, web search, photo & video management, health & fitness management, and QR code generation.
9. **Other**: Bots that do not belong to the categories above or are unknown given the input.

Using this categorization, two authors independently annotated another random sample of 100 bots based on their

usernames, descriptions, interactions, and the messages containing links to the bots. We treat each source of information equally, but look at the messages that may contain other contexts with caution. We choose only one category for each bot. If the bot operates in multiple domains, we follow the instructions specified in the prompt (see Appendix A.1). For instance, we classify "selling Netflix accounts" as Underground rather than Shopping or Content & Media. If not specified, the coders chose the most prominent category.

We only include messages that mention one bot, to 1) exclude messages enumerating many bots for advertisement, and 2) avoid referring to other bots. We only use the first 50 messages or a maximum of 2,000 characters to reduce the amount of text (for both LLMs and human annotators). We manually find that those thresholds are sufficient to capture the context. We provide both the original text and the English translation (using Google Translate) to help with human annotation. Coders are allowed to search keywords online, but not allowed to 1) search or interact with the bot directly or 2) ask LLMs for help.

The interrater agreement, Cohen's κ (Cohen 1960), for the two coders is 0.67, which is considered substantial agreement. Most of the discrepancies arise from 1) speculation based on the keywords in the username (e.g., is the keyword "ETH" sufficient to select Finance?), 2) ambiguity regarding the bot's target audience (e.g., is a Q&A bot intended for use by administrators or individual users?), 3) lack of domain knowledge (e.g., a clothing brand, an anime character), or 4) less commonly, ambiguity in the prompt. We discuss the disagreements one by one, reach a consensus, and slightly modify our codebook to clarify ambiguity.

We next use OpenAI LLMs to scale up the annotation to *all* bots. To verify the accuracy of LLMs, we run gpt-4o (2024-08-06) and gpt-4o-mini (2024-07-18) on the same 100 samples first and compare the results with human annotation (consensus). We do not translate any text and directly feed the original text to LLMs. For model parameters, we set *temperature* to 0 and *top_p* to 1 to reduce randomness and improve the reproducibility (default values for other parameters). The agreement between human consensus and LLMs is 0.73 for gpt-4o and 0.70 for gpt-4o mini. After a manual inspection on all samples, most of the disagreements come from reasons similar to those observed with human annotators—bot description ambiguity, e.g., "coin" can be both in Finance or Social & Gaming depending on the context. We rarely find cases where LLMs make mistakes due to misunderstanding the context, but rather sometimes correct humans' mistakes. Between two LLMs, the gpt-4o model appears to be slightly more conservative than the gpt-4o-mini model (e.g., uses Other more often, rarely makes judgment just based on username), while the differences appear to be minimal. We primarily use gpt-4o classifications for further analysis. Appendix A.2 provides the annotation results (e.g., domain distribution) for two human annotators and two LLMs.

Results We apply the same prompt for the entire sample and run two LLMs. We also ask them to produce a one-sentence summary so that we can review the categorization. Figure 3 shows the number of bots per category. We have 34% (n=11,206) of bots in Finance, 11% (n=3,517) in Admin Tools, 10% (n=3,222) in Content & Media, 9% (n=2,920) in Utility, 9% (n=2,818) in Social & Gaming, 6% (n=2,004) in Ideology , 5% (n=1,539) in Underground, 5% (n=1,451) in Shopping, and 11% (n=3,394) in Other. We also aggregate the number of unique channels that mention those bots for each category. As Figure 3 shows, Finance has a disproportionately high number of bots compared to its number of channels, while Social & Gaming shows the opposite. This indicates that Finance bots are distributed by a smaller set of channels. Indeed, when we look at the top ten channels that mention the most bots, nine of them are advertisement channels that are dedicated to the promotion of Finance bots (i.e., 70-100% are Finance bots). Those top ten channels discovered 4,674 Finance bots (41.7% of all Finance bots).

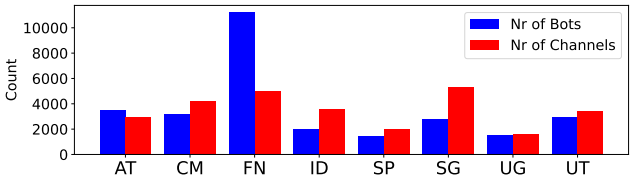


Figure 3: Number of bots and channels per category.

4.2 Analyzing Functionality

Through the manual bot annotation process in §4.1, three common sets of capabilities emerged across bots: payment processing, referral management, and input collection. Additionally, we observed a substantial number of bots using artificial intelligence to provide services. We refer to these functionalities as Payment, Referral, Crowdsourcing, and AI.

Payment We first examine whether bots process payments. To do this, we use a keyword matching approach, whereby we classify the payment functionality based on the command list that the bot provides. Initially, we attempted to use LLMs to infer functionalities, but they often failed to distinguish between the bot's functionalities (i.e., what it actually does) and the context in which the bot is used (e.g., the airdrop bot does not always process payment, but just helps expand channels). The keyword matching approach reduces false positives at the expense of being more conservative, thus providing a lower bound. We select the list of keywords based on 1) expert knowledge and 2) manual inspection of the frequent bot commands. We translate all commands and their descriptions to English using the Google Translate API. We follow the same process for Referral and Crowdsourcing functionalities.

