Jettisoning Junk Messaging in the Era of End-to-End Encryption: A Case Study of WhatsApp

Pushkal Agarwal
pushkal.agarwal@kcl.ac.uk
King's College London
United Kingdom

Aravindh Raman
aravindh.raman@telefonica.com
Telefonica Research
Spain

Damiola Ibosiola
d.i.ibosiola@qmul.ac.uk
Queen Mary University of London
United Kingdom

Nishanth Sastry
n.sastry@surrey.ac.uk
University of Surrey
United Kingdom

Gareth Tyson
gtyson@ust.hk
Hong Kong University of Science & Technology

Kiran Garimella
kiran.garimella@rutgers.edu
Rutgers University
United States of America

ABSTRACT

WhatsApp is a popular messaging app used by over a billion users around the globe. Due to this popularity, understanding misbehavior on WhatsApp is an important issue. The sending of unwanted junk messages by unknown contacts via WhatsApp remains understudied by researchers, in part because of the end-to-end encryption offered by the platform. We address this gap by studying junk messaging on a multilingual dataset of 2.6M messages sent to 5K public WhatsApp groups in India. We characterise both junk content and senders. We find that nearly 1 in 10 messages is unwanted content sent by junk senders, and a number of unique strategies are employed to reflect challenges faced on WhatsApp, *e.g.,* the need to change phone numbers regularly. We finally experiment with on-device classification to automate the detection of junk, whilst respecting end-to-end encryption.

CCS CONCEPTS

• **Social and professional topics → Censorship**; **User characteristics**; • **Computing methodologies → Machine learning**.

KEYWORDS

WhatsApp, end-to-end encryption, junk messaging, spam

ACM Reference Format:
Pushkal Agarwal, Aravindh Raman, Damiola Ibosiola, Nishanth Sastry, Gareth Tyson, and Kiran Garimella. 2022. Jettisoning Junk Messaging in the Era of End-to-End Encryption: A Case Study of WhatsApp. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512130

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512130

1 INTRODUCTION

WhatsApp is the most popular messaging app in the world, with over 1.5 billion active users each day. With this massive popularity, *unwanted messages* (such as spam) have become a challenge for WhatsApp. However, unlike existing messaging systems (*e.g.,* email, Twitter), where platforms can read the content, WhatsApp follows an end-to-end encryption model where the content is not accessible. Whereas this offers stronger guarantees on privacy, it makes content moderation difficult.

Although WhatsApp has made progress [25] in detecting users who send unsolicited messages to individuals, there is no solution for users who send such messages to *public WhatsApp groups* [16]. These are groups where the admins publicly share a link (*e.g.,* via a website or social platform) to join the group. WhatsApp provides strong protection from being contacted by strangers, allowing users to easily block unsolicited messages from those not in their contact list. In contrast, as long as a user is a member of a public WhatsApp group, they cannot avoid messages that strangers may send, regardless of whether the messages are germane to the group or not. Thus, these strangers can abuse public WhatsApp groups by sending messages that are irrelevant to the purpose of a group, *e.g.,* links to adult content or sexual services, phony job offers *etc.*

Understanding the nature of unwanted junk messages in public groups is therefore vital for improving users experience on WhatsApp and other messaging platforms. For instance, receiving excessive notifications can be highly disruptive and undermine the responsiveness to legitimate notifications [31, 37]. In this paper, we use the term "junk messages" to capture a range of messages that are unlikely to be of interest to the general group membership. This may be because the content is unsavoury (*e.g.,* links to adult content in political groups) or simply irrelevant (*e.g.,* asking people to fill out surveys). We therefore define junk messages as: "*those which are not considered of interest or suitable by admins for a group, leading such posters to be removed*".

To understand the nature of such misbehaviour we focus on India, a country where over 400 of the 460 million people online are on WhatsApp.[1] Given the population we are studying (mostly first time Internet users from India), the susceptibility of believing the content in junk message (*e.g.,* fake job offers) is high. As a common topic that attracts interest from across the population (all languages

[1] https://www.cjr.org/tow_center/india-whatsapp-analysis-election-security.php

and states), we focus on *national politics*, and gather 2.6 million messages from 5,051 public political WhatsApp groups in India (§2) which are posted before, during and after the elections in 2019.

In this paper, we analyse the characteristics of junk messages (§3) as well as the actions of junk senders (§4). While we do find parallels with junk posted in other platforms, we see distinct trends which are unique to WhatsApp. For example, as with platforms like Facebook [39], WhatsApp junk message consists of topics such as job advertisements and click bait often embedded within URLs. We also observe unique patterns to WhatsApp, *e.g.,* junk senders using multiple phone numbers to spread the same messages over a small number of days (note that phone numbers are typically more difficult to obtain than email addresses in India). These unusual traits lead us to explore junk senders' evasion tactics. We see many joining and leaving groups multiple times, thereby avoiding being removed by admins, and nearly a quarter of all phone number changes are performed by junk senders. Finally, we investigate ways to support admins (§5) by devising end-to-end encryption compliant classifiers to identify junk messages. We obtain good performance, with accuracy scores exceeding 0.85. This offers a foundation for future moderation efforts in the era of end-to-end encrypted messaging.

2 DATASET & METHODOLOGY

2.1 Data collection methodology

We take inspiration from prior work, which found that public groups that discuss politics are widely used in India [29, 43] and Brazil [33, 40]. Our study is based on data from *public* WhatsApp groups discussing politics in India. We select this focused subset of groups, as it allows us to more easily identify out-of-scope messages (*i.e.,* junk), as these pertain to topics outside of politics. To this end, we searched for WhatsApp group links (`chat.whatsapp.com`) on Facebook, Twitter and Google during November 2018 using an extensive set of 349 manually curated keywords[2] from multiple Indian languages (including English) relating to politics in multiple languages. This yielded 5,051 groups. These are typically created by political parties or party supporters in order to reach an audience which is only available via WhatsApp. Hence, most of these groups have a well defined organizational structure [5]. Note, due to end-to-end encryption, WhatsApp is limited in its ability to moderate content in these groups. Instead, moderation is mostly left to group admins, who also have powers to remove users.

With this list of 5,051 groups, we use the toolkit from [16] to collect data. We collect all messages between October 2018 and August 2019. Across the 5,051 groups, we collect 2.6 million messages posted by over 172K unique users over a period of 302 days. We also record 437K action events, covering actions taken by users including entering or leaving groups and changing phone numbers. Table 1 summarizes the actions performed by users within a group.

Ethics note: Our data collection abides by the terms of service of WhatsApp and was considered exempt by the Institutional Review Board at MIT. All data was anonymised before analysis, and any personally identifiable information was masked. All phone numbers were one-way hashed, after extracting the country code.

