Privacy-Preserving Abuse Detection in Future Decentralised Online Social Networks

Álvaro García-Recuero, Jeffrey Burdges, Christian Grothoff

Privacy-Preserving Abuse Detection in Future Decentralised Online Social Networks

Álvaro García-Recuero[1,2], Jeffrey Burdges[1], and Christian Grothoff[1]

[1] INRIA Rennes - Bretagne Atlantique
firstname.secondname@inria.fr
[2] Université de Rennes 1, Rennes, France
alvaro.garcia-recuero@univ-rennes1.fr

Abstract. Future online social networks need to not only protect sensitive data of their users, but also protect them from abusive behavior coming from malicious participants in the network. We investigate the use of supervised learning techniques to detect abusive behavior and describe privacy-preserving protocols to compute the feature set required by abuse classification algorithms in a secure and privacy-preserving way. While our method is not yet fully resilient against a strong adaptive adversary, our evaluation suggests that it will be useful to detect abusive behavior with a minimal impact on privacy.

1 Introduction

Users of online social networks (OSNs) currently face two systemic issues to their well-being: mass-surveillance and abusive behavior. Mass-surveillance in OSNs is a real threat for modern liberal societies [7]. OSN platform providers do not just need to self-impose limits on users' behavior[3], but now also avoid governments imposing draconian penalties to participants[4]. Abusive behavior where users in the OSN platform [9] or governments [14] send messages designed to harm potential victims, has been established as a significant risk factor for suicide [13] and a previous study is reporting it almost doubling the number of attempted suicides [8].

Future decentralised OSN designs such as [11] propose to protect users against censorship and mass-surveillance by decentralizing the OSN; namely establishing secure end-to-end encrypted communication between all participants, hiding meta data at the network level, and allowing pseudonymous interactions between participants. Thus it becomes plausible to address mass-surveillance threats. However, at the same time one would expect that threats from abusive behavior are likely to increase: Major centralised OSNs provide some safeguards, such as the Facebook-Imune-System (FIS) [15], to detect and block abusive behavior. Naturally, these centralised solutions typically exploit the comprehensive data available about the platform's users. Thus, these existing techniques will

[3] https://twitter.com/rules
[4] http://www.bbc.com/news/technology-16810312

not work in a privacy-preserving decentralised OSNs, where some of the data is not supposed to be exposed due to privacy constraints, and other data may be easily falsified by an adversary.

In this paper, we describe key building blocks for building a privacy-preserving abuse detection system for future decentralised OSNs. As starting point we evaluate our abuse detection system with data from a centralised OSN, specifically the second largest one as of today, Twitter. Our assumption is that the interaction culture would remain similar between today's Twitter and a future decentralised OSN, and thus the results for analyzing abusive vs. non-abusive interaction patterns would carry over. Like the FIS, we use supervised learning to classify messages as acceptable or abusive. By incorporating a broad set of features based on publicly available data from Twitter, we establish a baseline for the accuracy of the method without privacy-preservation or adversarial adaptation to the method. We then study which features could be securely obtained without unduly exposing sensitive data about users. Here, we propose two new privacy-preserving protocols for secure set intersection, showing how efficient secure multiparty computation can assist in obtaining key features for abuse detection. We finally evaluate supervised learning using the resulting restricted feature set to demonstrate the utility of the method.

2 Defining abuse

Before we can evaluate abuse detection methods, we need a definition of abusive behavior. From the rich literature on abuse, we found the Joint Threat Research Intelligence Group (JTRIG) of the British Government Communication Head Quarter (GCHQ) provided the most comprehensive and still reasonably simple definition in their characterization of their own work. JTRIG manipulates global opinion using techniques that they characterize with the four Ds: [14].

- Deny: They encourage self-harm to others users, promoting violence (direct or indirect), terrorism or similar activities. (This denies the victim health or even life, which are human rights.)
- Disrupt: They disrupt activities they disagree with using distracting provocations, denial of service, flooding with messages and generally promoting abuse of the intended victim.
- Degrade: They disclose personal and private data of others without their approval as to degrade their public image/reputation.
- Deceive: They deceive by spreading false information, including supplanting a known user identity (impersonation) for influencing other users behavior and activities, or assuming false identities. (The use of pseudonyms that are recognizable as such is not a deception.)