We select the following keywords: "pay," "payment," "purchase," "buy," "sell," "deposit," "withdraw," and "set-wallet." To verify the accuracy of keyword matching, we manually checked 100 random command name and descriptions and found no false positives. In total, 9.4% (864 out of 9,197 bots with a command list) have payment-

related commands. We manually group the command variants (e.g., `withdrawal`, `quickwithdraw`) to the base command (e.g., `withdraw`). The most common commands are to `withdraw` (n=474), `wallet` (n=369), `deposit` (n=206), `pay` (n=202), and `buy` (n=114). Users are more likely to withdraw than to deposit because they can cash out without initial deposit (e.g., airdrops, referrals). Bots also set or connect wallets, buy service credits, create payment links, invest, and upgrade to VIP memberships. We could not identify the type of payment method (e.g., crypto, credit card), since figuring this out typically involves additional interaction.

Referral We examine whether the bot produces referral links or asks users to invite friends. We follow the same strategy as Payment and select the following keywords: "refer," "referral," and "invite" for matching. We explicitly remove "references" because it is often used in different contexts (e.g., to reference a book). We only focus on the referral aspects, so we do not count the "gateway" bot that simply asks users to join the corresponding channel (e.g., `join` command). From randomly chosen 100 commands, we find no false positives (by only looking at the command descriptions). In total, 10.4% (n=954) of bots have referral-related commands. Those bots typically issue customized referral links to users and ask them to invite their friends: `referral` (n=638), `invite` (n=296). Based on the command description, they also give rewards ("earn credit"). The form of rewards is either free cryptocurrencies, tokens (e.g., TON), or VIP subscriptions.

Crowdsourcing We investigate how the bot collects textual information or files from a group of users. Likewise, we choose the following keywords: "upload," "submit," "report" to match the command list. We exclude *reports* (plural) command because it is often used to produce statistics. We manually find 11 false positives out of a random 100 commands, mostly `report` command, used for statistics. We ultimately decided against excluding "report" as it is generally used for accepting inputs from users. In total, we have 1.2% (n=113) bots with crowdsourcing-related commands: `report` (n=64), `upload` (n=17), and `submit` (n=12).

AI We finally investigate whether bots provides a service based on artificial intelligence (AI). Bots do not necessarily have a specific set of commands for AI-powered services, and the definition of "AI-powered" service is vague, so we perform the keyword matching on the bot's (translated) description. We generate 45 AI-related keywords, ranging from basic machine learning, such as "ml," "ocr," "translation," to the recent generative models such as "chatgpt." We do not allow partial matches because the description often contains a large set of words (e.g., "airdrop" matches "ai"). We manually check 100 random bots to verify the accuracy of keyword matching, and find that the keyword "transformer" is sometimes used as a robot but not the AI model, so we exclude it. 607 bots have AI keywords in their description. The most common matched keywords are general descriptions of AI: "ai" (n=363), "artificial-intelligence" (n=96), "translation" (n=42). We also observe

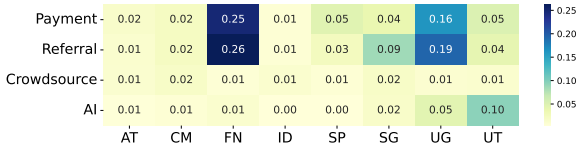


Figure 4: Distribution of bot domains and functionalities, e.g., 25% of Finance bots (with a command list) have a payment functionality.

many bots using the generative model: "chatgpt" (n=69), "gpt" (n=58), or image-generation: "midjourney" (n=20), and "dall-e" (n=15).

4.3 Domains and Functionality

We quantitatively show the relationship between the domains and functionalities. Figure 4 shows the distribution of bot domains (x-axis) and functionalities (y-axis). We divide by the total number of bots in each domain by the number of bots *with* command list (all bots for AI). Each bot can have multiple functionalities. Finance (FN) and Underground (UG) are significantly more likely to use bots as a payment processor or a referral system compared to other domains. While Shopping (SP) bots have a relatively low payment functionality given their nature, those bots still support online shopping through product search (e.g., catalog, price) or customer service (Q&A, reviews). Content & Media (CM) and Social & Gaming (SG), the two categories that often connects users, have relatively higher rates for crowdsourcing. In terms of AI usage, we find that Underground (UG) and Utility (UT) are the most prominent. The next section explains why certain domains predominantly use specific functionalities, and introduces some case studies for both benign and malicious use cases.

4.4 Benign Uses

We identify a variety of benign uses. For example, crowdsourcing bots mostly support legitimate use cases. First, there are bots that accept user input from various geographical locations. For example, we find a bot in Tbilisi that collects information about a polluted area in the city, or a bot in Greek refugee camps that aggregates the volunteer information or field status. Second, bots can assemble knowledge from the crowd, such as creating a public library—collecting and distributing books. Third, content moderation bots accept user reports on problematic users or messages.