Table 1: Actions captured within a group.

Action	Description	Action Counts	Unique Users
added	added by a member	61k	37k
added_by_admin	added by a group admin	73k	49k
joined_via_link	joined via an invite link	132k	54k
left	left the group	154k	73k
removed	removed from a group	9k	7.3k
number_changed	changed from one number to another	6k	1.5k

2.2 Message pre-processing

Our dataset consists of 2.6M messages out of which only 1.4M are unique. We filter and cluster these messages to have a unique and distinct set of messages.

2.2.1 Message filtering. Because the content of the message is important in identifying junk, we focus on the four top languages (Hindi, English, Telugu and Tamil) which collectively represent 74% of messages in our dataset.[3] We also remove messages containing just URLs (no accompanying text), boilerplate content such as 'hi' or 'good morning', and, messages consisting solely of emojis, which constitute around 25% of our data. We filter such content to avoid characterising low entropy posts – such messages could fall under both on-topic and off-topic as junk [18]. As later discussed, the percentage of boilerplate messages among junk posters is 7% (12% for legitimate users). Filtering these out, we are left with 766K messages from 2.6 million messages which is in turn quicker to cluster as we discuss in next section.

2.2.2 Message clustering. We qualitatively observe that many messages are close variants of each other. To group together near-similar variants of the same message, we use MinHash and Locality Sensitive Hashing (LSH) [17, 32]. We find that the best clustering performance is obtained by using 10 min-hashes in 5 bands. We made this choice by taking 100 near-identical messages and 100 distinct messages, and experimenting with a range of parameters to derive the optimal for separating out these two sets.

Overall, this results in 73K clusters, with an average of 10 (median 4) messages in each cluster. As a quality check, we randomly select 100 random clusters from each of the 4 languages and manually verify that 97% are similar (the rest are from a bot service [14]). Of the 73K clusters, 25.4K have at least 5 messages, and contain 420K messages in total. These 420K messages, which we term "frequently sent messages" (sometimes shortened to "frequent messages") are the object of study for the rest of the paper, except when studying URLs (§3.2.2), where we include all the messages containing URLs.

2.3 Junk and non-junk annotation

We next describe annotation of messages as 'junk' or 'non-junk'.

2.3.1 Manual annotation. We start by identifying a seed set of users who were manually removed from at least two groups by their admins. Recall that according to our earlier definition, messages that trigger users to be removed, are considered junk candidates. We therefore conjecture that users who are manually removed by different group administrators more than twice (257 users) may

[2] List available: http://tiny.cc/curated-keywords

[3] We tag messages with their language using Google's Language Detector [34] in R.

provide a useful set of candidate junk messages. We later confirm that this is true: over 90% of messages sent by these users are later annotated as junk and are often accompanied by comments by admins such as "pls don't spam in this grp with such posts".

We first extract the 68K messages (grouped into 1,004 clusters) sent by these 257 users and manually annotate each cluster as 'junk' or 'non-junk'. We perform the annotations in two stages. First, two English-speaking 'lead' annotators (domain experts) work independently to label the 220 English clusters contained within the 1,004 clusters. Since a large portion of junk content is self-evident, the inter-rater agreement (Cohen's-Kappa) is high (0.96). The remaining disagreements were discussed to reach full agreement. Based on the labeling of 220 clusters, two lead annotators prepare guidelines for further annotation on larger datasets by multiple members of the annotation team.

Following this initial step, which validated the ability to identify junk messages with high agreement levels, we progress to the second stage. We recruit 2 annotators for English, 3 for Hindi and 1 each for Telugu and Tamil. We ask them to further categorise the junk messages into the following categories: (*i*) Promotion messages (if non-political promotion as these groups are for political discussions). (*ii*) Adult content related messages. (*iii*) Invitations to register for external services via links, often for money. (*iv*) Offers to earn money, win prizes etc. Or (*v*) Anything which looks 'suspicious' and not relevant to the group at large. In cases where annotators find something suspicious but hard to classify, they were allowed to consult the lead annotators who prepared the guidelines. With this guidance, the set of 784 clusters were given to annotators (native speakers of Hindi, Telugu and Tamil). Each annotator received the subset of messages in their native language, and were given identical guidelines (as described above). In total, the above two steps result in 63K messages (from 663 clusters) being tagged as junk and 5K messages (341 clusters) being tagged as non-junk.

2.3.2 Semi-Automatic Annotation. The above produces a "gold standard" set of annotated messages. We next complement this with a semi-automatic annotation process to broaden our analysis. Here, we construct a dictionary of words that are used at least 5 times in the 63K messages classified as junk above. We then manually clean the list to obtain a set of 324 high precision junk words.[4] We then search for the occurrence of these junk words in the entire database of 420K frequently sent messages in Hindi, English, Telugu and Tamil. In each cluster where we find one or more of the junk words, the annotators examine the cluster and validate it as 'junk' or 'non-junk' using the same approach as above. Finally, we obtain a labelled dataset containing 295K junk (from 3.5K clusters) and 112K non-junk (from 3.2K clusters) messages as summarised in Table 2.

Table 2: Summary of the annotations set.

Message type	From	Unique messages	Total messages
Junk	Removed users	663	63K
Non-Junk	Removed users	341	5K
Junk	Semi-automation	2.8K	232K
Non-Junk	Semi-automation	2.9K	107K

2.3.3 Validation. We note that the above annotations are inherently subjective. Hence, to give greater confidence that our concept of junk and non-junk is reasonable, we finally create an independent panel of 11 assessors (distinct from the annotators) who look at a randomly chosen set of 100 messages (split into 50 junk and 50 non-junk messages). All assessors were Indian, and collectively represent 7 states with different age groups and genders. The panel were only told that these messages are from public WhatsApp groups related to Indian politics and were asked to mark whether they think such messages are 'junk' or 'non-junk' based on their own personal understanding. We find that there is a median agreement of 95% across the labels for all the independent assessors (labelling 100 messages without any guidelines) and our annotators (labelling large scale data with the prepared guidelines).

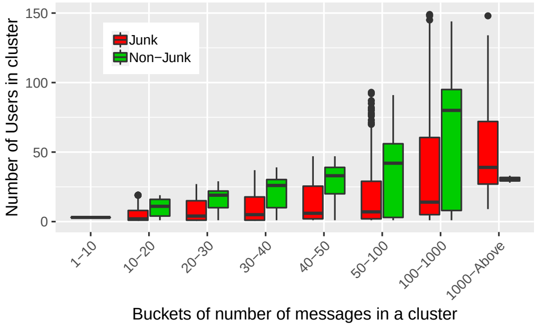


Figure 1: Numbers of users spreading a message, indexed by number of times the message or its close variants are seen.