We will now argue that these four terms nicely cover common characterizations of abusive behavior.

Several studies have defined cyber-bullying as the act of harassing another person via any form of digital communications. This behavior is intended to *degrade* the self-esteem or image of the victim [10].

According to[5], an "Internet troll" or "cyber-troll" is a member of an online community who posts abusive comments at worst or divisive information at best to repeatedly create controversy. These actions are covered by the terms *disrupt* and possibly *deceive*.

Trolldor[6] allows users to search for the statistics of a particular user in Twitter, and report him as "troll". Key reasons Trolldor lists for users to report a Twitter profile as a "troll" to Trolldoor include:

- Provocation: users who just look to provoke for fun (*disrupt*)
- Creep: users who fill other users timeline on a daily basis with messages worshiping their idols, friends, relatives and colleagues. (*disrupt*)
- Retweeter/Favoriter: users who never create their own content and just retweet and favorite other peoples messages.
- Insult/Threat: users who insult or threaten other users. (threats *deny*)
- False identity: profiles that seek to usurp anothers identity (*deceive*)

Twitter's guidelines on abusive behavior explicitly prohibit: violent threats (*deny*), harassment (*degrade*), hateful conduct (*deny*), multiple account abuse (*deceive*), private information disclosure (*degrade*), impersonation (*deceive*), promotion of self-harm (*deny*), and spam (*disrupt*).

The examples demonstrate that the four *Ds* cover common definitions of abusive behavior.

3 Data model

We consider two directed graphs whose set of vertices $\mathcal{V}$ represent the about one million user profiles collected from the OSN, Twitter. Let $\mathcal{G}_f=(\mathcal{V},\mathcal{E}_f)$ be a directed graph of subscription relationships, where an edge $(a,b)\in\mathcal{E}_f$ represents that user a is subscribed to posts from user b.

Let $\mathcal{G}_m=(\mathcal{V},\mathcal{E}_m)$ be a directed multi-graph of messaging relationships, where an edge $(a,b)\in\mathcal{E}_m$ implies that a directed a message specifically to b (on Twitter, this is done by either mentioning `@b` or by responding to a message from b). Note that $\mathcal{E}_m$ does not contain all messages that a broadcasts to all its subscribers, so it models the messages that are shown in the notifications of the user mentioned (`@b`), and which are thus a vector for potential abusive behavior.

To establish ground truth about abuse, we asked six reviewers to use JTRIG's four *Ds*-definition to manually annotate about 1000 Twitter messages as abusive, acceptable or undecided. The resulting data set (Table 3) provides the ground truth for supervised learning and evaluation of the methods presented in this paper.

Reviewers frequently disagreed about a message. For the *# agreement* value, we computed the agreement among the *other* reviewers and then checked whether this reviewer was in agreement with the rest of the reviewers about a tweet. On

[5] What does Troll mean? `http://www.techopedia.com/definition/429/troll`

[6] `http://trolldor.com`

reviewer	# reviews	% abusive	% accept.	# agreement	c-abusive	c-accept.	c-overall
1	754	3.98	83.55	703	0.71	0.97	0.93
2	744	4.30	82.79	704	0.66	0.97	0.94
3	559	5.01	83.90	526	0.93	0.95	0.94
4	894	4.03	71.92	807	0.61	0.94	0.90
5	939	5.54	69.54	854	0.88	0.90	0.91
6	1003	5.68	69.79	875	0.95	0.89	0.87
average	815	4.76	76.92	**745**	**0.79**	**0.94**	**0.92**
std. dev.	162	0.76	7.18	130	0.15	0.03	0.03

Table 1. Human baseline statistics. The c-values are explained in this Section, 3.

average, reviewer's ratings matched with the agreement among the other reviewers 745 times, corresponding to 92% of the reviewed messages (*c-overall*). *c-abusive* provides the agreement on abusive messages and *c-accept.* the agreement for acceptable messages. As expected, agreement on abusive messages is significantly lower: the reviewers agreed on about 79% of the abusive messages, and on over 94% of the acceptable messages.

