Pretending to be a VIP! Characterization and Detection of Fake and Clone Channels on Telegram

MASSIMO LA MORGIA, Department of Computer Science, Sapienza University of Rome, Roma, Italy
ALESSANDRO MEI, Department of Computer Science, Sapienza University of Rome, Roma, Italy
ALBERTO MARIA MONGARDINI, Computer Science, University of Rome La Sapienza, Rome, Italy
JIE WU, CIS, Temple University, Philadelphia, United States

Telegram is a widely used instant messaging app that has gained popularity due to its high level of privacy protection. Telegram has standout social network features like channels, which are virtual rooms where only administrators can post and broadcast messages to all subscribers. However, these same features have also led to the emergence of problematic activities and a significant number of fake accounts. To address these issues, Telegram has introduced verified and scam marks for channels, but only a small number of official channels are currently marked as verified, and only a few fakes as scams.

In this research, we conduct a large-scale analysis of Telegram by collecting data from 120,979 different public channels and over 247 million messages. We identify and analyze two types of channels: Clones and fakes. Clones are channels that publish identical content from another channel in order to gain subscribers and promote services. Fakes, on the other hand, are channels that impersonate celebrities or well-known services by posting their own messages. To automatically detect fake channels, we propose a machine learning model that achieves an F1-score of 85.45%. By applying this model to our dataset, we find the main targets of fakes are political figures, well-known people such as actors or singers, and services.

CCS Concepts: • **Information systems → Social networks**; • **Security and privacy → Intrusion/anomaly detection and malware mitigation**.

Additional Key Words and Phrases: Dataset, Telegram, Fake detection, Clone channels

1 Introduction

Telegram is likely the most controversial instant messaging platform. While it gives voice to dissidents in countries without freedom of speech [12], terrorists in Indonesia used Telegram to promote radicalism and provide instructions for carrying out attacks [1]. Neo-Nazi groups leverage Telegram to share their ideologies [4]. The platform has also become a hub for conspiracy theory communities [37] and cryptocurrency traders coordinating large group chats to arrange market manipulations like pump and dump frauds [43]. These activities were carried out by exploiting a distinct social network feature of Telegram: The channels. Channels are virtual rooms where only the administrator can write and broadcast the messages to their subscribers. However, just like what happens with fake accounts on online social networks [25, 58], fake channels are widespread in Telegram. As a fake account, a fake channel impersonates a service or person without authorization. A fake channel, to deceive the users, usually has the exact name of the target or a slight variation of it (*e.g.*, presence of emoji in the title). It attempts to qualify itself as an official using words such as official, real, and verified or adding the verified mark on the profile image. Indeed, by leveraging the popularity and influence of a well-known company or

Authors' Contact Information: Massimo La Morgia, Department of Computer Science, Sapienza University of Rome, Roma, Italy; e-mail: lamorgia@di.uniroma1.it; Alessandro Mei, Department of Computer Science, Sapienza University of Rome, Roma, Italy; e-mail: mei@di.uniroma1.it; ALBERTO MARIA MONGARDINI, Computer Science, University of Rome La Sapienza, Rome, Italy; e-mail: mongardini@di.uniroma1.it; Jie Wu, CIS, Temple University, Philadelphia, Pennsylvania, United States; e-mail: jiewu@temple.edu.

ACM 1559-114X/2024/11-ART
https://doi.org/10.1145/3705014

person, the fake channel quickly obtains a considerable number of subscribers and can begin to perform frauds or scams, spam, or spread new ideologies. Significant cases of fake channels and their dangers were those created to impersonate Coinbase [3] and Kraken [2], two popular cryptocurrency exchange sites. Here, the admins used fake channels to perpetrate scams and account takeovers. Due to the high number of users following fake channels on Telegram, it is urgent to develop specific detection models to alert them about possible malicious behavior. Even more so if we consider that Telegram is becoming more and more popular, and, as we observed, that the initial countermeasures like the verified mark are still underused.

To perform our study, we built two datasets: the TGDataset and the Fake Channel dataset. The first dataset, which we publicly release [40], includes over 120,000 channels gathered over a one-year period, while the second is a manually curated dataset containing only verified and fake channels. We leverage the Fake Channel dataset to understand distinctive features of verified and official channels and train a machine learning model able to detect fake channels with an F1-score higher than 85%. Then, we further assess our model on the English channel of the TGDataset. By performing a qualitative analysis of the discovered fake channels, we are able to determine the most preferred targets and the goals of the admins of the fake channels. Lastly, we analyze the phenomenon of clone channels. While fakes pretend to be an official channel and post messages different from those of the official one, a clone channel is a channel that mimics an official one publishing its exact content. We discover that both kinds of channels are exploited by political movements like QAnon and Sabmyk to spread their conspiracy theories.

Our main contributions are the following:

- **Fake channels characterization**. We study the phenomenon of fake channels on Telegram, performing quantitative and qualitative analyses. Through our study, we are able to understand that fake channels mainly target political figures to spread new ideologies, sell goods and promote other channels. Moreover, we notice that although fake channels usually have fewer subscribers than their official counterparts, they still reach a large audience.
- **Fake channels detection.** We analyze the problem of fake channels detection on Telegram, comparing it with the fake accounts in other Online Social Networks. We propose three machine learning models able to detect fake channels with a weighted F1-score of 85.45%. With the proposed model, we detected 258 allegedly fake accounts in the wild, of which we could confirm 88.
- **Clone channels analysis**. We describe and quantify the presence of clone channels within our dataset, finding 73 clone channels. Analyzing them, we discover that, as fakes, most of them aim to disseminate conspiracy theories.
- **Sabmyk: Conspiracy theory**. Analyzing our dataset as a graph, we identify the 236 channels composing the Sabmyk network. This movement extensively used fake and clone channels to reach a large audience quickly and spread its ideas.

2 Background and Related Work

2.1 Telegram

Telegram is a popular instant messaging platform that started in 2013, with more than 800 million monthly active users as of 2023 [51]. On Telegram, users can share text messages, images, videos, audio, stickers, and files weighing up to 2 GB. Aside from the standard one-to-one messaging, Telegram provides group chats and channels. Both have a unique username on the platform, a title, and a description, and they can be private or public. While groups allow many-to-many messaging (any member can write) and have a limit of 200,000 members, channels provide one-to-many communication (only admins can post content) and unlimited subscribers. Moreover, channels do not show info about the subscribers, except the total number. Although they serve different purposes, private chats, groups, and channels are not isolated but linked through message forwarding. This functionality

allows users and administrator's channels to forward content posted in a chat to a different user, group, or channel showing the author of the original message. In particular, Telegram channels are an effective solution for spreading information to a large pool of people. Indeed, several institutional public figures and companies opened an official Telegram channel to broadcast announcements and news [9]. Likewise, start to pop up on the platform channels aiming to impersonate official channels or leverage Telegram channels and groups to sell fake products or services. Telegram introduced the *verified* and the *scam* marks to face this phenomenon. Channels, groups, and bots can achieve the verified mark proving to Telegram that the profile has the verified status on at least two social media platforms (*e.g.,* TikTok, Facebook, Twitter, Instagram) [13]. Instead, Telegram flags a channel or a group as a scam if several users report it for fraud [14].

2.2 Telegram channels analysis

Several works focused on the Telegram ecosystem or emerging research issues related to it. Hashemi et al. [35] collect Iranian channels and groups on Telegram to identify high-quality groups, such as business groups, among low-quality groups (e.g., dating groups). They show that high-quality groups distinguish themselves from low-quality ones through longer messages and more user engagement. Nobari et al. [30] present a structural and topical analysis of messages posted on Telegram on a dataset of more than 2,000 groups or channels. This study indicates that there is no correlation between the Page Rank of channels or groups and their number of subscribers. Baumgartner et al. [21] publish a dataset of over 27,800 thousand channels and 317 million messages from 2.2 million unique users. Their dataset includes a wide range of right-wing extremist groups and protest movements. In their work, Weerasinghe et al. [56] reveal that Telegram hosts several organized groups, called pods, where each member interacts with each other's content to increase the popularity of their Instagram accounts. Other works [43, 47, 59] reveal a vast presence on Telegram of channels and groups focused on pump and dump, a cryptocurrency market manipulation. Finally, several studies focus on the activity of terrorist organizations, like ISIS, that utilize Telegram for disseminating content and recruiting followers [26, 60].

2.3 Fake accounts on other OSNs

Fake accounts are widespread in Online Social Networks [25, 42, 58]. The meaning of fake account is broad as it indicates deception contained in its content and personal information [29]. Thus, fake accounts represent several types of accounts aiming to deceive a user for different purposes. These goals can be spamming, malware distribution, impersonating people, and creating artificial interaction on the platform, for instance, using bot accounts to increase the followers of the target account [27, 54]. Several works address the problem of fake accounts, especially on Twitter. Ershain et al. [31] study the fake Twitter accounts that do not belong to a real human. They propose a classifier using features based on user behavior, such as the number of tweets, the number of accounts followed, and the number of followers. The underlying idea of their classifier is that humans behave differently. A very similar problem is the one related to Bot detection on Twitter. This task is also addressed in PAN, a series of scientific events and shared tasks on digital text forensics and stylometry [39]. In the PAN context, a classifier can rely only on stylometric features to detect bot accounts, achieving an F1-score higher than 90% on multilingual settings [19]. Instead, Caruccio et al. [27] focus on the problem of fake followers, fake accounts created specifically to increase the number of followers of a target account. The author's technique relies on the Relaxed Functional Dependencies to discriminate fake accounts from real ones. Also do Cresci et al. [29] face the problem of fake followers in Twitter. After evaluating the most relevant features and rules exploited in the Twitter fake accounts detection, they discovered that it is possible to detect with high accuracy fake followers using lightweight features such as profile information and the ratio between followers and following accounts. Gupta et al. [33] addresses the problem of detecting fake accounts on Facebook. The authors propose a classifier based on features related to user activity, such as likes and comments posted, which can detect fake accounts with

Categories retrieved
Sales, Humor & Entertainment, News & Mass media, Video & Movies, Business & Startups, Cryptocurrencies, Politics, Technologies, Sport, Marketing, Economics, Games, Religion, Software & Applications, Lifehacks, Fashion & Beauty, Medicine, Adults

Table 1. The 18 categories to which belong the most popular 100 channels according to Tgstat.

an accuracy of 79%. Bilge et al. [22] shows the threats of fake accounts on Facebook. In this study, the authors forge fake accounts of the target victims using public information. Then, they send a friend request to the victim's contacts from the fake account, observing that the contacted victim trusts the request of the fake account.