3 CHARACTERISTICS OF JUNK CONTENT

We first look the nature and content of WhatsApp junk *messages*.

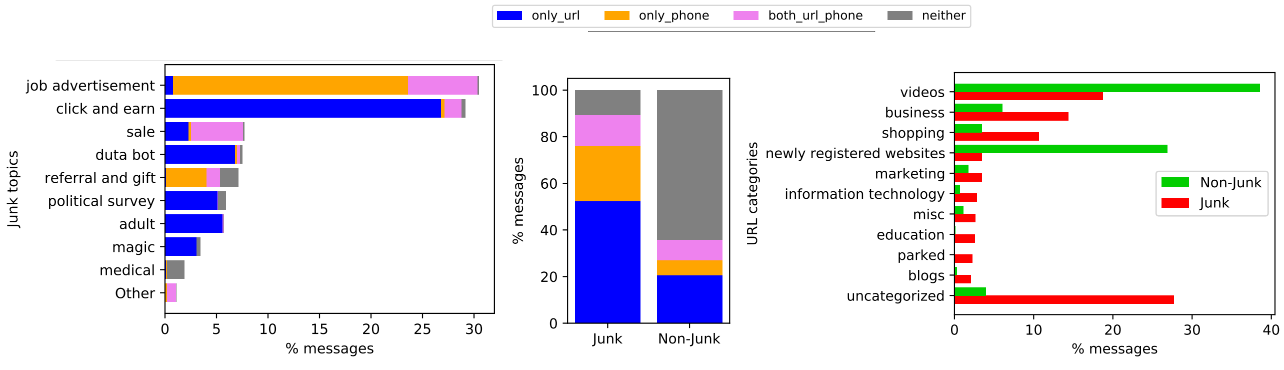
3.1 Understanding the scale of junk

We first consider two aspects of scale: the number of times a message is posted, and the number of users who send it. Each message cluster obtained (see §2.2) may contain several messages that are closely related variants. Figure 1 groups message clusters into different buckets based on the number of times those messages are found in our data, across different groups.

We find that clusters containing junk messages have larger numbers of messages on average, although the median numbers are similar (24 for junk clusters *vs.* 23 for non-junk clusters; compared to mean 83.6 in junk *vs.* 35 for non-junk). This indicates a highly skewed distribution of messages per cluster, particularly for junk clusters. For instance, there are over 37 clusters of junk with more than 1000 messages, the largest cluster having over 27K messages. In contrast, only 2 non-junk clusters have more than 1000 messages and the largest having 2,065 messages. These are videos URLs and associated text related to political speeches.

We also see that popular clusters, which contain messages that are forwarded many times, inevitably involve more users. However, Figure 1 shows that for messages in junk and non-junk clusters

[4] The threshold 5 was set by manually inspecting the results obtained by selecting various values from 3 to 10 in order to obtain a list of junk words in all four languages. Stop words were removed in all languages using multi-lingual dictionary from 'stopwords' library in R.) Some example words include free, job, iphone, samsung, IELTS, sex, click, sales and vacancy.



(a) (b) (c)

Figure 2: (a) Topics found in the top 100 clusters of Junk (48% of all Junk messages); (b) Overall fraction of URLs and phone numbers found in the content of messages; (c) Categories of URLs as identified by VirusTotal in messages.

of similar sizes (*i.e.,* in the same "bucket" size on the x-axis), the junk clusters are driven by fewer users. This indicates that junk is disseminated in a more proactive manner by a smaller set of users.

3.2 Understanding junk content

3.2.1 Junk topics. To explore the topics discussed within junk, we employ three annotators to manually examine messages in the top 250 clusters of junk messages. By performing an initial qualitative analysis, we identify 10 core topics. We then ask the annotators to categorise the messages from the top 250 clusters into these 10 topics. The top 250 clusters comprise 61% (181K messages) of the total number of junk messages (295K) in our dataset. Each message cluster is examined by at least one annotator and the category label applied was checked by a second annotator. All differences of opinion were resolved in a discussion and we finally obtained 100% inter-agreement between annotators.

Figure 2a captures the relative frequencies of the topics, as well as the frequency of URLs and phone numbers within the messages. We emphasise that this taxonomy is based on the fact that these messages are post to *political* groups. Some of these may not be considered junk if posted to other types of groups.

Job Advertisements. Comprising nearly 30% of our annotated junk, the most widespread junk is advertisements for jobs. Nearly all (97%) of job advertisements provide a contact phone number, or a phone number as well as a URL. Of course, these may be genuine job advertisements, although they are off-topic junk for political WhatsApp groups. Though we are not able to identify if these were genuine job ads, the template structure of these ads makes us believe (since such users are removed) these are junk and could involve a scam. Interestingly, over 95% of junk in Telugu language forums consist of job advertisements.

Click and Earn. These comprise 29% of junk messages, and ask users to click on a URL, promising a reward. 98% of these messages contain a URL, but no phone number.

Sales. These constitute 7.7% of junk and offer items for sale. 65% of these messages contain URLs and a phone number, and could be genuine items for sale.

Referral and gifts. These junk messages (7.5%) offer a gift in return for referrals of users to an online service subscription, and consist mostly of a URL to click.

Duta Bot. These are (benign) junk messages (7.1%) sent by a news bot service called Duta Bot [14] comprising regular news or sports updates. Note that many admins do not welcome the Duta Bot posts: they are regularly removed from groups. We find that, 26 bots who posted 1000 junk messages are removed by admins.

Adult. These (5.7%) mostly contain URLs that lead to adult websites or offer adult sex-related services.

Political Survey. These (5.6%) mostly contain junk URLs that invite users to participate in fake political surveys.

Magic. These (3.4%) messages contain text which asks user to forward a message to experience something supernatural, *e.g.,* "Forward and see magic: your phone battery will get charged to 100%".

Medical. These (1%) messages offer treatment for common and often embarrassing ailments, *e.g.,* "ayurvedic treatment for piles".

Other. Approximately 2% cannot be categorised into any of the above groups and consist of junk such as "daily event update".

3.2.2 URLs and Phone Numbers. As shown above, a significant number of junk messages contain URLs (167K messages) and phone numbers (74K messages). Figure 2b compares the fraction of junk and non-junk messages that contain phone numbers, URLs or both. We see a marked difference between junk and non-junk, with nearly 90% of junk messages containing either a phone number, a URL or both (in contrast to just 36% for non-junk). We also notice 19.4K unique phone numbers present in the *content* of the messages (note that for ethical reasons, all numbers are one-way hashed before analysis). Out of this, only 9.5% of the numbers were found to be senders of any messages, and 85% of these were junk.