There are several benign apps that use AI. Bots often perform basic tasks such as translation, OCR (e.g., extracting texts from photos), photo editing, or internet search, supporting the high use of AI in Utility bots. In addition, some bots utilize generative AI models to converse with users as a friend (e.g., for fun or language learning) or a supporter (e.g., to provide emotional or fitness support). Interestingly, there is a bot that allows users to use AI anonymously (and use cryptocurrency as a payment), which could be used for circumventing censorship. In Finance, we frequently observe some bots that use AI for trading (analyzing social media or assets).

4.5 Malicious, Illicit, and Exploitative Uses

We next identify various cases in which bots help facilitate malicious activities (e.g., scams and fraud), illicit (e.g., selling cybercriminal goods, piracy), and exploitative (e.g., undressing apps). The following analyses are based on sampling of 100 bots with Finance & scam labels and 100 bots categorized with the Underground domain.

Scams and Fraud. Bots used for scams were the most prominent. Telegram currently has a label for bots that are suspected of being scams, which is visible when users try to start a conversation with the bot (§3). Around 10% of the Finance bots in our dataset had been assigned a scam warning, significantly higher than the 1% average in non-Finance domains. We also find that Finance *scam* bots have exceedingly higher payment and referral functionality (71% and 65% respectively), suggesting that they often ask or offer payment to users and employ tactics to attract more users. To understand the nature of those bots, we investigate their types and tactics.

A substantial number of bots offered airdrops or giveaways. Those bots typically use "free" as a keyword, and solicit users to 1) join a channel, 2) invite friends, or 3) watch advertisements. Some bots use the term "mining," which is not necessarily related to mining blockchain with computational resources, but as a way to earn tokens in online games. The second prominent case was investment bots that promise a high return by depositing or adding liquidity to the pool. For instance, one bot offers users a chance to earn tokens without investment, but can earn faster by investing more. Most bots are related to cryptocurrencies such as BTC, BSC, and TRX and sometimes ask users to provide their blockchain addresses. The results verify the high payment and referral functionalities of Finance. Lastly, we also identify non-financial scams such as referral schemes or offering free items (in Underground).

Illicit Goods and Services. Many bots enable underground markets offering illegal goods or services, such as stolen data (e.g., "combo" lists (URL, login, password), credit cards, breached databases), drugs, or doxing (e.g., "bombing" texts for annoyance). For example, users can use bots to access inventory of leaked databases containing personal information through lookup commands such as `name` and `phone`. Another common category of goods and services were those associated with artificial engagement (e.g., views, likes, members). We also identified bots allow access to unauthorized content, likely violating the terms of service of the original providers. For instance, there are bots that generate phone numbers, giving access to streaming services (e.g., Netflix) for free or referral points, premium accounts of various apps (e.g., VPN services, online shopping, social media), or reselling accounts. These bots typically monetized their services through direct in-app payment or membership plans, and employed a variety of referral tactics to attract users. These results show an increasing trend of illicit activities which were commonly offered through Tor hidden services, but are now facilitated through Telegram in a more user-friendly and mobile-first manner.

Exploitative Content. Many bots facilitate the search, distribution, or generation of adult content. More alarmingly, the majority of those bots offer services to create deepfake images, undressing pictures, or swapping faces, which aligns with the surge of nudification websites (Han et al. 2025; Gibson et al. 2025) and non-consensual image sharing channels on Telegram (Semenzin and Bainotti 2020). While these bots could be used with the consent of the image subject, some of them are advertised for potentially non-consensual purposes. Some bots provide a disclaimer that users must be over 18 or adults to use the application, although they may not have mechanism to verify the age.

5 Bot Usage

This section analyzes channels surrounding each bot to understand the bot usage: supported languages, duration, reuse, and topology.

5.1 Languages

Language is an important aspect of bot usage as it can reflect the target geographic audience. We run the language detection (using langdetect library (Danilk 2025)) on both 1) the bot description and 2) the messages that mention the bot. Our data contains short text, abbreviations, and slang, which make language detection challenging. To address this, we remove URLs and emojis before detection and perform several rounds of adjustments where we manually inspect 200 classifications and add/subtract a bias in detection probability to the languages that are often under-/overclassified until no significant improvements can be made. For the bot description, 42% of bots are in English, 26% in Russian, 11% in Farsi, and 6% in Arabic. For the messages that mention bots, we run the detection for all deduplicated messages that are associated with each bot, and take the most frequent language. 30% are in Russian, 30% in English, 13% in Farsi, and 8% in Arabic. This suggests that bots that have descriptions in English are often used in non-English speaking communities. We also confirm that bots whose description is in Russian, Farsi, and Arabic are mostly used by the same language communities. This result highlights the importance of looking at the contexts (i.e., collecting messages) to understand the bot usage.

We next compare the language (in messages) against the domains we discover in §4.1. Figure 5 shows the distribution of bot domains (x-axis) and the top five languages (y-axis). We find that the primary language significantly differs by domain. English is only dominant in Finance (FN), 42%, compared to 2% in Ideology (ID) while Russian is widely used in many domains. We also observe a relatively high concentration of Arabic in Ideology and Farsi in Shopping (SP) and Social & Gaming (SG). Underground (UG) has a diverse set of languages.