In this work, we deal with the problem of detecting fake channels on Telegram that, to the best of our knowledge, was never tackled in literature. At first sight, a Telegram channel could appear very similar to a Twitter account or a Facebook page. However, the Telegram platform mechanics make them substantially different. For instance, a channel can not follow other channels or users, the interaction between channel subscribers and content is very limited, and the content visibility is limited to the channel's subscribers. The differences between Telegram and other OSNs require leveraging different features. We discuss features used in other works related to fake profile detection and their usage in the detection of fake Telegram channels in Sec. 4.2.

3 Data collection

3.1 The TGDataset

Existing Telegram datasets are designed for specific studies. Thus, they contain only channels related to a particular topic [21, 36] or country [35]. Conversely, our work aims to study the phenomenon of fake channels on the Telegram ecosystem. Thus, we need a dataset representing an actual snapshot of Telegram covering many popular and connected channels. For these reasons, we build the TGDataset [41].

Dataset construction. To explore Telegram and, in particular, the most popular and connected channels, we use a snowball approach, as previously done in [21]. We start from a list of seed channels covering different topics and expand the dataset by adding, for every forwarded message in the seed channels, the original channel of the message. To select the seed channels, we leverage Tgstat [8], a popular service that indexes more than 150,000 Telegram channels and collects statistics about them. Although Tgstat does not offer free APIs to collect the indexed channels, it freely reports the rank of the top 100 channels by the number of users. From this rank, we retrieve all the categories to which these channels belong, finding the 18 categories shown in Tab. 1.

Then, we select as seeds the 10 most popular channels by the number of subscribers from each category. Overall, we obtain a total of 180 seed channels. From each seed channel, we download the last 10,000 messages through the Telethon APIs [5], an open-source Python wrapper of the official Telegram APIs. Although a channel can contain more than 10,000 messages, we decide not to download more than that. Indeed, even though Telegram's API does not have a hard limit on the number of messages that can be retrieved, the platform actively discourages the retrieval of large amounts of messages, delaying requests when retrieving more than 3,000 historical messages [15]. Since 10,000 messages cover the entire history of more than 97.84% channels, we prefer to limit the number of requests to avoid flooding the Telegram services with further requests that go beyond our primary goals. After downloading the data, we parse the messages to discover new channels analyzing the forwarded messages. Finally, to further expand the TGDataset, we use the newly discovered channels as new seeds and iterate the above-described procedure.

Data retrieved. Data collection started on 4 January 2021 and ended on 31 July 2022. Overall, the TGDataset is 235 GB in size and contains 247,662,141 messages and 120,979 different channels. Among the channels, 656

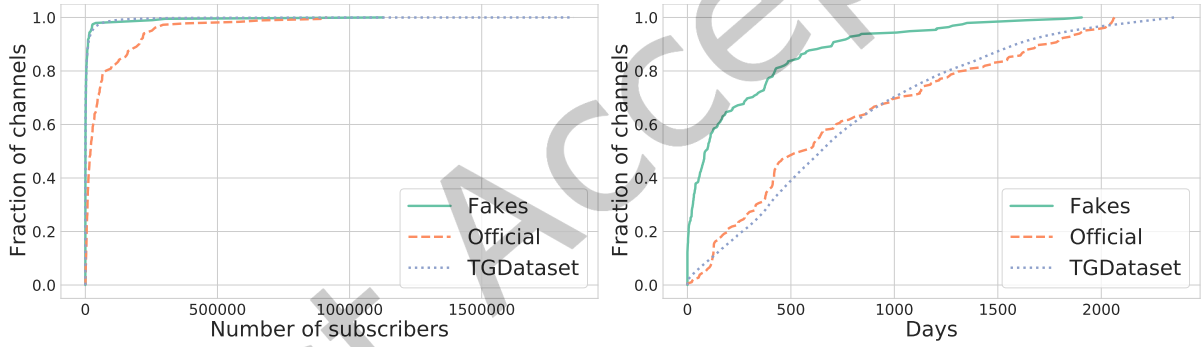
(0.53%) are verified channels, and 184 (0.15%) are scam channels. From each channel, we store the following information: The title, the description, the channelID, the creation date, the number of subscribers, and if it is marked as a scam or verified. Concerning messages, we store the channelID, the timestamp, and, in case of forwarded messages, the original channelID where the message has been posted, and the original posting date. Finally, we store the content of the text messages, while just the title and the file format of the media messages.

3.2 The Fake Channels dataset

To understand the main differences between fake and official channels and later train a machine learning model able to detect fake channels, we build a dataset of channels whose status (official or fake) is known with certainty. To this respect, we create the Fake Channels dataset. To build it, we use the following approach: We first leverage the *Telemetr.io* [10] services to retrieve a list of verified channels. Then, for each verified channel, we look for fake channels claiming to be the official ones, taking care to not select fan channels. At the end of this process, the Fake Channels dataset consists of 342 different channels, 184 of which are officials and 158 fakes. While selecting the channels, we ensure they are not already present in the TGDataset. In this way, we can use the Fake channel dataset as training data while developing our detector.

4 Fake channels detection

4.1 Analysis of the Fake Channels dataset



(a) Subscribers

(b) Lifetime.

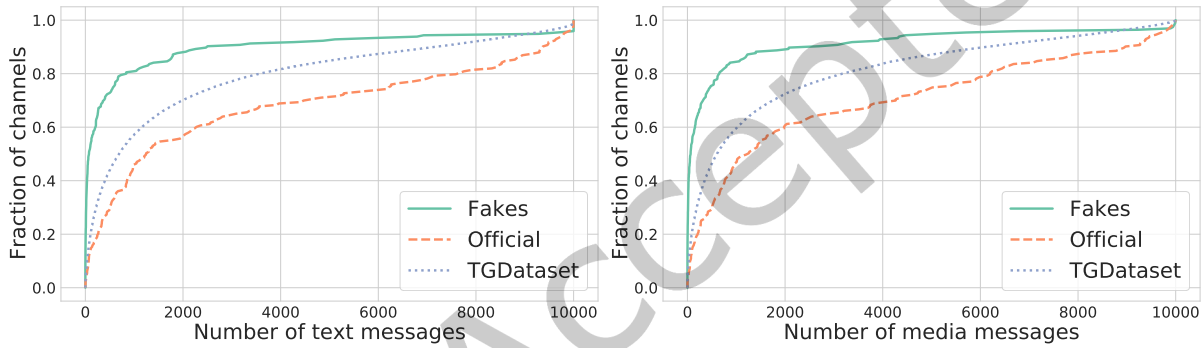
Fig. 1. CDFs of the number of subscribers (1(a)) and the lifetime of the channels (1(b)) for fake, verified and TGDataset channels.

As a first step toward constructing our detector model, we separately analyze the fake and verified channels contained in the Fake Channels dataset, and we use the channels of TGDataset as a reference of the average behavior of the Telegram channels. Although the TGDataset contains verified and fake channels, given its vast number of channels, we believe it can represent very well the behavior of standard Telegram channels.

We start by studying the number of subscribers of the three sets of channels taken into account, showing them in Fig. 1(a). As we can expect, verified channels (dashed orange line), in general, have more subscribers than fake (green line) and standard channels (dotted blue line). In contrast, fake and standard channels have very similar distributions. Comparing the number of subscribers between the verified channels and their fake version, we notice that the fake channels have, on average, 10% of the number of subscribers of the corresponding verified channel. However, in our dataset, we have two cases in which the fake channels have more subscribers than the verified

one. Both cases are related to *@AnuragxCricket*, a channel of the Indian fantasy cricket influencer Anurag Dwivedi. Here, the verified channel has 280,212 subscribers, while its fakes *@AnuragCricket* and *AnuragxCricket_team* have 301,742 and 1,126,330 subscribers respectively. A possible reason behind the success of the first fake could be that it was created on 2019-10-07, more than one year before the verified channel (2021-03-10). Instead, the second and bigger fake channel was created one month after (2021-04-25) the verified one. Thus this abnormal number of subscribers is less explainable. We conjecture that the fake channel achieved this success by leveraging some promotional services or the help of other fake channels, as we notice in Sec. 5.3. However, we can not confirm this suspect as we do not find evidence in our dataset.

Then, we proceed with the lifetime of the channels. We define the lifetime of a channel as the time elapsed between its creation and its last message. As shown in Fig. 1(b), fake channels have a shorter lifetime (average 251.85 days) than verified (average 750.39 days) and standard channels (average 764.02 days), whereas these last two kinds of channels have similar duration. This result suggests that fakes cease to post content at a certain point as they may have been discovered or because they have reached their goals.



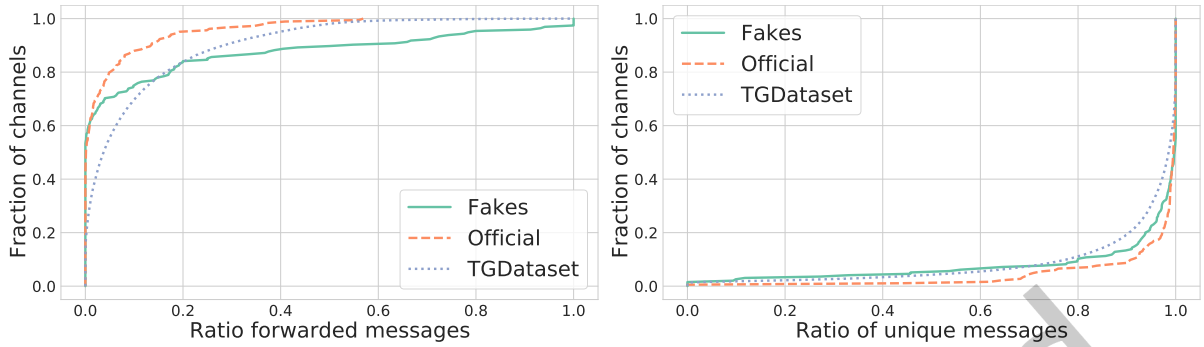
(a) Text-based messages

(b) Media-based messages.

Fig. 2. CDFs of the number of text-based (2(a)) and media-based messages (2(b)) for fake, verified and TGDataset channels.