We next take the 56.8% of junk and 27.5% of non-junk messages that contain URLs. To explore the nature of these URLs, we use VirusTotal[5] to classify each according to its type of activity [22, 27]. For 81.1% of domains we identify the category associated with the domain. Figure 2c shows the distribution of URL categories for both junk and non-junk. We see that video URLs are popular in junk messages, as well as newly registered sites (manual analysis reveals these are primarily news websites, *e.g.,* `upchaupal.com`). In contrast, junk messages carry far more business and shopping URLs. Worryingly, 1% of the messages carry URLs marked as 'elevated exposure', *e.g.,* `apkmaster.xyz`. These are sites that camouflage

[5] https://www.virustotal.com/

their true nature or identity, or that include elements suggesting latent malign intent.[6] Note, a notable fraction of junk emails are classed as "uncategorized". This is because these junk messages often contain a unknown fringe URLs, that are not present in the domain classifier service.

To explore these risks, we run all URLs through the 73 antivirus engines in VirusTotal. Overall, 26% of URLs in junk messages are tagged as dangerous by 3 or more antivirus, and 15% are tagged by 9 or more *e.g.,* `trycryptocoins.com`, `amazon.bigest-sale-live.in`. In contrast, only 1.5% of non-junk were tagged as malicious by 3 or more engines (0.42% by 9+ engines).

4 JUNK SENDERS AND THEIR ACTIONS

Next we look at the *users* who produce junk in terms of their locations, temporal patterns and group membership.

4.1 Operational definition of junk senders

Our methodology identifies junk messages by their content rather than junk senders directly. Some junk messages may be inadvertently posted by enthusiastic or naïve users who do not realise it is junk. Figure 4 plots the fraction, f, of messages by a user that are marked as junk with respect to total messages. As expected, this follows a bimodal pattern, with junk senders on one end (nearly 100% of their messages are junk) and non-junk senders at the other end (with almost no junk messages). This suggests that users with a junk fraction beyond any reasonable threshold will capture all intentional spammers. In this section, we adopt an operational definition of junk senders as any user who has posted more than $f = 50\%$ messages that our methodology identifies as junk (our results are robust to other similar thresholds). Using this methodology, we identify 17.6K users as junk senders and 32.9K as non-junk senders who share a total of 239K and 1.3M messages, respectively.

4.1.1 Distribution of junk senders across groups.

We start by inspecting the distribution of junk senders across the groups. Figure 3 presents the distribution of (*i*) junk messages per group; (*ii*) junk senders per group; as well as (*iii*) the number of users removed per group. First, we see that 75% of groups have at least one junk sender, with an average of 13 per group (sending 58 junk messages per group). In the most extreme case, we observe 2722 junk messages in a single group. Overall, the top 25% of groups have at least at least 18 junk senders. This confirms that a large fraction of groups suffer with the accumulation of junk messages. Figure 3 also shows that many groups actively filter junk senders: 17% of groups have at least one example of a junk sender being removed. In fact, 2% of groups (around 95) have removed at least 8 users.

4.2 Junk sender locations

We next use the phone number country codes to geolocate all users. Figure 5 presents the results for both junk and non-junk senders. Unsurprisingly, the majority have Indian country codes, although we also see a range of other countries. Interestingly, these third party countries tend to be primarily junk senders. Most striking is Russia, which has 6823 junk senders yet *no* non-junk senders. These users exclusively post junk content (9029 messages spread across 334 text clusters). We also note that all Russian users' junk messages are in English. Similar patterns are seen in phones from other countries, albeit on a smaller scale (*e.g.,* Romania, Kyrgyzstan).

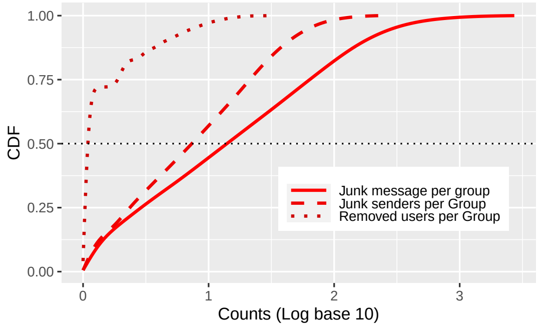


Figure 3: Count of junk messages, junk senders and removed users per group.

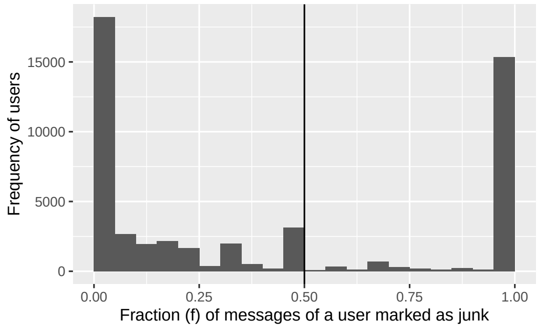


Figure 4: Fraction (f) of messages of a user marked as junk. A user is considered a junk sender if they posted more than $f = 50\%$ junk messages.

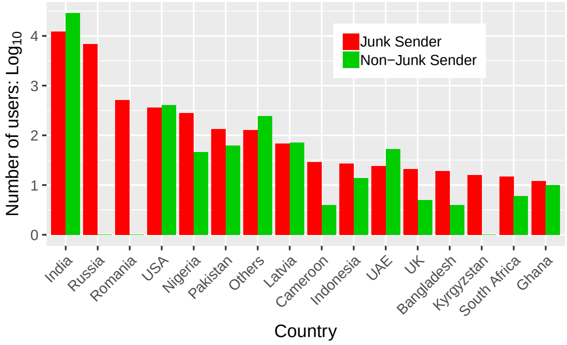


Figure 5: Country codes used by junk *vs.* non-junk senders

4.3 Longevity of junk messages

We now explore the temporal patterns of junk messages. Recall that a cluster refers to group of similar messages (Section 2.2). We term any day when at least 10 messages within a cluster are sent as an *active day* for that particular cluster.

We start by inspecting the lifetime of junk *vs.* non-junk messages. We measure this as the difference between their first and last occurrence, shown in Figure 6. We see that junk messages have

[6] https://bit.ly/34aBZ3V

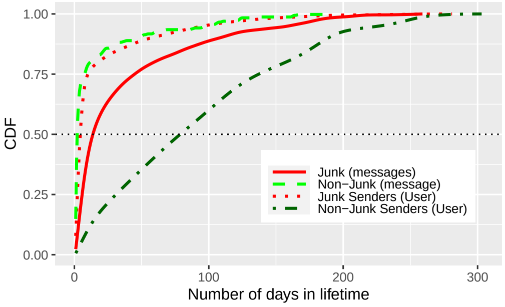


Figure 6: Cumulative Distribution Function (CDF) of lifetimes of junk and junk message clusters and users.