5.2 Duration and Reuse

We next analyze how long the communities use bots and whether they reuse them. We also look at the aggregated trend of bot usage over time. We define bot duration as the time difference between the first and last time the bot is mentioned in messages, which reflects the time period during which the bot is actively used. We exclude bots that have

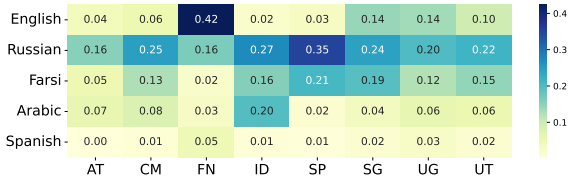


Figure 5: Distribution of bot domains and languages. Showing only the top five languages, so the columns do not add up to 1.

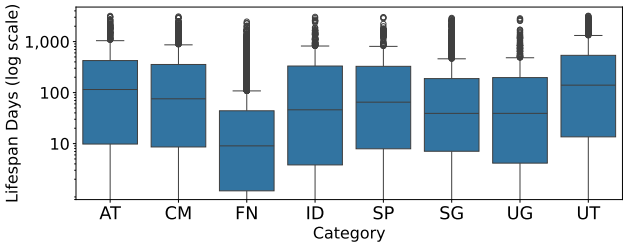


Figure 6: Distribution of bot duration per category.

been mentioned only once. The average (median) duration is 178 days (21 days). Figure 6 shows the distribution of bot duration per category. Expectedly, Utility (UT) and Admin Tools (AT) have long durations (median 140, 115 days, respectively) as those bots provide relatively static services. On the other hand, Finance has the shortest duration (median 9 days), likely due to 1) the short life cycle of investment opportunities (e.g., airdrop), 2) the malicious use, as evidenced by a high number of scam warnings in §4.5, and 3) the reuse of the same bot with different usernames.

We indeed observe that some bot developers appear to recreate the bot with the different username (i.e., reuse). About 3.4% (1,074 bots, including the original bot) have the exact same descriptions (with the text length of more than 10 characters). Of those, 43% belong to the Finance category. At maximum, one bot has been reused 30 times. We also look at the similarity of usernames that have the same description. Bot developers either 1) use the same username with a slight modification with numbers and characters (e.g., *aidropbot*, *aidrop2bot*, ...), 2) iterate different keywords (e.g., swapping financial asset names), or 3) use completely different ones. To quantify the similarity of usernames, we calculate the Levenshtein distance[5] for every pair in the group that shares the same description. 385 (35.9%) of bots have a username that is similar to at least one username in the group (with a distance of 1 or 2), which is likely to fall into the first category.

We further look at the overall trend in the number of active bots. Figure 7 shows the number of monthly active bots over time, as well as the ratio (%) of AI-related bots (as defined in §4.2). We define a bot as active if the bot has been mentioned at least once in that month. Generally, the

[5]Number of operations needed to replace one string with another, including swapping.

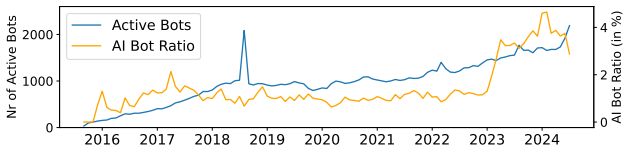


Figure 7: Number of monthly active bots over time with the ratio of AI-related bots.

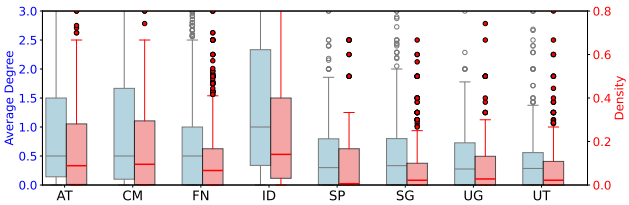


Figure 8: Distribution of channel network metrics per bot domain (blue: average degree, red: density).

number of active bots has been increasing over time in our dataset. The significant spike in Aug. 2018 is likely caused by one channel that advertised 1,016 bots in three days. Bots that provide AI-related services have significantly increased since early 2023, which coincides with the release of GPT-4 in March 2023. Telegram's user friendly interface may have contributed to the adoption of AI for mobile-users, which we discus in §6.

5.3 Channel Topology

We next look at the network topology of the community surrounding each bot. We prepare a set of *channels* that mention the same bot and investigate if there is any interaction within those channels (i.e., a channel mentioning another channel through URLs). We construct an undirected graph for the set of channels that mentions the same bot and draw an edge if one channel mentions another channel (i.e., no bot in the graph). To represent the connectivity of the graph, we use two metrics: (1) *density*, which is the ratio of the number of edges to the number of possible edges, and (2) *average degree*, which measures the average number of connections per node in the graph. For both metrics, higher values indicate a more connected network (i.e., a dense community).