Finally, we analyze the type of messages shared by the channels. Fig. 2(a) and Fig. 2(b) reveal that verified channels tend to share more messages, both text-based or media-based, than the standard Telegram channels and fake channels. Verified channels post on average 3,176.46 text messages and 2,892.27 media content, while fake and standard channels post 1,036.59 and 2,030.05 text messages and 862 and 1,817.82 media, respectively. The fewer messages shared by fake channels are aligned with their short life. Instead, verified channels have a lifetime similar to standard channels. Thus, the abundant number of content they produce could be a suitable feature for our classifier.

A distinctive feature of fake channels is the number of forwarded messages. Fig. 3(a) shows the ratio between the forwarded messages by the channels and the total number of messages shared. As we can see, while the verified channels tend to forward few messages, fake channels are more prone to forward messages from other channels, with a fraction of fake channels (approx 18%) extensively using this Telegram functionality. Lastly, we investigate the ratio of distinct messages published over the total number of messages published by the channels (Fig. 3(b)). Here, we notice that all three kinds of channels mostly produce fresh content, with both the fake and verified channels more active in producing new content than the standard channels.



(a) Forwarded messages

(b) Unique messages.

Fig. 3. CDFs of the number of forwarded messages (3(a)) and the ratio of unique messages (3(b)) for fake, verified and TGDataset channels.

4.2 Features

As we saw in Sec. 2.3, the topic of fake accounts has been widely studied in OSNs, particularly on Twitter and Facebook. However, Telegram channels, despite having some common traits with OSNs' accounts, present limited social interaction functionalities. A key difference is that in OSNs, an account can interact with others, such as commenting content of other accounts, following other accounts, appreciating content generated by other users (*e.g.,* likes), and republishing content (*e.g.,* retweeting). Instead, a Telegram channel can only post content in its channel and can not interact with anyone outside of it (*e.g.,* subscribing to other channels or texting private messages to users). Moreover, it is virtually impossible to interact with the content generated by the channels. Indeed, even if Telegram recently added the functionality to comment or react with emoticons to the content of a channel, we observe that this feature is enabled only by a tiny fraction of channels. Unlike other OSNs, Telegram discloses only the number of channel subscribers, not the list of subscribing accounts. These differences make unavailable the use of the most discriminating features to detect fake accounts on other OSNs, such as the ratio between the number of users following the account (usually low) and the number of users followed by the fake account (usually high) [27, 29] or the number of likes (given or received). Some features are unique to a particular platform (*e.g.,* Twitter list or usage of Facebook application) and, therefore, cannot be used in our scenario. Nevertheless, we can adapt some features used in the previous works (e.g., biography could be considered the description of a channel) on Telegram channels and evaluate them in our scenario. Tab. 2 shows the main features used by the works focused on detecting fake accounts on other OSNs, a description of them, and if they can be reproduced.

Regarding the other classification work on Telegram [35], the authors focus on detecting high-quality groups. Even in this case, we cannot utilize all their features due to differences between the channels and groups. In groups, every user can post a message like in a chat room, the list of group members is accessible, and the personal accounts of group administrators are disclosed. Conversely, in channels, only the administrator can post, and the accounts of both subscribers and channel administrators are not visible.

To build our classifier to detect fake channels, we evaluate all the previous features and reproduce them in the context of Telegram channels. Moreover, we consider also what we learned in the previous subsection (*e.g.,* number of text messages published, ratio of forwarded messages) and new features specifically for this task. We tried several sets of features to build our model. In the following, we describe the features that achieved the best performance.

Feature	Description	Works	Available
Profile information			
Profile image	The profile has an image	[17, 29, 35]	Y
URL in profile	The profile contains an URL	[29]	Y
Biography	The profile has a biography	[29, 35]	Y
"bot" in profile	The profile contains the word bot	[29]	Y
Address in profile	The profile contains a physical address	[29]	Y
Belong to/follow Twitter list	Twitter lists followed by the account and lists to which it belongs	[17, 29]	N
Verified account	The account is verified	[35]	Y
Private account	The account is private	[35]	N
Intra-platform interaction			
# messages/tweet	Number of messages or tweets published	[29]	Y
Account age	Account activity period	[29]	Y
# hashtags per message/tweet	Average number of hashtag per message or tweet	[17, 33, 35]	N
# unique hashtags	Number of unique hashtags	[17, 33, 35]	N
# char per message/tweet	Average number of characters per message or tweet	[17]	Y
# images	Number of images published	[17]	Y
# messages sent at the same time	Number of messages sent at the same time	[17]	Y
Avg post liked (received/given)	Average number of post liked received and average number of likes given	[33]	N
Avg post comment (received/given)	Average number of post comments received and of post commented	[33]	N
Ratio # friends and # followers	Ratio between friends and followers of the account	[17, 29, 35]	N
# friends	Number of friends of the account	[17, 29, 35]	N
# followers	Number of followers of the account	[17, 29, 35]	Y
# mentions in messages/tweet	Average number of mentions per message or tweet	[17, 35]	Y
# times the account is retweeted	Number of times the account is retweeted	[17, 33]	Y
Favorites/received Account	Number of account favorites and number of favorites received by the account	[17, 35]	N
Cross-platform interaction			
# link	Number of links posted	[17, 35]	Y
# app used	Number of apps used	[33]	N

Table 2. Features used in previous works to detect fake accounts on other OSNs.

- **Writing style features:** average message length, average number of emojis per message, average number of non-alphanumeric characters per message, number of characters in the title and description, and average number of non-alphanumeric characters in the channel's title.
- **Temporal features**: number of text messages published in the last 3, 6, 9 months, and average posting time between two consecutive messages.
- **External interaction features**: number of forwarded messages, standard deviation of the number of source channels for the forwarded messages, number of shared links, and number of duplicate messages containing at least one link.

4.3 Classifiers and results

We use the features described above to train three different models: a Random Forest classifier [23], an SVM with Linear kernel [50], and a Multilayer Perceptron (MLP) [32]. Moreover, to better assess our models, we implement two baselines. Since there are no studies dealing with fake Telegram channels, we select as the first baseline the Twitter fake account classifier that leverages the highest number of features that can also be implemented on Telegram. It is the classifier proposed by Cresci et al. [29], which uses nine adaptable features. As the second baseline, we chose the classifier of Hashemi et al. [35] to detect high-quality groups on Telegram. Also in this case, we use only the available features on Telegram channels. To implement all the models except for the MLP, we use the Sklearn [49] Python library and tune the hyper-parameter through grid search. Instead, to implement the MLP classifier, we use Pytorch [48]. The MLP classifier is made of three linear layers with Rectified Linear Unit function (ReLU) [34] as the activation function, the Adam optimization algorithm [61] as the optimizer, and binary cross-entropy (BCE) [45] as the loss function.

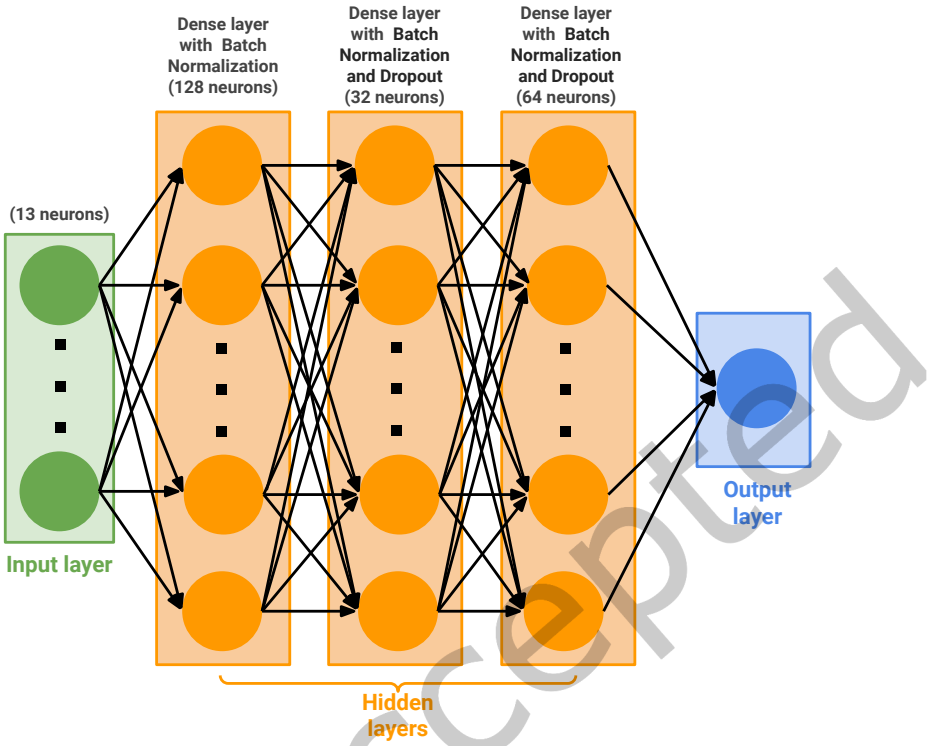


Fig. 4. Architecture of the MLP classifier.

Specifically, the neural network model has an input layer with 13 neurons for the identified features, followed by three dense hidden layers with 128, 32, and 64 neurons, respectively. In particular, all the hidden layers present a batch normalization, while the second and the third also include a dropout (with a rate of 0.10). Finally, the output layer consists of a single neuron with a Sigmoid activation function. Fig. 4 provides a detailed illustration of the architecture of the MLP classifier.

We assess the models' performances through 5-fold cross-validation [18] using the weighted F1-score as the evaluation metric. Table 3 reports the results we achieve by the 5 different models. As we can see, the models based on the proposed features outperform the two baselines. The model that performs worst, slightly better than a random classifier (54.54% F1-score), is the one replicating the results of Cresci et al.. This result is quite expected, given the differences between the Twitter and Telegram platforms. Instead, the model proposed by Hashemi et al. achieves a weighted F1 score of 72.16%. Through the analysis of the results, it is possible to note that the precision (66.94%) and the recall (85.68%) of this classifier are unbalanced. This is due to the model's tendency to classify channels as fakes. Inspecting the weight of the features, we observe that the classifier assigns a high weight to the number of subscribers, leading to classify as fake channels with a low number of subscribers. Finally, we have the three different classifiers based on the features proposed in this work. The MLP model is the classifier that performs better, achieving an F1-score of 85.45%, outperforming the best baseline of 13 percentage points, and obtaining a good trade-off between precision and recall. Instead, both the Random Forest and the

Model	Precision	Recall	F1 weighted	Accuracy
Cresci et al.	52.94%	56.25%	54.54%	55.07%
Hashemi et al.	66.94%	85.68%	72.16%	72.79%
Random Forest	82.05%	81.03%	80.35%	81.03%
SVM linear	81.77%	81.06%	81.01%	81.62%
MLP	84.24%	85.86%	85.45%	85.49%

Table 3. 5-fold cross validation classification results.