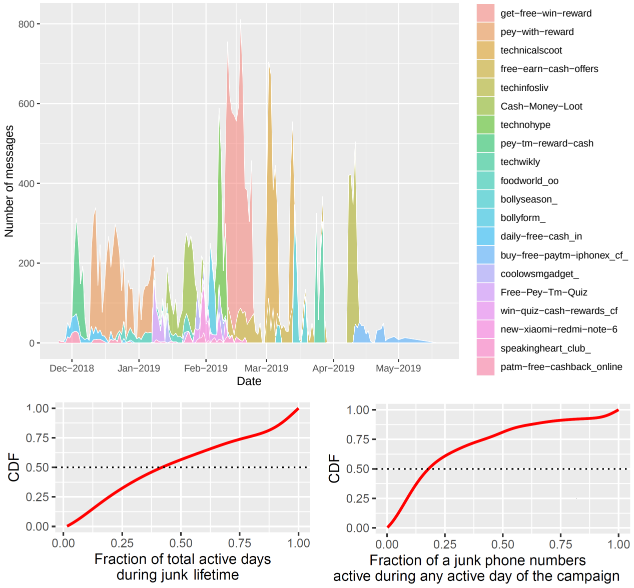


Figure 7: Top: Number of times a day that a junk message is posted, for 15 exemplar clusters. Bottom Left: Fraction of days that are 'active' (> 10 junk messages sent) during the lifetime of multi-day junk clusters. Bottom Right: Fraction of junk phone numbers used during active days.

consistently longer lifetimes than non-junk. The median campaign duration for junk is 29 days compared to 12 days for non-junk. Interestingly, we see the opposite trend when computing the lifetime of senders. Here, we compute the lifetime of a user as the difference between their first and last post (calculated across the user's activity in all groups in our dataset). As shown in Figure 6, non-junk senders have substantially longer lifetimes than their junk senders counterparts, partly due to their removal from groups by admins.

To gain a better understanding of these temporal trends we show 15 highly occurring example junk message from a single cluster in Figure 7 (top). These messages are repeatedly sent across multiple days and are highly focused, with aggressive peaks on a small number of days surrounded with different URLs. To generalise this across all junk campaigns, Figure 7 (bottom left) plots the number of active days seen during the lifetime each cluster. This shows that most are sent across relatively few active days: On average less than half the days during the lifetime of a junk cluster are active with 10 or more messages. This suggests junk senders take a staggered approach, rather than issuing junk messages every day. Despite this, there are a notable set of highly active messaging campaigns where messages are sent on most days: 25% of clusters involve sending messages on at least 80% of the days during their lifetime.

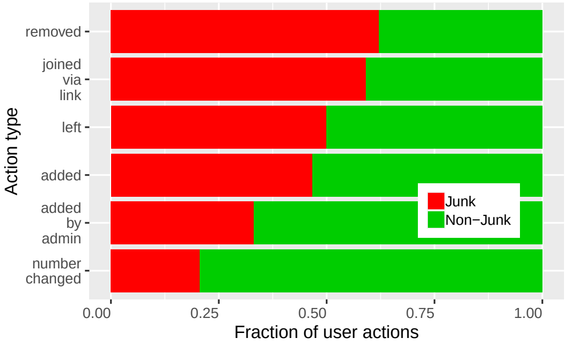


Figure 8: Relative proportions of join and leave actions.

Table 3: Relation between leaving and joining methods for junk senders (equivalent numbers for non-junk senders in parenthesis).

Actions	Joined via link	Added	Added by admin
Left	75% (62%)	17% (20%)	8% (18%)
Number changed	74% (37%)	12% (28%)	14% (35%)
Removed	80% (48%)	5% (18%)	15% (34%)

The above leads us to conjecture that multiple phone numbers may be involved in sending messages for each cluster. To explore this, Figure 7 (bottom right) shows the fraction of the phone numbers involved in sending messages from each cluster that are active each day. We observe that, indeed, messages contained within a cluster are sent from multiple phone numbers. On average, under 20% of the phone numbers are involved on any active day.

4.4 Joining and leaving groups

Finally, we examine how users join and leave different groups. Figure 8 presents the fraction of junk senders *vs.* non-junk senders who join using each technique. A clear difference exists: it is more common for junk senders to join WhatsApp groups via a link. Junk senders also comprise a disproportionate fraction of users who are 'removed' from a group by the admin.

Table 3 examines how users leave groups and how these actions are related to the method they use to join. Users who leave the group by any method are overwhelmingly likely to have joined via an invite link, although this is noticeably higher for junk senders than non-junk senders. Note that users who have been added by a group admin are the least likely to have left the group.

We can further inspect what users *do* after joining. After joining via a link, nearly 40% of junk senders (30% of non-junk senders) post URLs. 18% (5%) post messages with a phone number, and 21% (21%) simply leave. Again for junk senders, after posting a junk message with a URL, 86% of the time the next action is to post another junk message with a URL. We also check the actions of junk senders immediately before they are removed from a group. We find that

54% of the time they posted a junk message with a URL, and 19% of times they post a junk messages with a phone number in it. In total, 73% of user removals by admins are immediately after the user posts a junk message.

5 JUNK MITIGATION STRATEGIES

In this section, we examine how we can automatically identify junk messages. The ultimate goal is to build a tool to (*i*) Help group admins identify junk senders who might need to be removed; and (*ii*) Automatically filter such message proactively for users.

5.1 Content-based junk detection

We start by training a single global model using the entirety of our dataset. To represent the text of messages, we rely on MuRIL (Multilingual Representations for Indian Languages) word embeddings [26]. For each message in the 7k (3.5k junk) unique messages dataset that we label (Table 2), we obtain MuRIL embedding vectors of size 768, as described in [26]. These 768 values constitute our input features. We experiment with three binary classification models, for which the target is to classify messages into junk *vs.* non-junk: Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF). We use a 80:20 train:test split using Sklearn library in Python 3. We tune the hyper-parameters in each model using Sklearn's GridSearchCV module. Finally, we tune LR (C: 100 ; penalty:L1, solver=liblinear), SVM (kernal:rbf; C:100) and RF (number of estimators:400, min split:2; criterion: entropy), each with five-fold cross validation.

The results are presented in Table 4 after testing with various combinations of penalty, estimators and optimisation criterion. We find that RF and SVM both perform well on embeddings (nearly 88% accurate and 0.88 F-1 Score). This confirms that there *are* sufficient determinant features to assist users with automated classification.