We only consider bots that are mentioned by at least 4 channels to exclude isolates, dyads, and triads. Figure 8 shows the distribution of density and average degree per domain. We find that Ideology (ID) bots have the highest density and average degree, whereas Utility (UT) bots have lower scores in those metrics. This implies that communities around Ideology bots are more interconnected. Due to their coordinated activities related to religion, politics, war, and social movements, their bots are likely designed for a specific community. Utility bots, on the other hand, are likely mentioned by a diverse set of channels with less interaction among them, and are mostly designed for individual use.

Figure 9 shows the three example channel networks; the left and the middle figures are in Ideology and the right is in



Figure 9: Channel networks for three example bots.

Utility. We keep the isolates in those figures. The left Ideology bot appears to collect information on Russia's military operations to Ukraine, and the middle Ideology bot seems to distribute an Iranian leader's publications. All the well-connected channels appear to be related to those specific topics. The right Utility bot helps users discover other bots. While some Brazilian communities happen to use the same bot, which creates a small cluster, the rest is highly disconnected. Bots in different domains exhibit different purposes and usage patterns.

6 Discussion

This section highlights how our findings can inform future research, platform policy, and regulation. We also discuss the limitations of our study and ethical considerations.

6.1 Implications

We provide the first characterization of Telegram bots use at large scale. While the majority of bots are benign, we uncovered bots which are used for illegal or concerning purposes, such as carrying out fraud (n=1,331, 4%), commercializing illicit goods and services, or providing access to questionable services, such as AI non-consensual deepfakes (n=1,539, 5%). Unlike traditional darknet marketplaces or forums on the Tor network that require desktop environments, Telegram offers a mobile-first, easier-to-use (for both developers and end users) platform that lowers the barrier to provide questionable or outright illicit services. We theorize that Telegram fills a gap between clearnet websites and hidden services. The growth of "gray" AI services and the abundance of cryptocurrency/blockchain-related bots support this hypothesis.

Our findings also confirm that bots are a crucial part of the infrastructure that process payments, expand channels, and provide access to content - both benign and malicious. We posit that future studies on Telegram should take into account the role of bots in supporting these activities; we provide a comprehensive dataset for researchers to do so. We also suggest moderators and law enforcement consider bots as a potential intervention point to disable monetization and expansion. Focusing on bots could be more effective than targeting users or channels simply because the number of bots is significantly smaller than the number of channels or users.

We offer several suggestions to moderate bots. First, our method of collecting the bot's command list and the messages mentioning bots is effective in profiling bot operations (§3.3). Not only is it important to uncover better information, but also 36% of existing bots do not have a description, so it would not be possible to infer their use without interaction. Second, we observed an association between languages and domains (§5.1). Language may indicate geographic location, which could help focus law enforcements' efforts. Third, some bots (e.g., such as those in Finance) are short-lived, and often reused, which makes moderation challenging (§5.2). Telegram could reduce the default number of bots users can create (currently 20) or freeze other bots created by the same user if one of the bots is flagged (i.e., guilt-by-association). Clustering bots with similar description and usernames could also help uncover more bots. Fourth, purpose-specific bots (as opposed to generic utility ones) tend to serve closely knit communities, which could be another avenue for moderation (§5.3). Finally, bots offering payment and referral functionalities should be treated with caution, especially in the Finance domain (§4.5).

6.2 Limitations

Our dataset may not constitute a representative sample of Telegram. Our dataset does not contain any private channels that require invitations, which likely causes us to underestimate Underground domains. Our bot sample, extracted from channels, groups, and third-party lists, may also be biased towards popular bots. Lastly, we only send universal commands to bots and do not have any multi-stage interaction. For instance, we could not identify the types of payment methods or AI models at scale because some bots only disclose them after a few interactions. Beyond the basic commands, Telegram bots accept a wide range of input methods, from inline queries (triggered from anywhere in Telegram) to buttons, plain text, and non-textual inputs such as images or location data. Multi-stage interaction also requires to mange the conversation state/flow, which will significantly increase the data collection costs and potentially limit the number of bots we could interact with. This study prioritized a broader sample over in-depth exploration. Future work could potentially use LLMs to have a longer conversation with bots while adhering to any ethical concerns discussed below.

6.3 Ethics statement

We only collected publicly accessible data on Telegram using their official APIs, complying with Telegram's ToS. The Telegram API allows reading channel public messages without joining them, minimizing impact on the community. We did not collect any private channels and did not attempt to deanonymize users. Under such circumstances, informed consent may be waived (British Society of Criminology 2015).

To avoid downloading malicious or sensitive information, we did not collect any media files. However, our dataset contains some problematic channels, such as financial scams or underground services. Given the size of the dataset (up to 700 GB), we cannot eliminate the possibility that some users publicly disclose sensitive information (e.g., personally identifiable information). Malicious actors could also misuse our dataset to find bots that facilitate illicit activities. In this way, our data was released, guided by FORCE11

(2020) and Gebru et al. (2021), under CC Attribution No Derivatives 4.0 International License, but made only available to researchers who agree to our data usage policy through Zenodo[6] to minimize such harms.