SVM model perform slightly worst than the MLP model, achieving an F1-score of 80.35% and 81.01%, respectively, but better than the baselines.

5 Discovering fake channels in the wild

Selection of suspicious channels. After validating our classifier, we leverage it to detect fake channels on the TGDataset. For this task, we consider only English channels, so that we can validate the channels and perform qualitative analysis. To select English channels, we perform language detection. To this end, we pre-process the messages by normalizing and polishing them. In particular, for each channel, we take into account only the pure text messages, remove mentions and get rid of numbers, hyperlinks, emoji, and messages shorter than 15 characters as they could compromise the accuracy of the tool [20, 53]. Then, we tokenize the messages using the *RegexpTokenizer* developed by NLTK [6] and provide them as input to the tool. At this point, to detect the languages of the channels, we leverage LangDetect [52], a language detection library implemented by Google with precision over 99% for 53 languages. At the end of the process, we get 21,078 English channels that account for 17.54% of the TGDataset. Hence, we collect the channels that have in their title, description, or username the words *real*, *official*, or *verified*. To further expand the dataset, we consider all the channels with a similar name (edit distance less than 2) to one of the verified channels. Also in this case, we manually inspect these channels to ensure they are not fan channels. In the end, we collected a set of 511 channels.

Channels evaluation. Since we do not have a ground truth for this set of channels, we check all of them manually to assess the results. In particular, we consider a channel:

- **Official**: if Telegram marked it as verified or there exists an official source (*e.g.,* Website, Facebook, Instagram, Twitter) of the person/service indicating the Telegram channel as the official one.
- **Fake**: if there is another channel that we consider official with the same name or an official source states that there is no official Telegram channel.
- **Allegedly fake/official**: if our classifier detects the channel as fake/official, but there is no evidence of their status. In particular, there are no channels with the same or a similar name that we consider official and the related official web pages or social media pages do not mention any Telegram channel.

Results. Tab. 4 reports the results we obtain after the manual investigation. Globally, we mark as fakes or officials 228 channels out of 511. In particular, among the 258 channels recognized as fakes by our model, there are 88 fakes, 142 allegedly fakes, and 28 official. Among the channels classified as official, 103 are actual official channels, 141 are allegedly official, and 9 are fakes. Thus, for the channels we have evidence of their status, our classifier was able to classify 191 channels out of 228 correctly, equivalent to an accuracy of 83.77%, which aligned with the results obtained in the cross-validation.

	Label			
Prediction	Fake	Official	All. fake	All. official
Fake	88	28	142	0
Official	9	103	0	141

Table 4. Results of the MLP classifier on the TGDataset.

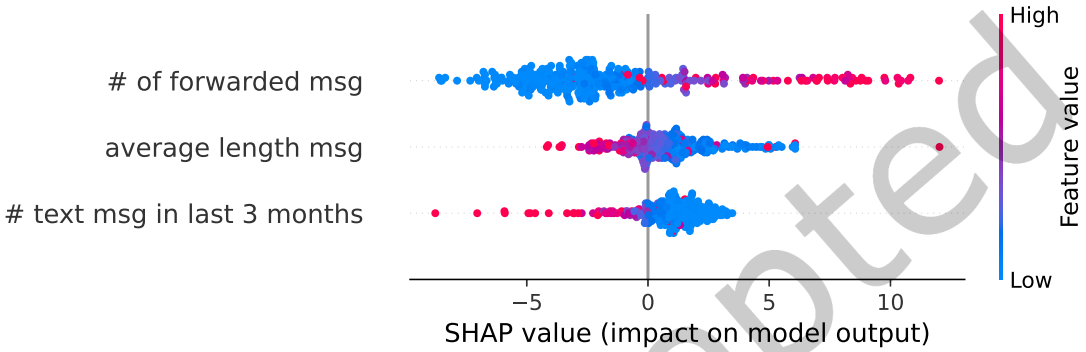


Fig. 5. SHAP values of the 3 most contributing features.

5.1 Features analysis

To understand which features are more relevant to our model, we use the Shapley Additive Explanations (SHAP) value [44]. It determines the contribution of each feature based on game theory principles and local explanations. Fig. 5 shows the SHAP values of the three features that contribute the most to the model's predictions.

According to the SHAP value, the three most significant features are the number of forwarded messages posted within a channel, the average length of text messages posted, and the number of text messages posted in the last 3 months. Interestingly, a high number of forwarded messages suggests to the model that the channel is fake. Indeed, as seen in Sec. 4.1, fake channels tend to forward more messages than official ones. Instead, a high average of message length led the model to flag a channel as official. This behavior reflects that official channels generally post more lengthy and elaborate text messages (average 339.19) than fakes (average 287.71). Moreover, a large number of posts published in the last three months inclines the model to consider a channel as an official. The cause could be that some fake channels, unlike the official ones, tend to have a short life of activity, as shown in Sec. 4.1.

5.2 Misclassification analysis

We leverage the SHAP force plots to understand the main features that drive the model to wrong predictions. They indicate the contribution of each feature in pushing the classifier to its predictions. Fig. 6 shows three explicative examples of SHAP force plots. In particular, we report two false positive and one false negative instances. Analyzing the force plots of the channels classified wrongly, we discover that the two main features driving the model to misclassify official channels as fakes are the rate of non-alphanumeric characters in the title and the number of forwarded messages. Concerning the first feature, we find eight official channels of political figures, including in their titles many emojis, such as the American flag and the thunderbolt icon (both included

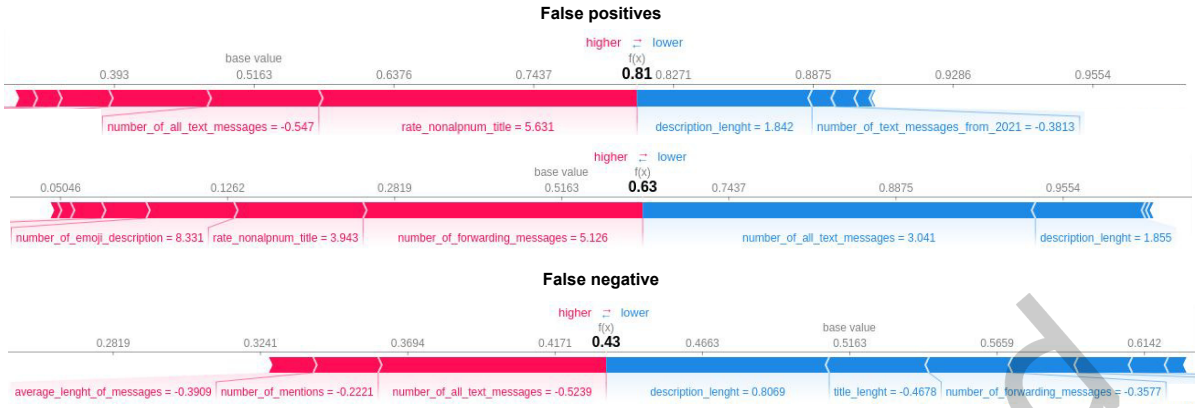


Fig. 6. SHAP Force plots for two false positive and one false negative instances.

in the title of the official channel of Sydney Powell). However, using many emojis in the title is a habit of fake channels to attract users (especially including those emojis that mimic the verified channel symbol). About the second feature (the number of forwarded messages), we recall that a high number of forwarded messages is a characteristic of fakes, as shown in Sec. 4.1. Nevertheless, nine official channels dealing with conspiracy theories tend to forward many messages, leading our model to misclassify them.

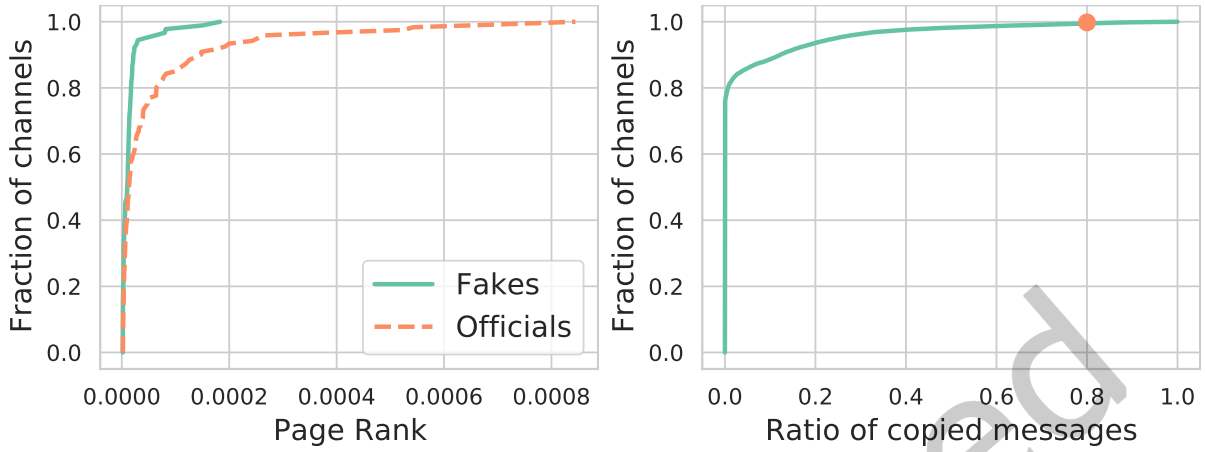
Conversely, the description and title length are two features that drive the model to classify some fakes as officials. Indeed, one distinguishing aspect between officials and fakes is the greater description length of the first ones (since it contains more personal information and links to other social platforms and websites). Further, the titles of fake channels often include the words *real*, *official*, or *true* in an attempt to emphasize their (false) official status, thereby making their titles longer than those of real official channels. Anyway, four fake channels, like the one targeting Michael Flynn, present a detailed description of the impersonating person, similar to what the official channels do, and their titles do not incorporate words to stress their false official status. Moreover, those channels forward only a few messages and have a higher lifetime (more than one year) if compared to that of most fakes.