5.2 Metadata-based junk detection by platform

A limitation of the above strategy is that training and classification requires access to raw text content. This, however, is challenging for the platform to implement due to WhatsApp's end-to-end encryption. Thus, models cannot easily be trained centrally, and text-based inference *must* take place on the recipient's end device. However, on-device ML may be challenging for low power mobile phones common in countries like India.

With this in mind, we next experiment with an alternative approach that can be computed centrally by the WhatsApp platform, without access to text content. Our key insight is that *although content is encrypted, the actions of the users, such as joining or leaving groups are still visible.* That said, some of the most important features are use of phone numbers and URLs in text, which are contained in over 90% of junk messages (§3.2.2). Thus, we propose a simple (optional) modification wherein each sender's WhatsApp client encodes a 2-bit signal on whether the message sent contains a phone number, a URL, both or neither. The truthfulness of this signal can be verified by the recipients after decrypting, and the signal (though not the actual phone number or URL) can be made visible to the platform without compromising privacy.

Using this, we centrally build *user profiles* upon which classification can be performed. This contains each user's actions per-group

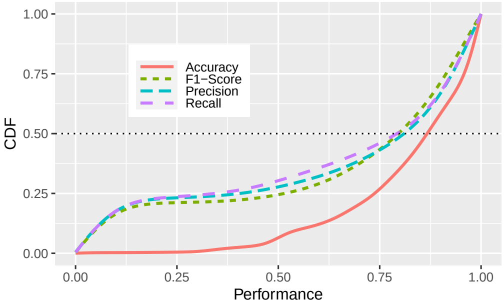


Figure 9: CDF of metrics for per-group Random Forest models.

as a vector (or features) of counts for the different types of actions (*e.g.,* number of times they joined/left a group). Table 5 (column 1) provides the full list of features. Using these features, we again try a number of different methods including Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF). As a dataset, we consider all users with at least 2 actions. This leaves us with 47K user profiles with 15K removals and 32K non-removals across 3.6K groups. We again perform a random 80:20 five-fold cross-validation split for training and testing. We tune hyper-parameters using the same approach as in the previous subsection, resulting in LR (C=0.001, solver='sag'; penalty:L2), SVM (kernal:rbf; C:1000) and RF (number of estimators: 500, min split: 2; criterion: entropy).

Our metadata-based classifier achieves similar performance to that of content-based modelling, with an F1-score (for best RF model) of 0.9 and accuracy of 87.5% (Table 4). We also compare against an alternative model where our classifier does not have access to the 2 bit signal proposed above (this prevents the classifier from checking the presence of URLs or phone numbers). This decreases the F1-score to 0.67 and the accuracy to 0.82, indicating that even without this adaptation, our approach still can flag the majority of junk senders. When inspecting feature importance, we observe that the most important feature is the number of messages posted (0.52 for RF), followed by the use of a non-domestic phone number (0.15 for RF). In cases where the 2-bit signal is not available the use of international phone number becomes the most important feature (0.42 for RF).

5.3 Content & metadata detection by device

Finally, we revisit the idea of performing local classification on the end device, such that we can use *both* text content and metadata features simultaneously without undermining end-to-end encryption. This is possible because each user can gain vantage on the text content contained within their own groups. Thus, for each group, we construct a local on-device profile for each participating members. This user profile contains all the features shown in Table 5, as well as the fraction of messages sent that are tagged as junk by content classifier. Importantly, these features can be locally constructed by any user in the group and do not require central computation.

We train a local Random Forest model on a *per-group* basis (using the 3/4 of groups in our data that contain at least one junk sender). Here, we assume that supervised labels for local training can be obtained by annotation (*e.g.,* by the group admin). Our results show

Table 4: Junk detection performance using machine learning based classifiers.

Features type	Model	Accuracy	F-1	Recall	Precision
Content (Word Embedding)	Logistic Regression	87.9%	0.88	0.87	0.89
Content (Word Embedding)	SVM	87.5%	0.88	0.87	0.88
Content (Word Embedding)	Random Forest	**88.1**%	0.88	0.88	0.89
Metadata (All action+ISD)	Logistic Regression	67.5%	0.8	0.67	0.97
Metadata (All action+ISD)	SVM	74.4%	0.82	0.74	0.93
Metadata (All action+ISD)	Random Forest	**87.5%**	**0.90**	**0.87**	**0.94**
Content+Action per group	Random Forest (Per group model)	86% (mean); sd=0.18	0.63 (mean); sd=0.42	0.63 (mean); sd=0.44	0.64 (mean); sd=0.43

that only a small number of training samples would need to be collectively tagged by the admin and/or end users (a few hundred).

In our experiment, we use 80% of users in each group for training, and 20% for testing, tuned similarly as shown in models in previous sub-section. Table 4 summarises the results. We obtain 86% mean accuracy across groups, which is close to the global model (0.9). Figure 9 shows the distribution of performance scores across each WhatsApp group. A minority of groups obtain poor performance, largely due to the small number of users. For groups that gain under 60% accuracy, we see just 300 messages and 35 users on average. Such groups contain insufficient data to effectively train the local models. That said, 77% of groups obtain accuracy exceeding 75%. These tend to have larger populations, allowing end devices to better learn. We therefore confirm that our classifier *can* assist admins in identifying junk senders to remove, and assist users in proactively filtering such messages simultaneously.

6 RELATED WORK

Several studies have explored messaging patterns within other community mediums, *e.g.,* Reddit [45], 4chan [8, 20] and IRC [42]. Although there also have been studies of junk sending on social media [15, 46, 49], email [12] and SMS [36], junk has not yet been studied on WhatsApp except for anecdotal observations [1, 3, 10, 35, 47]. These studies have largely focussed on qualitative methodologies, *e.g.,* interviews, surveys, focus groups. For example, [6] collected quantitative data via the installation of a logging tool on user devices. By recruiting 70 participants, they analysed 58K sent messages. Although powerful, this approach is largely non-scalable. Other messaging apps, such as WeChat [21], have been explored at scale although the focus has not been on the content and interactions. Instead, coarser analyses have been performed, *e.g.,* size of messages. Studies that have explored more social features have, again, limited themselves to small-scale surveys [28].

There have been prior studies on the misuse of messaging services more generally, *e.g.,* SMS fraud in Pakistan [36]. There are also a small set of related works looking at the dissemination of malicious content via WhatsApp groups. These, however, are largely focused on misinformation [23, 24]. Resende et al. [41] investigated the dissemination of misinformation during the Brazilian elections, and the impact of introducing limits on message forwarding [13]. In another similar study of Brazilian elections, Victor *et al.* [11] found partisan activities in political groups. Reis [40] proposed an architecture to flag misinformation in WhatsApp without breaking end-to-end encryption. Unlike our approach, this relies on a manually annotated set of image hashes.