While most of our data collection is passive (e.g., reading messages from channels), we interact with bots. We consider bots as public API endpoints and only use Telegram's official API to send commands. We limit ourselves to sending two basic commands (e.g., `/start`, `/help`), which, we believe, falls within acceptable bot use. The IRB (i.e., ethics committee) at our institution only reviews human subject studies, which does not apply to our case.

7 Conclusion

This paper is the first to look at the programmable aspects of Telegram, namely *bots*. We develop a novel system to continuously collect Telegram messages and interact with bots. This collected data enables us to analyze their functionalities and usage patterns at scale. Despite many legitimate use cases (e.g., crowdsourcing), some bots help illicit activities such as financial scams, cybercrime, and non-consensual image sharing. This paper alerts various stakeholders, including researchers, Telegram, and law enforcement, to recognize bots as an emerging software infrastructure.

Acknowledgement

We confirm that all text in this paper was written by the authors. AI-based writing assistants (e.g., Grammarly, Claude) were used solely for grammar and spelling checks and to improve the clarity of the author-written text. We sometimes used Copilot for automatic code completion (primarily for figures) to reduce typing errors. However, we manually verified all lines of codes.

This work was supported by the Cylab Presidential Fellowship, CyLab Security and Privacy Institute seed funding grant, the Nakajima Foundation, King's College Cambridge, and the Cambridge Trust, European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant No 949127.

References

Alrhmoun, A.; Winter, C.; and Kertész, J. 2023. Automating terror: The role and impact of telegram bots in the Islamic State's online ecosystem. *Terrorism and Political Violence*, 36(4): 409–424.

Baumgartner, J.; Zannettou, S.; Squire, M.; and Blackburn, J. 2020. The Pushshift Telegram dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, 840–847.

Bawa, A.; Kursuncu, U.; Achilov, D.; Shalin, V. L.; Agarwal, N.; and Akbas, E. 2025. Telegram as a Battlefield: Kremlin-Related Communications During the Russia-Ukraine Conflict. *Proceedings of the International AAAI Conference on Web and Social Media*, 19: 2361–2370.

Bijmans, H.; Booij, T.; Schwedersky, A.; Nedgabat, A.; and van Wegberg, R. 2021. Catching phishers by their bait: Investigating the Dutch phishing landscape through phishing kit detection. In *30th USENIX Security Symposium (USENIX Security 21)*, 3757–3774.

Blas, L.; Luceri, L.; and Ferrara, E. 2025. Unearthing a billion Telegram posts about the 2024 US presidential election: Development of a public dataset. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, 729–732. New York, NY, USA: Association for Computing Machinery. ISBN 9798400713316.

British Society of Criminology. 2015. Statement of ethics. https://www.britsoccrim.org/ethics/.

Burgess, M. 2020. Telegram still hasn't removed an AI bot that's abusing women. https://www.wired.com/story/telegram-still-hasnt-removed-an-ai-bot-thats-abusing-women/. Accessed Sep. 24th, 2025.

Büyükkaya, A. 2024. ONNX Store: Phishing-as-a-service platform targeting financial Institution. https://blog.eclecticiq.com/onnx-store-targeting-financial-institution. Accessed Sep. 23rd, 2024.

Cernera, F.; La Morgia, M.; Mei, A.; and Sassi, F. 2023. Token spammers, rug pulls, and sniper bots: An analysis of the ecosystem of tokens in Ethereum and in the Binance Smart Chain (BNB). In *32nd USENIX Security Symposium (USENIX Security 23)*, 3349–3366.

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20: 37–46.

Danilk, M. 2025. langdetect 1.0.9. https://pypi.org/project/langdetect/. Accessed Oct. 7th, 2025.

De Oliveira, J. C.; Santos, D. H.; and Neto, M. P. 2016. Chatting with Arduino platform through Telegram bot. In *2016 IEEE International Symposium on Consumer Electronics*. IEEE.

FORCE11. 2020. The FAIR data principles. https://force11.org/info/the-fair-data-principles/.

Franco, M.; Gaggi, O.; and Palazzi, C. E. 2024. Characterizing non-consensual intimate image abuse on Telegram groups and channels. In *Proceedings of the 4th International Workshop on Open Challenges in Online Social Networks*, 26–32.

Gangopadhyay, S.; Dessi, D.; Dimitrov, D.; and Dietze, S. 2025. TeleScope A longitudinal dataset for investigating online discourse and information interaction on Telegram. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, 2423–2433.

Gao, B.; Wang, H.; Xia, P.; Wu, S.; Zhou, Y.; Luo, X.; and Tyson, G. 2020. Tracking counterfeit cryptocurrency end-to-end. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(3): 1–28.

Gebrekidan, S.; and Dong, J. 2025. The scammer's manual: how to launder money and get away with it. https://www.nytimes.com/2025/03/23/world/asia/cambodia-money-laundering-huione.html. Accessed Sep. 24th, 2025.

[6]The request for the dataset can be made at https://zenodo.org/records/17281308 after logging into your Zenodo account

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Gibson, C.; Olszewski, D.; Brigham, N. G.; Crowder, A.; Butler, K. R.; Traynor, P.; Redmiles, E. M.; and Kohno, T. 2025. Analyzing the AI nudification application ecosystem. In *34th USENIX Security Symposium (USENIX Security 25)*.