5.3 Studying fake channels

Fakes targets. The majority of the channels we verified to be fake target real people (76 out of 97). Among them, the most targeted categories are politicians (59), including nine claiming to be Donald Trump, and 17 celebrities (*e.g.*, influencers, actors, and athletes). Moreover, ten fake channels emulate news services, and seven are crypto-related services. Finally, we find four fakes pretending to be well-known companies.

Effectiveness of the fake strategy. A suitable metric for understanding fake channels' effectiveness is to examine the number of subscribers they have attracted. It emerged that the fake strategy is very effective since fakes have an average of 19,636.31 subscribers and more than 45% of them have more than 10,000 subscribers.

The goal of Fake channels. After understanding the target of the fakes, we manually inspect these channels. It turns out that 32 fakes seem to have the goal of spreading conspiracy theories, such as QAnon [57], but also new ones, like Sabmyk [11]. The latter is a conspiracy theory that proposes itself as a better alternative to QAnon and promotes a singular quasi-religion centered around a messianic figure known as Sabmyk [7]. In particular, we find 23 fake channels posting content about Sabmyk that likely belong to a greater network (about a hundred channels) spreading Sabmyk's messages according to the *"HOPE not hate"* organization [11]. In Sec. 7,



(a) PageRank (b) Lifetime.

Fig. 7. 7(a) CDF of PageRank values of fake and official channels of the TGDataset. 7(b) CDF of the ratio of copied messages of each channel.

we explore this network of channels in detail. Other 14 fake channels mainly advertise. There are eight fakes focused on promoting other channels sharing their invitation links and forwarding their messages. Lastly, one fake asks for funds to be sent to a wallet on Monero, a cryptocurrency focused on private and censorship-resistant transactions [46].

Status of fakes and officials. Among the 126 official channels found within the TGDataset, only 70 (55.55%) are marked as verified by Telegram. Nevertheless, there are several channels that we presume are official upon careful manual analysis but that neither appear to be verified by Telegram nor have a link to the channel on their social pages or website. Instead, the fakes marked as scam by Telegram are only 9 out of 82 (8.53%).

Officials are more influential than fakes. It is worth examining whether fake channels have become popular and influential. To this end, we represent the TGDataset as a directed graph $G = (V, E)$ in which nodes in V are the channels, and edge $u \rightarrow v$ in E represents the presence in channel u of a message originally posted in v and forwarded to u by the admin of channel u. Since the users of channel u can navigate the forwarded message and land on channel v, the edge represents in a natural way the possible flow through channels of a user following forwarded messages. Once built the graph, the next step involves the search of the most influential nodes, i.e., the channels spreading the information more frequently and faster [38]. One of the most popular approaches to identify the influential nodes is to use centrality metrics like PageRank [24, 28]. The idea is to define the channels with the highest PageRank as the most relevant nodes.

Fig. 7(a) shows the CDF of the Page Rank values for the fake and official channels within the TGDataset. It highlights that official channels have a higher Page Rank value than fakes, on average 0.000059 and 0.000015, respectively. However, some fakes reach a reasonably high level of influence, such as a fake channel of Donald Trump, which has a Page Rank value of 0.00018. Nobari et al. [30] noticed that the Page Rank of channels does not help detect high-quality channels (channels with a high number of subscribers and with few spam messages). Nonetheless, Page Rank could be a feature for fake channel detection. However, to use Page Rank, it is required to know the entire graph of the Telegram channels, an arduous task. For this reason, we do not use the Page Rank as a feature in our experiment.

	Distance in hop			
	1	2	3	4+
Officials → Fakes	18	22	4	1
Fakes → Officials	13	34	0	0
Fakes → Fakes	38	19	27	11

Table 5. Number of hops (shortest path distance) required to reach a fake channel from an official one (Officials → Fakes), hops needed for a fake channel to reach an official channel (Fakes → Officials) and another fake channel (Fakes → Fakes).

5.4 Connection between officials and fakes

One intriguing detail to investigate is whether and how official channels are connected to fake channels. It is interesting since they are the most popular (Sec.4.1) and could be considered trusted by the users. To this end, we compute for each official channel the shortest path to reach a fake channel. Moreover, we study the connection, always using the shortest path, between fake channels and the official channel as well as two fake channels. Tab. 5 shows the results we obtained.

Officials → Fakes. Looking at the shortest path between official and fake channels (Tab. 5), we notice that most official channels are very close to fake ones. Indeed, 40 (30.5%) official channels reach 70 different fake channels with at most two hops. These results show that it can be really easy for a user of an official channel to navigate to a fake channel. In particular, 18 official channels are at only one hop of distance from at least a fake, overall connecting to 21 fake channels. These 18 official channels belong to members of the American Republican Party or are related to it. Among those, the official channel connected to the largest number of fakes is the channel *Blessed2teach*. It forwards messages from six fake channels targeting American right-wing political figures. Instead, the fake channel most forwarded is the fake pretending to be the American politician Marjorie Taylor Greene, with five official channels forwarding its posts. A possible reason for the behavior of these official channels could be that they forward messages of the fakes to alert their users. However, by examining the messages, we find that none of them act in this way. So, they forward messages from fakes, maybe unaware of their nature.

Examining the connection between the 22 officials at two hops from a fake channel (overall connecting 49 fakes), we observe that their connection with fakes mainly relies on three standard channels that act as a hub. The most relevant hub is the *Midnight Rider Channel* with over 151,000 subscribers. Its messages are forwarded by 15 official channels, while it forwards messages from the fake channels of the Right Side Broadcasting Network.

Fakes → Official. Analyzing how fakes are connected to officials, we find that 13 fake channels directly forward messages from 15 official ones. As in the previous case, these fakes pretend to be republican politicians and forward messages from the official channels of other famous right-wing figures. Looking at the channels at two hops distance, the most relevant hub between the fakes and the officials is the channel on the news about the British far-right activist Tommy Robinson. This channel, with over 150,000 subscribers, eases the navigation of the users from 25 fake channels to 4 different Republican official channels.

Fakes → Fakes. There is also a strong connection between fake channels: 38 (39.17%) of them are only one hop away from another fake channel. Upon further investigation, we discovered that 24 of them compose a complete graph (*i.e.,* each pair of channels is connected by an edge). Moreover, we find that all these interconnected channels forward messages from a standard channel, while the latter never forwards messages from other channels. Reading the content of the messages, it turns out that all these channels aim to spread the Sabmyk conspiracy theory (see Sec. 7), and the standard channel itself is entitled *Sabmyk*.

6 Clone channels

While investigating the fakes, we notice pairs of channels posting identical messages. Clearly, the actual creator of the content is only one of the two, and we refer to it as the original channel. Instead, we call *clone channels* those that publish the exact content of the original one. To understand the reasons behind the creation of a clone channel and how common this phenomenon is, we examine the English channels of the TGDataset.

6.1 Detection of clones

To find the clone channels, we compare the messages of each channel with those of all other channels. To avoid messages that could be coincidentally identical, we only take into consideration messages longer than five words and do not consider forwarded messages or messages indicating Telegram violated terms (*e.g., "This channel can't be displayed because it violated Telegram's Terms of Service"*). Finally, we analyze the distribution of copied messages in our dataset (Fig 7(b)). As we can see, more than 90% of channels have less than 10% identical messages in common with other channels. To find the clones, we restrictively select the tail of the CDF (the orange dot in the figure) that represents the channels with 80% or more identical messages with another channel. We also consider channels with a ratio lower than 100%, as some clones could start posting content of their own when they reach a reasonable number of subscribers. We consider the channel B a clone of the original channel A if, for each common message, the one of B has a publication date later than that of A. With this approach, we find 73 clone channels.

6.2 Analyzing clones

Manually investigating the English channels, we find that the target of a clone is often the official channel of a celebrity or service. In particular, five clones have a different name with respect to the original channels, but they post all the messages of the original ones. Moreover, they interleave the original messages with links to an external platform to buy goods (*e.g.,* books, microwaves) or links to join other channels. For instance, we find a clone of a cryptocurrency-related channel that promotes another channel that arranges pump and dump operations [43]. Five channels clone a celebrity's official one and have a similar name. These clones post additional messages with controversial political content, such as anti-vaccine campaigns. Then, we find a group of 10 channels cloning channels of politicians close to Donald J. Trump or Republican news channels. In this case, all the messages not taken from the original channels promote the same Trump product (*e.g.,* Trump coin) of fake channels. There are also two perfect clones with the same content, title, description, and profile image as another channel. These two channels copied the original channel for weeks and then started to post messages about Sabmyk (see Sec. 5.3). We also find 13 channels cloning fake channels that spread conspiracy theories. Interestingly, we find four clones that, as the original channel, post books protected by copyright. We believe that the admin of the clones is the same as the original channel and uses the clones as a backup of the material shared. If this is the case, this technique appears to be effective. Indeed, checking the original channel a month after the data collection, we found that Telegram removed its content while the clones continued their activity. Finally, concerning the other clones, we notice nothing suspicious other than being clones. However, it is crucial to remark that they are the clones with fewer subscribers (less than 1,000). Thus, they could not have awakened yet, or the admin stopped his cloning activity, as we found in one case. Through the analysis of their behavior, it is clear that the goal of clone channels is to take advantage of the popularity and content generated by the original channel to gain subscribers and promote other services. The clone strategy is very effective. Indeed, the average number of subscribers of the clone channels is 7,033.35. The larger clone channel is the one targeting the official channel of Lin Wood, with 75,011 subscribers. It is not surprising since, in this case, the clone and the official channel are virtually indistinguishable without knowing the channel's username.

7 A case study: SABMYK

Analyzing the fake channels detected by our model, we notice a group of 23 channels related to *Sabmyk*. This is a conspiracy theory that proposes itself as a better alternative to QAnon and promotes a singular quasi-religion narrative centered around a messianic figure known as Sabmyk [7]. According to the *"HOPE not hate"* organization, the Sabmyk network has over a million members distributed on about one hundred Telegram channels [11]. In particular, the mastermind of this operation is a German artist, Sebastian Bieniek, who has previously used social media to publicize his work. Intrigued by the considerable number of members achieved by this conspiracy theory, we dig into the TGDataset to investigate more about Sabmyk and its network of channels.