In studying the presence of misuse, there have been a works attempting to detect unsolicited spam campaigns and click-baits, primarily via emails or social media. For instance, Xiao et al. [48] used supervised learning methods to detect groups of fake spam accounts on social media. This include tools for detecting unwanted videos in social media [7, 19, 38, 44]. Boykin and Roychowdhury [9] investigated the use of social graphs to filter spam. There are various content-based approaches to detecting junk too [2, 4, 30]. These tend to rely on building document models and training machine classifiers (*e.g.,* SVMs) to detect spam messages. In the case of WhatsApp, this can be problematic due to its use of end-to-end encryption. We consider such studies orthogonal to our work.

7 CONCLUSIONS

This paper has presented the first study of junk messages in public WhatsApp groups. We have gathered 2.6 million messages from 5,051 public politics-related groups in India, and analysed the content, URLs and temporal patterns of spam. We find that junk is commonplace on WhatsApp, and senders tend to post across a large number of groups. They also exhibit interesting patterns of leaving and rejoining groups multiple times, to avoid being removed by admins. We further find evidence of campaigns — spreading the same spam message over a small number of 'active' days. These strategies may help improve the visibility of junk, by providing a longer 'shelf life' in the recent messages.

Our results have clear implications for junk detection. For example, a key indicator of junk is the presence of particular URLs and phone numbers. We have shown that this can be used as part of automated detection, and have demonstrated that models can be trained to detect WhatsApp junk. This can assist admins in quickly flagging and removing such users. However, as WhatsApp uses end-to-end encryption, such information cannot be accessed by the platform. We have therefore proposed techniques that can be used by the end device or the platform (centrally), whilst still respecting end-to-end encryption guarantees. To aid reproducibility, our annotated dataset and code is publicly available for researchers.[7]

Acknowledgements

We thank the annotators from India. This work is partially supported by EU H2020 grant agreements No 830927 (Concordia) and No 101016509 (Charity). Gareth Tyson was supported by grant EP/S033564/1. Kiran Garimella did the work at MIT, supported by a Michael Hammer Postdoctoral Fellowship.

[7] More information is available at: http://tiny.cc/netsys-whats-app

REFERENCES

[1] Pushkal Agarwal, Kiran Garimella, Sagar Joglekar, Nishanth Sastry, and Gareth Tyson. 2020. Characterising user content on a multi-lingual social network. In *ICWSM*.

[2] Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of SMS spam filtering: new collection and results. In *Symposium on Document engineering*.

[3] Venkat Ananth and Samidha Sharma. 2019. On Instagram, in India, it's sex for sale. The Economic Times. (2019). https://economictimes.indiatimes.com/tech/internet/on-instagram-in-india-its-sex-for-sale/articleshow/67968645.cms?from=mdr

[4] Serkan Ballı and Onur Karasoy. 2018. Development of content-based SMS classification application by using Word2Vec-based feature extraction. *IET Software* (2018).

[5] Shakuntala Banaji et al. 2019. WhatsApp vigilantes: an exploration of citizen reception and circulation of WhatsApp misinformation linked to mob violence in India. *Dept. of Media and Communications, LSE* (2019).

[6] Agathe Battestini, Vidya Setlur, and Timothy Sohn. 2010. A large scale study of text-messaging use. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*. ACM.

[7] Fabrício Benevenuto et al. 2009. Detecting spammers and content promoters in online video social networks. In *ACM SIGIR*.

[8] Michael S Bernstein, Andrés Monroy-Hernández, Drew Harry, Paul André, Katrina Panovich, and Gregory G Vargas. 2011. 4chan and /b/: An Analysis of Anonymity and Ephemerality in a Large Online Community.. In *ICWSM*.

[9] P Oscar Boykin and Vwani P Roychowdhury. 2005. Leveraging social networks to fight spam. *IEEE* (2005).

[10] Gadgets Now Bureau. 2021. WhatsApp bans Indian accounts. (2021). bit.ly/3nyr0br.

[11] Victor S Bursztyn and Larry Birnbaum. 2019. Thousands of small, constant rallies: A large-scale analysis of partisan WhatsApp groups. In *ASONAM*. IEEE.

[12] Gordon V Cormack. 2008. *Email spam filtering: A systematic review.* Now Publishers.

[13] de Freitas Melo et al. 2019. Can WhatsApp Counter Misinformation by Limiting Message Forwarding?. In *Conf. Compl. Netw. and Appl.*

[14] DUTA. 2018. Duta.in: BRINGING THE INTERNET TO THE NEXT BILLION. (2018). duta.in.

[15] Hongyu Gao et al. 2010. Detecting and characterizing social spam campaigns. In *SIGCOMM IMC*.

[16] Kiran Garimella and Gareth Tyson. 2018. Whatapp doc? A first look at whatsapp public group data. In *ICWSM*.

[17] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *VLDB*.

[18] Aakriti Gupta, Sunil Kumar Singh, Kabir Ahuja, and Ankit Gupta. 2019. Good Morning Turning to Spam Morning. In *ICICCT*.

[19] Nestor Hernandez, Mizanur Rahman, Ruben Recabarren, and Bogdan Carbunar. 2018. Fraud De-Anonymization for Fun and Profit *(CCS)*. ACM.

[20] Gabriel Emile Hine, Jeremiah Onaolapo, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Riginos Samaras, Gianluca Stringhini, and Jeremy Blackburn. 2017. Kek, Cucks, and God Emperor Trump: A Measurement Study of 4chan's Politically Incorrect Forum and Its Effects on the Web.. In *ICWSM*.

[21] Qun Huang, Patrick PC Lee, Caifeng He, Jianfeng Qian, and Cheng He. 2015. Fine-grained dissection of WeChat in cellular networks. In *Quality of Service (IWQoS), 2015 IEEE 23rd International Symposium on*. IEEE.

[22] Muhammad Ikram, Rahat Masood, Gareth Tyson, Mohamed Ali Kaafar, Noha Loizon, and Roya Ensafi. 2019. The Chain of Implicit Trust: An Analysis of the Web Third-party Resources Loading. *Web Conference* (2019).

[23] R Tallal Javed, Mirza Elaaf Shuja, Muhammad Usama, Junaid Qadir, Waleed Iqbal, Gareth Tyson, Ignacio Castro, and Kiran Garimella. 2020. A First Look at COVID-19 Messages on WhatsApp in Pakistan. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 118–125.