Guo, Y.; Wang, D.; Wang, L.; Fang, Y.; Wang, C.; Yang, M.; Liu, T.; and Wang, H. 2024. Beyond app markets: Demystifying underground mobile app distribution via Telegram. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 8(3).

Han, C.; Li, A.; Kumar, D.; and Durumeric, Z. 2025. Characterizing the MrDeepFakes sexual deepfake marketplace. In *34th USENIX Security Symposium (USENIX Security 25)*.

Hanley, H. W.; and Durumeric, Z. 2024. Partial mobilization: Tracking multilingual information flows amongst Russian media outlets and Telegram. In *Proceedings of the International AAAI Conference on Web and Social Media*, 528–541.

Imperati, V.; La Morgia, M.; Mei, A.; Mongardini, A. M.; and Sassi, F. 2025. The conspiracy money machine: Uncovering Telegram's conspiracy channels and their profit model. *34th USENIX Security Symposium (USENIX Security 25)*.

Kireev, K.; Mykhno, Y.; Troncoso, C.; and Overdorf, R. 2025a. Characterizing and detecting propaganda-spreading accounts on Telegram. *34th USENIX Security Symposium (USENIX Security 25)*.

Kireev, K.; Mykhno, Y.; Troncoso, C.; and Overdorf, R. 2025b. A Telegram dataset of propaganda and its moderation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, 2510–2518.

La Morgia, M.; Mei, A.; and Mongardini, A. M. 2025. TG-Dataset: Collecting and exploring the largest Telegram channels dataset. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD'25, 2325–2334. New York, NY, USA: Association for Computing Machinery.

Lieber, R. 2023. Stolen checks are for sale online. We called some of the victims. https://www.nytimes.com/2023/12/09/business/stolen-checks-telegram.html. Accessed Sep. 24th, 2025.

Marjanov, T.; and Hutchings, A. 2025. SoK: Digging into the digital underworld of stolen data markets. In *2025 IEEE Symposium on Security and Privacy (SP)*, 1–18. IEEE.

Marjanov, T.; Tsuchiya, T.; Ioannidis, K.; Hughes, J.; Christin, N.; and Hutchings, A. 2026. Stayin' Alive: How Global Stolen Data Markets Thrive on Telegram. In *Proceedings of the 35th USENIX Security Symposium (USENIX Security'26)*.

Mirtaheri, M.; Abu-El-Haija, S.; Morstatter, F.; Ver Steeg, G.; and Galstyan, A. 2021. Identifying and analyzing cryptocurrency manipulations in social media. *IEEE Transactions on Computational Social Systems*, 8(3): 607–617.

Mozur, P.; Satariano, A.; Krolik, A.; and Myers, S. 2024. How Telegram became a playground for criminals, extremists and terrorists. https://www.nytimes.com/2024/09/07/technology/telegram-crime-terrorism.html. Accessed Sep. 24th, 2025.

Ng, L. H. X.; Kloo, I.; Clark, S.; and Carley, K. M. 2024. An exploratory analysis of COVID bot vs human disinformation dissemination stemming from the Disinformation Dozen on Telegram. *Journal of Computational Social Science*, 7: 695–720.

Nigam, R.; and Wilhoit, K. 2018. User Telegram bot to remotely control infected devices by malware. https://unit42.paloaltonetworks.com/unit42-telerat-another-android-trojan-leveraging-telegrams-bot-api-to-target-iranian-users/. Accessed Sep. 23rd, 2024.

Nikkhah, S.; Miller, A. D.; and Young, A. L. 2018. Telegram as an immigration management tool. In *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 345–348.

Nizzoli, L.; Tardelli, S.; Avvenuti, M.; Cresci, S.; Tesconi, M.; and Ferrara, E. 2020. Charting the landscape of online cryptocurrency manipulation. *IEEE Access*, 8: 113230–113245.

Perlo, A.; Paoletti, G.; Jha, N.; Vassio, L.; Almeida, J.; and Mellia, M. 2025. Topic-wise exploration of the Telegram group-verse. In *Companion Proceedings of the ACM on Web Conference 2025*, 1792–1801.

Ricaldi, R.; Marjanov, T.; Allodi, L.; and Hutchings, A. 2025. Uncovering the Trust Signals Supporting Telegram's Cybercrime Economy. In *2025 APWG Symposium on Electronic Crime Research (eCrime)*, 1–17. IEEE.

Roy, S. S.; Vafa, E. P.; Khanmohamaddi, K.; and Nilizadeh, S. 2025. DarkGram: A large-scale analysis of cybercriminal activity channels on Telegram.

Semenzin, S.; and Bainotti, L. 2020. The use of Telegram for non-consensual dissemination of intimate images: Gendered affordances and the construction of masculinities. *Social Media+ Society*, 6(4).

Steffen, E. 2025. More than Memes: A Multimodal Topic Modeling Approach to Conspiracy Theories on Telegram. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, 1831–1844.

Telegram. 2025a. Telegram bot features. https://core.telegram.org/bots/features. Accessed Sep. 19th, 2025.