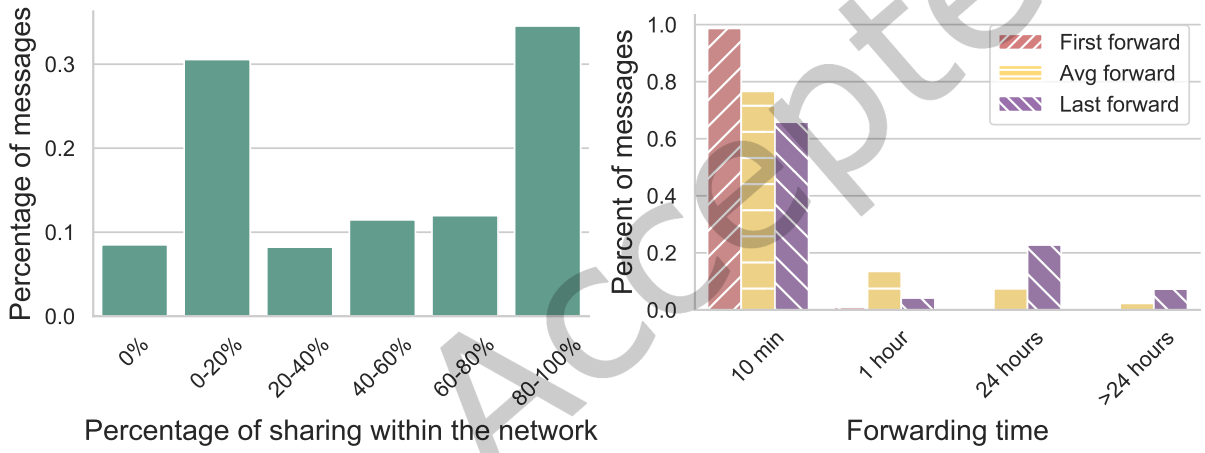
To discover the other channels of the network, we leverage the graph we built in Sec. 5.3 and a community detection algorithm. A community in a graph is a subset of nodes that are densely connected to each other and weakly connected to nodes in other communities. To uncover the Sabmyk community, we used the Leiden algorithm [55]. In this way, we discover a community of 236 channels containing the 23 channels we already know. By manually investigating the channels of this community, we can confirm that all of them are related to Sabmyk. Moreover, as we will see in the following, there is clear evidence that all of them are involved in spreading Sabmyk's theory.

Looking at the creation date of these channels, we find that the first channel of the network was created in April 2020, while the following two channels were created in December 2020. However, it is only in 2021 that most of them appeared on Telegram (76 in January and 55 in February). After that, the network expanded steadily with fewer channels until February 2022, when the last channel was created.

By analyzing the graph of the Sabmyk network, we find that it consists of 2 strongly connected components. One is of a single node, the channel entitled *Sabmyk*, and the other component contains the remaining channels. Interestingly, the Sabmyk channel is the only one in the network that never forwards a message, whereas the whole network forwards all messages posted by the Sabmyk channel. Therefore, all channels in the Sabmyk network are at one hop from the Sabmyk channel. Thus, it could be quite easy for users who joined one of the network channels to end up in the Sabmyk channel. Conversely, the users who joined the Sabmyk channel directly could remain unaware of the rest of the network. The whole network contains 1,279,424 messages. However, analyzing these messages, we find that the number of distinct messages is only 134,196 (10.48%). Indeed, most of the messages are forwarded multiple times within the network. The most shared messages are an image related to the "Great Awakening Channel" posted 14,658 times (1.14% of total messages), the invitation link to join the channel of "John F. Kennedy Jr." posted 1,989 times, and the invitation link to Antigates channel, posted 1,500 times.

Fig. 8(a) shows the percentage of the network reached by each message. About 30% of messages are shared between 20% and 80% of the network, while almost 34% of messages by nearly the whole network. Of particular interest are the messages that have never been forwarded (0% in the figure), accounting for about 8%. They are all messages belonging to channels that, in their early life, act as clone channels of VIPs. Then, they woke up and started to forward and share content related to Sabmyk. The remaining 30% of the messages forwarded by less than 20% of the channels are not written in English. Indeed, we notice that in the network, there are some channels targeting specific languages (*e.g.,* German, French). Administrators share messages in these channels only in English or the target language. As a further insight, we analyze the delay in forwarding messages from the time of content creation. As shown in Fig. 8(b), the first forward of a new message happens in the 98.6% of the cases within 10 minutes. It is likely because the content creator also manages other channels and instantaneously forwards the messages to them. The time that the whole network forwards a new message is incredibly fast: 65.8% of messages cover the network in just 10 minutes, and more than 90% in the first 24 hours. Since the messages do not cover the whole network simultaneously, we believe that the forwarding is not managed by software or a single person but by many highly coordinated people.

Sabmyk extensively used the strategy of creating fake and clone channels to reach a broad audience, attract numerous subscribers, and maximize the dissemination of its messages. As seen in previous sections, channels of services or public figures attract numerous subscribers, and it is challenging to distinguish an official channel from a fake one. Tab. 6 in the Appendix reports the name and the ID of the channels belonging to the Sabmyk network. Sabmyk exploited this idea by creating fake channels of famous people (16.09% of the network), institutional entities (*e.g.,* Department Of Defence, US Navy Channel, US Marines Channel), or news (14.55%,*e.g.,* Liverpool Times, London Post, Chicago Reporter). Finally, another technique used to attract members was to create channels that target specific kinds of users near the Sabmyk theory. This category includes channels related to QAnon (12.26%), far-right (4.21%), or other conspiracy theories (*e.g.,* Obama Gate Truth, Chemtrails News). In addition, there are 14 (5.36%) channels related to cryptocurrencies (*e.g.,* Bitcoin Inventors, StablecoinNews, Coinbase Report). These approaches were successful for the growth of the network. Indeed, in a few months, these channels went from zero subscribers to an average of more than 4,362.78, with the biggest channel *Great Awakening Channel* with 119,103.



(a) Copied messages

(b) Message forwarding time

Fig. 8. Fig. 8(b) Forwarding time of first, average, and last forwards of messages. 8(a) Percentage of message sharing within the Sabmyk network.

8 Discussion

Telegram marks. We find that 55.55% of the official Telegram channels have verified status. Even if this number could appear reasonable, we believe it is an overestimation of the number of official channels that actually acquired the verified status. Indeed, by design, our methodology relies on the verified mark to infer the status of the channels, potentially introducing a bias on this measure. Also scam/fake marks are clearly underused on the platform. The entire TGDatset contains only 184 channels marked as scam/fake, which are slightly more of the channels we are able to confirm as fake on a subset of the English channel.

Semantic features. During our investigation, we evaluated leveraging the semantic content of the messages shared by channels to improve the performance of our detector. Indeed, we find several recurrent topics among fake channels, such as anti-vaccine campaigns, controversial political ideologies, and conspiracy theories. However,

verified channels can also discuss and endorse these topics, as we observed in our dataset. Thus, we chose not to leverage this type of feature to avoid introducing any kind of bias in our classifier.

Practical relevance. Fake channels are widespread on Telegram, and at the same time, the use of verified and scam/fake marks is not very common. Although it could appear easy for a human to distinguish a fake from an official channel, it is not. Indeed, while searching for a channel, Telegram dynamically shows the matching results as the user inputs characters, providing a partial list of 3 channels. Thus, even the more careful user could experience difficulties comparing several channels and joining the right one. In particular, we believe our work can help the Telegram platform and Telegram analytic services (*e.g.,* Tgstat, Telemetr.io) flag the detected fake channels as suspicious. This action will help raise users' awareness of the potential threat and thus encourage them to consider the channel's information and the promoted products more conscientiously.

Adopting and evolving the proposed solution for the services described above is not computationally expensive. Indeed, our model and features are designed for easy scalability and efficient training. We conducted our experiments using a desktop computer equipped with an Intel Core i7-12700 CPU running at 2.10GHz, 32 GB of RAM, and an NVIDIA RTX3600 with 12 GB of RAM, running ManjaroLinux 22.1.0. Training our model detector typically takes about five minutes. Our features are also highly scalable. Most of them can be obtained effortlessly with a single query to the Telegram API or require minimal computational resources to compute. However, features related to the temporal aspect may require more effort, as they involve retrieving the entire history of channels. Nonetheless, this can be accomplished with a limited number of requests to the Telegram APIs. If there is a need to achieve higher accuracy, it is possible to add PageRank as a feature (as we see in Sec. 5.3). Unlike other features, PageRank requires knowledge of the network connection graph in addition to channel information. While retrieving this graph may initially require significant effort, leveraging pre-built resources like TGDataset can substantially reduce the time required. Additionally, these resources can be extended as needed to classify channels not covered by the existing dataset.

In this work, we focused our investigation to English fake channels, however, the model can be improved by training it on fake channels in other languages. Since our model uses stylistic, temporal, and behavioral features rather than linguistic or semantic, we believe the same set of features is effective across languages. Finally, although we expect that the core characteristics defining fake channels will remain consistent, it is important to acknowledge that the strategies employed by such channels may evolve over time. Thus, it is important to continuously update the model with the new fake channels detected attempting to capture the novel tactics utilized by fake channels.

Finally, we publicly released the TGDataset, to the best of our knowledge, the largest dataset of Telegram's channels. Indeed, it could help to understand the Telegram ecosystem further by investigating the borderline activities and the conspiracy theories on the platform.

9 Limitations

In this work, we focus only on the English channels that contain some keywords that can be used to deceive the users (*e.g.,* verified, real, official). Thus, our results reflect a specific target community.

Moreover, as we said in Sec. 4.2, to detect fake channels we do not leverage semantic features because we notice that they do not improve the accuracy of our detector. However, given the advancements in Large Language Models, we are optimistic about the potential to incorporate semantic features, which could lead to even better results. This enhancement could allow the automatic understanding of the objectives of fake channels, a task that, in our current study, we achieve through extensive manual investigation.

Lastly, although our classifier performs well in distinguishing fakes from official channels, it can be tricked by channels that behave like an official or perfectly mimic (clone) an official channel. Behaving like an official channel requires running the fake channel for a longer time with respect to the regular lifetime of fake channels,

producing content, and avoiding forwarding or using many mentions. However, all these precautions require a constant effort by the administrator and limit their goals, such as advertising other channels or promoting products.

10 Ethical Considerations

In this work, we analyzed 120,979 channels on Telegram for a total number of more than 240 million messages. During the data collection, we put particular effort into collecting only data belonging to Telegram's channels.