[24] R Tallal Javed, Muhammad Usama, Waleed Iqbal, Junaid Qadir, Gareth Tyson, Ignacio Castro, and Kiran Garimella. 2022. A deep dive into COVID-19-related messages on WhatsApp in Pakistan. *Social Network Analysis and Mining* 12, 1 (2022), 1–16.

[25] Matt Jones. 2017. How WhatsApp Reduced Spam while Launching End-to-End Encryption. (2017). usenix.org/conference/enigma2017/conference-program/presentation/jones

[26] Simran Khanuja et al. 2021. MuRIL: Multilingual Representations for Indian Languages. Arxiv, tfhub.dev/google/MuRIL/1. (2021).

[27] Dae Wook Kim et al. 2015. Detecting fake anti-virus software distribution webpages. *Computers & Security* (2015).

[28] Che Hui Lien and Yang Cao. 2014. Examining WeChat users' motivations, trust, attitudes, and positive word-of-mouth: Evidence from China. *Computers in Human Behavior* (2014).

[29] CSDS Lokniti. 2018. How widespread is WhatsApp's usage in India? Live Mint. (2018). livemint.com/Technology/O6DLmIibCCV5luEG9XuJWL/How-widespread-is-WhatsApps-usage-in-India.html

[30] Jialin Ma et al. 2016. Intelligent SMS spam filtering using topic model. In *Intl. Conf. on Intelligent Netw. and Collab. Systems (INCoS)*.

[31] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My phone and me: understanding people's receptivity to mobile notifications. In *CHI*.

[32] Lincoln Mullen. 2015. textreuse: Detect text reuse and document similarity. *rOpenSci* (2015).

[33] Nic Newman, Richard Fletcher, Antonis Kalogeropoulos, and Rasmus Kleis Nielsen. 2019. Reuters Institute Digital News Report 2019 . (2019).

[34] J Ooms and D Sites. 2018. cld2: Google's Compact Language Detector 2. (2018).

[35] Priya Pathak. 2019. WhatsApp is banning 2 million accounts every month. India Today. (2019). indiatoday.in/technology/features/story/whatsapp-is-banning-2-million-accounts-every-month-and-will-ban-yours-if-you-do-any-of-these-things-1450211-2019-02-07

[36] Fahad Pervaiz et al. 2019. An assessment of SMS fraud in Pakistan. In *ACM CCS*.

[37] Martin Pielot, Karen Church, and Rodrigo De Oliveira. 2014. An in-situ study of mobile phone notifications. In *International conference on Human-computer interaction*.

[38] Mizanur Rahman, Nestor Hernandez, Ruben Recabarren, Syed Ishtiaque Ahmed, and Bogdan Carbunar. 2019. The Art and Craft of Fraudulent App Promotion in Google Play *(CCS)*. ACM.

[39] Elissa M Redmiles, Neha Chachra, and Brian Waismeyer. 2018. Examining the demand for spam: Who clicks?. In *CHI*.

[40] Julio CS Reis, Philipe Melo, Kiran Garimella, and Fabrício Benevenuto. 2020. Can WhatsApp benefit from debunked fact-checked stories to reduce misinformation? *Harvard Misinformation Review* (2020).

[41] Gustavo Resende et al. 2019. (Mis) Information Dissemination in WhatsApp: Gathering, Analyzing and Countermeasures. In *Web Conference*.

[42] E Sean Rintel, Joan Mulholland, and Jeffery Pittam. 2001. First things first: Internet relay chat openings. *Journal of Computer-Mediated Communication* (2001).

[43] Punyajoy Saha, Binny Mathew, Kiran Garimella, and Animesh Mukherjee. 2021. Short is the Road that Leads from Fear to Hate: Fear Speech in Indian WhatsApp Groups. In *Web Conference*.

[44] Hyejin Shin, WooChul Shim, Saebom Kim, Sol Lee, Yong Goo Kang, and Yong Ho Hwang. 2021. # Twiti: Social Listening for Threat Intelligence. In *Web Conference*.

[45] Philipp Singer, Fabian Flöck, Clemens Meinhart, Elias Zeitfogel, and Markus Strohmaier. 2014. Evolution of reddit: from the front page of the internet to a self-referential community?. In *Web Conference*. ACM.

[46] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2010. Detecting spammers on social networks. In *CCS*.

[47] Thread-Reader. 2020. Fake Flipkart website. (2020). bit.ly/fake-FK

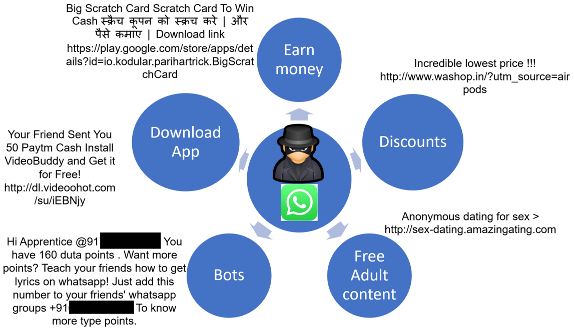
[48] Cao Xiao, David Mandell Freeman, and Theodore Hwa. 2015. Detecting clusters of fake accounts in online social networks. In *Wksp. AI & Security*.

[49] Sarita Yardi, Daniel Romero, Grant Schoenebeck, et al. 2010. Detecting spam in a twitter network. *First Monday* (2010).

APPENDIX

Example Junk Messages

Figure 10 shows several examples of junk messages. Table 5 summarises the most significant features in our Random Forest Classifier (see §5.2).



Big Scratch Card Scratch Card To Win Cash स्क्रैच कूपन को स्क्रच करे | और पैसे कमाए | Download link https://play.google.com/store/apps/details?id=io.kodular.parihartrick.BigScratchCard

Incredible lowest price !!! http://www.washop.in/?utm_source=airpods

Anonymous dating for sex > http://sex-dating.amazingating.com

Hi Apprentice @91[redacted] You have 160 duta points . Want more points? Teach your friends how to get lyrics on whatsapp! Just add this number to your friends' whatsapp groups +91[redacted] To know more type points.

Your Friend Sent You 50 Paytm Cash Install VideoBuddy and Get it for Free! http://dl.videoohot.com/su/iEBNjy

Figure 10: Some top examples of junk messages.

Table 5: Feature importance of Random Forest Classifier to separate non-jettisons and jettisons.

Feature	With 2-bit signal	No 2-bit signal
Posted simple message	0.52	0.37
Non-domestic number	0.15	0.42
Posted URL	0.12	N/A
Joined via link	0.08	0.075
Posted phone number	0.05	N/A
Left group	0.04	0.05
Added by member	0.023	0.03
Added by admin	0.021	0.025
Removed from group	0.01	0.015
Number changed	0.003	0.003