Telegram. 2025b. Telegram browser, mini app store, gifting stars and more. https://telegram.org/blog/w3-browser-mini-app-store. Accessed Sep. 19th, 2025.

Telegram. 2025c. Telegram FAQ. https://telegram.org/faq/. Accessed Oct. 4th, 2025.

Telegram. 2025d. Telegram login widget. https://core.telegram.org/widgets/login. Accessed Sep. 19th, 2025.

Telegram. 2025e. Telegram passport blog. https://telegram.org/blog/passport. Accessed Sep. 19th, 2025.

Telegram. 2025f. Telegram stars: pay for digital goods and more. https://telegram.org/blog/telegram-stars. Accessed Sep. 19th, 2025.

Telegram. 2025g. TON: the open network. https://ton.org/. Accessed Sep. 19th, 2025.

Xu, J.; and Livshits, B. 2019. The anatomy of a cryptocurrency Pump-and-Dump scheme. In *28th USENIX Security Symposium (USENIX Security 19)*, 1609–1625.

Paper Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes, we never deanonymize users in our Telegram dataset.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes.
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes, in §2.3 and §3.
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes, our dataset and analysis code.
   (e) Did you describe the limitations of your work? Yes, in §6.2.
   (f) Did you discuss any potential negative societal impacts of your work? Yes, in §6.3.
   (g) Did you discuss any potential misuse of your work? Yes, in §6.3.
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? Yes, in §6.3.
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.
2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? NA
   (b) Have you provided justifications for all theoretical results? NA
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? NA
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? NA
   (e) Did you address potential biases or limitations in your theoretical framework? NA
   (f) Have you related your theoretical results to the existing literature in social science? NA
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? NA
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? NA
   (b) Did you include complete proofs of all theoretical results? NA
4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? NA
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? NA
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? NA
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? NA
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? NA
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? NA
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
   (a) If your work uses existing assets, did you cite the creators? Yes, we used two existing datasets (in §3.1).
   (b) Did you mention the license of the assets? Yes in §6.3.
   (c) Did you include any new assets in the supplemental material or as a URL? Yes, the code and the dataset are released for researchers (§6.3).
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? Yes, in §6.3.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Yes, in §6.3.
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see FORCE11 (2020))? Yes, in §6.3.
   (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see Gebru et al. (2021))? Yes, we discussed in §6.3.
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
   (a) Did you include the full text of instructions given to participants and screenshots? NA
   (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? Yes, our work is not considered human subject research.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA
   (d) Did you discuss how data is stored, shared, and deidentified? Yes, in §6.3

A Appendix

A.1 Bot domain categorization prompt

Choose one category that fits the Telegram bot best from the list below. Do not create a new category.

Admin Tools: Bots that manage groups on behalf of owners, such as membership management, question answering (e.g., using as a point of contact), content moderation, and group statistics.

Content & Media: Bots that help distribute or collect educational & training materials, streaming (music, movies, TV series), and news media, but do not include ones that are from underground markets.

Ideology: Bots that are used for political campaigns, social movements, or religion-related purposes.

Finance: Bots that provide access to financial services (including cryptocurrencies and NFTs) such as online wallets, trading, airdrops, mining, or providing financial information, but do not include ones with games or gambling components.

Shopping: Bots that facilitate online shopping, including selling and buying products, collecting reviews, providing customer service, and product search, but do not include ones from underground markets.

Social & Gaming: Bots that facilitate online interactions, including chatting for fun (not Q&A), dating, games, and gambling.

Underground: Bots that support underground operations, such as cybercrime (e.g., hacking, stolen data, phishing) and adult content.

Utility: Bots that provide tools or functions to individual users (not groups), such as using LLM endpoints, developer tools, web search, photo & video management, health & fitness management, and QR code generation.

Others: Bots that do not belong to the categories above or are unknown given the input.

For each bot, we will send you the following:

```
{
    "name": <str> the username of the bot which may sometimes
                  contain important keywords,
    "description": <str> the description of the bot,
    "command_list": <str> the list of commands the bot has,
    "start_response": <str> the response message for the /start command,
    "help_response": <str> the response message for the /help command,
    "message": <str> the messages that include the link to the bot
    (they may not necessarily be about the bot).
}
```

Given an input, I want you to choose one category (`category`) and produce a one-sentence summary of the bot (`summary`). Output format:

```
{
    "category": <str>,
    "summary": <str>
}
```

A.2 Bot domain categorization annotation results

	Annotator 1	Annotator 2	GPT-4o	GPT-4o-mini
Admin Tools	0.14	0.02	0.06	0.04
Content & Media	0.04	0.10	0.09	0.09
Ideology	0.06	0.05	0.04	0.05
Finance	0.40	0.39	0.38	0.42
Shopping	0.02	0.04	0.08	0.06
Social & Gaming	0.12	0.13	0.14	0.13
Underground	0.01	0.01	0.01	0.01
Utility	0.08	0.11	0.08	0.11
Others	0.13	0.15	0.12	0.09

Table 2: Annotation results for bot domain categorization (i.e., the distribution of each category).