First, we prioritized the protection of user privacy. This involved deliberately excluding any personally identifiable information (PII) from our dataset. Specifically, only admins can write content on Telegram channels, and the platform does not disclose the usernames of admins. During our study we never attempt to deanonymize the identity of the administrators nor to link them to external platforms identities. Even if channel subscribers can not write on the channels and Telegram does not provide any information about the subscribers, we are aware that administrators in their messages can leak information about them or someone else's identity. To mitigate this potential privacy risk, we anonymized any references to usernames denoted by the format "@username" within the collected messages. This anonymization process consisted of replacing usernames with the generic placeholder "#USER". By anonymizing user identities in this manner, we aimed to protect user privacy while preserving the data's integrity for analysis. Additionally, during our data retrieval we have committed to comply with the Telegram API Terms of Service [16]. This included respecting the platform's policies regarding data collection and usage. Consequently, according to our IRB's policy, we did not need any explicit authorization to perform our experiments. Finally, we reported the fake and clone channels we detected to Telegram to prevent other users from falling prey to those identified in our research, hoping the platform would flag these channels as scams.

11 Conclusions and Future works

Telegram is becoming more popular every day, both as a classic instant messaging app and as a platform to deliver live updates and content to a large audience. Thus, it becomes increasingly important to understand what happens on the platform and how it will evolve in the future. In this paper, we faced the problem of fake and clone channels on Telegram. We characterize these kinds of channels and analyze how admins of these channels take advantage of them. We propose a machine learning model that achieves an F1-score of 85% in detecting fake accounts. Running our detector on a subset of TGDataset, we found 258 allegedly fake accounts, of which we could confirm 88. Given the extent of the phenomenon, the high number of subscribers, and the difficulty of distinguishing fake channels from official ones, the need for institutions, famous people, and organizations to obtain verified status for their channels is on the rise. Indeed, we notice only a few official channels leverage this opportunity.

With this work, we shed light on one of the several controversial activities that run on the Telegram platform. However, we believe further investigations are needed to illuminate the Telegram ecosystem completely. Indeed, in our research, we noticed a heavy presence of channel networks that attempt to spread conspiracy theories by exploiting fake and clone channels. Thus, it is interesting to understand how these networks are organized, how they evolve over time, and which is their target audience. Moreover, we believe Telegram public groups are a vast portion of Telegram and deserve further exploration. Indeed, here it is possible to easily access the complete list of subscribers, compromising the users' privacy and impersonating the administrators to carry out frauds.

Acknowledgments

This work has been partially funded by projects: MUR National Recovery and Resilience Plan, SERICS (PE00000014); and ST3P (B83C24003210001) under the "Young Researchers 2024-SoE" Program funded by the Italian Ministry of University and Research (MUR).

References

[1] 2017. Telegram to block terror channels after Indonesian ban. https://www.bbc.com/news/business-40627739.
[2] 2018. Another Phishing scam 'Kraken Official Telegram Channel'. https://steemit.com/cryptocurrency/@techstack/another-phishing-scam-kraken-official-telegram-channel.
[3] 2019. Anatomy of a telegram scam. https://blog.coinbase.com/anatomy-of-a-telegram-scam-9fd3dfb8c310.
[4] 2019. Telegram the latest safe haven for white supremacists. https://www.adl.org/blog/telegram-the-latest-safe-haven-for-white-supremacists.
[5] 2019. Telethon's Documentation. https://docs.telethon.dev/en/stable/.
[6] 2021. NLTK RegexpTokenizer. https://www.nltk.org/_modules/nltk/tokenize/regexp.html.
[7] 2021. Sabmyk Network: Founder of bizarre new religion targeting QAnon believers 'unmasked' by Hope Not Hate. https://www.independent.co.uk/news/world/europe/sabmyk-network-qanon-conspiracy-theories-b1820639.html.
[8] 2021. Telegram Analytics. https://tgstat.com/.
[9] 2021. Telegram, the powerful COVID-19 choice of communications by many governments. https://www.channelnewsasia.com/commentary/coronavirus-covid-19-government-telegram-whatsapp-fake-news-info-936061.
[10] 2021. TelemeterIo. https://telemetr.io/en/channels.
[11] 2021. Unmasked: the QAnon 'messiah'. https://www.hopenothate.org.uk/unmasked-the-qanon-messiah//.
[12] 2021. Why journalists and dissidents turn to Telegram. https://www.indexoncensorship.org/2021/06/telegram/.
[13] 2022. Page Verification Guidelines. https://telegram.org/verify.
[14] 2023. Scammers in telegram and how to report. https://www.telegramadviser.com/scammers-in-telegram-and-how-to-report/.
[15] 2023. TelegramClient. https://docs.telethon.dev/en/latest/modules/client.html?highlight=iter_messages#telethon.client.messages.MessageMethods.iter_messages.
[16] 2024. Telegram API Terms of Service. https://core.telegram.org/api/terms.
[17] Mansour Alsaleh, Abdulrahman Alarifi, Abdul Malik Al-Salman, Mohammed Alfayez, and Abdulmajeed Almuhaysin. 2014. TSD: Detecting Sybil Accounts in Twitter. In *2014 13th International Conference on Machine Learning and Applications*. 463–469. https://doi.org/10.1109/ICMLA.2014.81
[18] Davide Anguita, Luca Ghelardoni, Alessandro Ghio, Luca Oneto, and Sandro Ridella. 2012. The 'K'in K-fold cross validation. In *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. i6doc. com publ, 441–446.
[19] Andrea Bacciu, Massimo La Morgia, Alessandro Mei, E Nerio Nemmi, Valerio Neri, and Julinda Stefa. 2019. Bot and gender detection of Twitter accounts using distortion and LSA. Notebook for PAN at CLEF 2019. In *Working Notes Papers of the CLEF 2019 Evaluation Labs volume 2380 of CEUR Workshop*.
[20] Timothy Baldwin and Marco Lui. 2010. Language identification: The long and the short of the matter. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. 229–237.
[21] Jason Baumgartner, Savvas Zannettou, Megan Squire, and Jeremy Blackburn. 2020. The Pushshift Telegram Dataset. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 840–847.
[22] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. 2009. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*. 551–560.
[23] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
[24] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30, 1-7 (1998), 107–117.
[25] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the detection of fake accounts in large scale social online services. In *9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*. 197–210.
[26] Zhenfeng Cao, Minzhang Zheng, Yulia Vorobyeva, Chaoming Song, and Neil Johnson. 2017. Dynamical patterns in individual trajectories toward extremism. *Available at SSRN 2979345* (2017).
[27] Loredana Caruccio, Domenico Desiato, and Giuseppe Polese. 2018. Fake account identification in social networks. In *2018 IEEE international conference on big data (big data)*. IEEE, 5078–5085.
[28] Duan-Bing Chen, Hui Gao, Linyuan Lü, and Tao Zhou. 2013. Identifying influential nodes in large-scale directed networks: the role of clustering. *PloS one* 8, 10 (2013), e77455.

[29] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2015. Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems* 80 (2015), 56–71.
[30] Arash Dargahi Nobari, Negar Reshadatmand, and Mahmood Neshati. 2017. Analysis of Telegram, an instant messaging service. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2035–2038.
[31] Buket Erşahin, Özlem Aktaş, Deniz Kılınç, and Ceyhun Akyol. 2017. Twitter fake account detection. In *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 388–392.
[32] Matt W Gardner and SR Dorling. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment* 32, 14-15 (1998), 2627–2636.
[33] Aditi Gupta and Rishabh Kaushal. 2017. Towards detecting fake user accounts in facebook. In *2017 ISEA Asia Security and Privacy (ISEASP)*. IEEE, 1–6.
[34] Kazuyuki Hara, Daisuke Saito, and Hayaru Shouno. 2015. Analysis of function of rectified linear unit used in deep learning. In *2015 international joint conference on neural networks (IJCNN)*. IEEE, 1–8.
[35] Ali Hashemi and Mohammad Ali Zare Chahooki. 2019. Telegram group quality measurement by user behavior analysis. *Social Network Analysis and Mining* 9, 1 (2019), 1–12.
[36] Mohamad Hoseini, Philipe Melo, Fabricio Benevenuto, Anja Feldmann, and Savvas Zannettou. 2021. On the Globalization of the QAnon Conspiracy Theory Through Telegram. *arXiv preprint arXiv:2105.13020* (2021).
[37] Vincenzo Imperati, Massimo La Morgia, Alessandro Mei, Alberto Maria Mongardini, and Francesco Sassi. 2023. The Conspiracy Money Machine: Uncovering Telegram's Conspiracy Channels and their Profit Model. *arXiv preprint arXiv:2310.15977* (2023).
[38] David Kempe, Jon Kleinberg, and Éva Tardos. 2005. Influential nodes in a diffusion model for social networks. In *International Colloquium on Automata, Languages, and Programming*. Springer, 1127–1138.
[39] Mike Kestemont, Efstathios Stamatatos, Enrique Manjavacas, Walter Daelemans, Martin Potthast, and Benno Stein. 2019. Overview of the Cross-domain Authorship Attribution Task at PAN 2019. In *CLEF 2019 Labs and Workshops, Notebook Papers*, Linda Cappellato, Nicola Ferro, David E. Losada, and Henning Müller (Eds.). CEUR-WS.org.
[40] Massimo La Morgia, Alesssandro Mei, and Alberto Maria Mongardini. 2023. TGDataset. https://github.com/SystemsLab-Sapienza/TGDataset
[41] Massimo La Morgia, Alessandro Mei, and Alberto Maria Mongardini. 2023. Tgdataset: a collection of over one hundred thousand telegram channels. *arXiv preprint arXiv:2303.05345* (2023).
[42] Massimo La Morgia, Alessandro Mei, Alberto Maria Mongardini, and Jie Wu. 2023. It'sa Trap! detection and analysis of fake channels on telegram. In *2023 IEEE International Conference on Web Services (ICWS)*. IEEE, 97–104.
[43] Massimo La Morgia, Alessandro Mei, Francesco Sassi, and Julinda Stefa. 2020. Pump and Dumps in the Bitcoin Era: Real Time Detection of Cryptocurrency Market Manipulations. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–9.
[44] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*. 4768–4777.
[45] Shie Mannor, Dori Peleg, and Reuven Rubinstein. 2005. The cross entropy method for classification. In *Proceedings of the 22nd international conference on Machine learning*. 561–568.
[46] Monero. 2023. What is Monero (XMR)? https://www.getmonero.org/get-started/what-is-monero/
[47] Massimo La Morgia, Alessandro Mei, Francesco Sassi, and Julinda Stefa. 2022. The Doge of Wall Street: Analysis and Detection of Pump and Dump Cryptocurrency Manipulations. *ACM Trans. Internet Technol.* (2022). https://doi.org/10.1145/3561300 Just Accepted.
[48] Adam Paszke et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035.
[49] F. Pedregosa et al. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[50] Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. 2000. New support vector algorithms. *Neural computation* 12, 5 (2000), 1207–1245.
[51] Rohit Shewale. 2023. 80+ Telegram Statistics In 2023 (Demographics & Financials). https://www.demandsage.com/telegram-statistics/.
[52] Nakatani Shuyo. 2010. Language Detection Library for Java. http://code.google.com/p/language-detection/
[53] P Sibun and JC Reynar. [n. d.]. Language determination: Examining the issues. In *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval*. 125–135.
[54] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. 2013. {Trafficking} Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse. In *22nd USENIX Security Symposium (USENIX Security 13)*. 195–210.
[55] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* 9, 1 (2019), 1–12.
[56] Janith Weerasinghe, Bailey Flanigan, Aviel Stein, Damon McCoy, and Rachel Greenstadt. 2020. The pod people: Understanding manipulation of social media popularity via reciprocity abuse. In *Proceedings of The Web Conference 2020*. 1874–1884.
[57] Mike Wendling. 2021. QAnon: What is it and where did it come from? https://www.bbc.com/news/53498434.

[58] Cao Xiao, David Mandell Freeman, and Theodore Hwa. 2015. Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*. 91–101.
[59] Jiahua Xu and Benjamin Livshits. 2019. The anatomy of a cryptocurrency pump-and-dump scheme. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 1609–1625.
[60] Ahmet S Yayla and Anne Speckhard. 2017. Telegram: The mighty application that ISIS loves. *International Center for the Study of Violent Extremism* (2017).
[61] Zijun Zhang. 2018. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–2.

A The Sabmyk network

Received 30 December 2023; revised 17 May 2024; accepted 24 October 2024

Table 6. Channel ID and username of the Sabmyk channels. Prepending the string *https://t.me/* to the username is possible to obtain the URL of the channel (https://t.me/username).

ch_ID	username	ch_ID	username	ch_ID	username
1373606065	AmeliAchaemenes	1230442614	GreatAwakeningOfficial	1223946758	QAnonItaliano
1177244321	AmericanTribune	1218448464	GreatAwakeningUK	1514752514	QAnonMessiah
1382087559	AncientFeed	1454816244	GreatAwakeningUS	1237833060	QAnonPeople
1426840488	AntiCoronaRegime	1407623625	GreatCoronaCoup	1195901626	QAnonStormBase
1419302828	AntiCoronaTerrorism	1197879088	GreaterMAGA	1300222229	QDonaldJTrump
1359669362	AntiFakePandemic	1223585300	GreatMarch	1149544865	QdropsFeed
1188237003	AntiGates	1323194217	GreatUnmatrix	1448102539	QShaman
1214316989	AntiilluminatiOfficial	1253794545	GregorGysi	1393121192	qspeaking
1331286870	antiNewsweek	1241336568	GrosseAufwachen	1529258280	QuantityNews
1447259941	Antipafi	1201562722	GrossesAufwachen	1135024441	Querdenker24
1517396641	AntivaxMessiah	1571701728	HamburgPresse	1191673380	quotationnation
1315033511	ArizonaReporter	1352200753	Hardhauer	1258313992	RealJoshHawley
1396140109	artisallaround	1412593838	HerbertKickl	1270127441	RealMikePompeo
1421276621	AtlantisOfficial	1228780906	HereIsQ	1186678754	RealRonDeSantis
1271371846	atmumra	1311957508	HereJoeM	1207633990	RealSteveBannon
1259842157	AtmumraDeutsch	1234393139	HistoryFeed	1457158321	RepublicanToday
1278353388	AustraliaTimes	1321695674	HopeNotFear	1390498282	RGiuliani
1233417816	AwakenMovement	1496318488	HuanOsa	1369199894	RisveglioItaliano
1432356346	AwakenWeAre	1757974647	inchnews	1366803185	RonWatkinss
1409730795	BayernPresse	1466180915	IndiaTVN	1460620427	RowanAtkin
1315681248	BBCpost	1286224154	IranAT	1381999699	RussiaRA
1385495873	BerlinerNachrichten	1496771986	IrlandDaily	1452287396	sabmyk
1428580796	BestTokenNews	1199281120	JapanAwakening	1383773284	SabmykAwakening
1669428026	BitcoinInventors	1432820298	JCMiller	1185671778	SabmykDeutsch
1449097796	BlackWhiteUnite	1618557062	JeffBridgemaker	1344764443	Sabmykpedia
1219960269	BLMnews	1176633798	JesusAmerica	1176880888	SatanicArt
1494824103	Bravetower	1407364366	JoeBidenDaily	1579224104	SBMKcoin
1363596352	BritishPatriotsParty	1426082961	JohnFKennedyJr	1768264210	SBMKme
1582382405	BrunoPuno	1224553147	JonVoightReal	1375609806	ScotlandFirst
1305236052	CanadaFreeNews	1115756426	JoschkaFischer	1284622328	shawunawaz
1428334973	CandaceOw	1570783961	JuAssange	1236952515	ShawunawazDeutsch
1442585851	CapitolNews	1296893170	KanyeOW	1221625398	Shawunuwaz
1795851747	CardanoReport	1380284892	KeanuReevesReal	1703265721	SHIBAinuInfo
1236491691	CharlesFlynn	1203451831	KoelnInfo	1346243157	SideyPowellAcount
1453914542	ChemtrailsNews	1337183608	KoreaAwaken	1173114609	SpaceForceNews
1488443509	ChicagoReporter	1491342628	LibertyOnlineNews	1703437243	StablecoinNews
1340956717	ClintEastwoodReal	1335746275	LiverpoolTimes	1171930401	StarseedChildren
1609605643	CoinbaseReport	1362828115	LLWoodChannel	1151473843	supernarrativ
1750848305	CryptoartMuseum	1370932092	LondonPost	1262467306	SylvesterSt
1544644340	CryptoPunksClub	1230303566	LosAngelesPost	1155637672	TheAmericanProphet
1745881983	CTBCh	1504784360	LoveNotVaccines	1422512576	TheAntiguardian
1780223986	DAOautonomous	1788784877	MakePeopleFreeAgain	1303530488	TheBritishNews
1448683167	DasGrosseAufwachen	1495856197	MassimoGaravaglia	1755470272	TheCryptoMeme
1327675983	DasGrosseE	1165294845	MaurizioCattelan	1749103723	TheEconomyNews

Continued on next page

ch_ID	username	ch_ID	username	ch_ID	username
1231373396	DemocratsNews	1302834936	MelGibsonReal	1466056636	TheEuropeNews
1232401093	DepartmentOfDefence	1260233653	memepow	1447582094	TheGhostEzra
1335404040	DianaFSpencer	1723058370	MetaverseAccount	1358923616	TheGreatAwakening2
1243046900	DigitalExodus	1492749012	MichaelJacksonAlive	1467950709	TheGreatAwakenings
1477204083	DigitalSupersoldier	1267146193	MichaelWendlerNews	1484273466	TheGreatMarch
1486325736	DonaldTrumpJrInfo	1154227479	MikeJLindell	1206325310	TheGreatUnmatrix
1321909590	DrainTheSwampNews	1181587153	MikePenceInfo	1647143572	TheMedicalFreedom
1467553986	DTrumpt	1217936887	MuetterUndVaeter	1358714361	TheNewsweek
1579161417	EdwardSnwdn	1385560963	NesaraGesaraInfo	1340214082	TheRealAwakening
1380332721	ElectricNewsChannel	1553561389	NFTcollectorGroup	1701555011	TheRevolutionNOW
1532914074	ElonMuskInfo	1748569733	NFTfair	1378594542	TheSummerOfResistance
1457271656	EnglandFirst	1167638684	NFTreport	1491943863	TheTrumpists
1519092394	EricClaptons	1498660836	NibiruInfo	1323344228	TillSchw
1724583330	EthereumDaily	1160572206	NicolaTeslaNews	1283532745	TimesOfChina
1154689322	EvaHermanNachrichten	1225570590	NoahProphezeiung	1440801049	TomBradyReal
1776706282	EvaVlaardingerbroek	1224814251	NoahsProphecy	1337470749	TOSullivan
1208971990	FirstFlushNews	1486609740	NoahsProphezeiung	1403866025	TuckerCarlsonNews
1184705910	FloridaDaily	1264767418	NostradamusInfo	1421434447	UfoOfficial
1460863131	FrancescoDeGregori	1632849858	NovakDjoko	1194083615	UniteNotDivide
1439127441	FranceToday	1467066582	ObamaGateTruth	1365542123	UnsplitSoul
1223443993	FriedlichZusammen	1476087987	OfficialAnonymous	1410791343	UnsplitSouls
1773218676	FuckNaturalimmunityDenier	1323653137	OfficialSatoshi	1562258004	UnvaccinatedWelcome
1367961220	GaiasKinder	1266753937	OfficialTimes	1466731211	USmarinesChannel
1179964408	Genapostle	1315856681	OhioDaily	1154131355	USmilitaryVoice
1421108458	GeneralFlynnInfo	1325372115	OklahomaNews	1264167396	USnavyChannel
1221008617	GeorgiaTribune	1366146677	OskarLafontaine	1154525863	USPatriots
1343537775	GerhardSchroder	1304953034	otevremecesko	1568221393	VaxFr
1216131889	GodWinsOfficial	1238694982	PatriotPartyUS	1429538933	VvanV
1209424557	GoldTradeNews	1204135756	PatriotsRepublicNews	1433940064	vyacheslavvelichko
1151585245	Govapostle	1393635062	People4Freedom	1444644332	WeAreTheFaithful
1646760828	GovRon	1408766118	PierreTati	1299461938	WikiOfficial
1493291081	GreatAwakeningChannel	1581960001	PolitischeKryptokunst	1156118630	WolfgangThierse
1422343221	GreatAwakeningDe	1166088338	QAnonCentral	1284034051	WorldAwaken
1190461302	GreatAwakeningFrance	1217710953	QAnonDEU	1486985550	WWG1WGAhere
1361740061	GreatAwakeningItalia			1298735255	YellowstoneWolf