4 Learning without privacy

At a high-level, the system has each user locally compute whether a message is likely to be abusive, and then allowing the user's software to take appropriate action, such as giving messages that are likely to be abusive a lower relevance in the user's timeline ranking. For this computation, the decision process should only use data that is available in the vicinity of the respective user. This approach ensures that the computation is compatible with decentralised OSNs that lack a central service provider.

Table 2 summarizes the feature set we used to evaluate abusive behavior. We experiment with various supervised models [1] from *scikit-learn*[7]. We present data from those classifiers that performed best. Specifically, we compare decision trees (DT), random forest (RF), extra trees (ET) and the gradient boosting (GB) classifier [3]. We also configure all our classifiers with a "depth" of eight, and using "balanced" for the "class weight" option. While we tried other supervised earning algorithms such as logistic regression, k-means clustering and NB-trees, the aforementioned tree-based methods performed best, and thus we limit our presentation to those.

A lower bound for the performance of the classifiers is provided by a base rate classifier (BR), where each messages is classified according to the most predominant class (acceptable in our case). This classifier classifies all abusive messages incorrectly, and all acceptable messages correctly. An upper bound for our performance expectations is the human baseline classifier (HB), described in

[7] `http://scikit-learn.org/stable/supervised_learning.html`

	Feature	Description
5.1	# lists	how many lists the sender has created
	# subscriptions	number of subscriptions of the sender
	$\frac{\#\text{ subscriptions}}{\text{age}}$	ratio of subscriptions made in relation to age of sender account
	$\frac{\#\text{subscriptions}}{\#\text{subscribers}}$	ratio of subscriptions to subscribers of sender
5.2	# mentions	number of mentions in the message
	# hashtags	number of hashtags in the message
	$\frac{\#\text{ mentions}}{\#\text{ messages}}$	ratio of mentions made in relation to messages written the sender
	# retweets	number of retweets the sender has posted
	# favorited messages	number of messages favorited by sender
5.3	message invasive	false if sender subscribed to receiver and receiver subscribed to sender
5.4	$\frac{\#\text{ messages}}{\text{age}}$	ratio number of messages in relation to age of sender account
5.5	age of account	days since sender account creation
5.6	# subscribers	number of subscribers to public feed of the sender
	$\frac{\#\text{ subscribers}}{\text{age}}$	ratio of subscribers in relation to age of sender account
5.7	subscription ∩ subscription	size of the intersection among subscriptions of sender and receiver
5.8	subscriber ∩ subscriber	size of the intersection among subscribers of sender and receiver
5.9	$\text{subscriber}^r \cap \text{subscription}^s$	size of the intersection among subscribers of receiver and subscriptions of sender
	$\text{subscription}^r \cup \text{subscriber}^s$	size of the intersection among subscriptions of receiver and subscribers of sender

Table 2. Features, ordered following use in Section 5

Classifier	Metric	Arithmetic Mean	Geometric Mean	Only Acceptable	Only Abusive
HB	Precision	0.87 ± 0.09	0.86 ± 0.07	0.94 ± 0.03	0.79 ± 0.15
	Recall	0.76 ± 0.06	0.72 ± 0.03	0.98 ± 0.01	0.53 ± 0.10
	F-score	0.80 ± 0.07	0.78 ± 0.05	0.96 ± 0.02	0.63 ± 0.12
BR	Precision	0.48 ± 0.00	0.00 ± 0.00	0.95 ± 0.01	0.00 ± 0.00
	Recall	0.50 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00
	F-score	0.49 ± 0.00	0.00 ± 0.00	0.98 ± 0.00	0.00 ± 0.00
DT	Precision	0.66 ± 0.10	0.59 ± 0.04	0.98 ± 0.01	0.35 ± 0.19
	Recall	0.77 ± 0.09	0.76 ± 0.10	0.94 ± 0.06	0.61 ± 0.18
	F-score	0.70 ± 0.10	0.64 ± 0.07	0.96 ± 0.03	0.43 ± 0.18
RF	Precision	0.73 ± 0.08	0.69 ± 0.04	0.98 ± 0.01	0.49 ± 0.17
	Recall	0.74 ± 0.10	0.70 ± 0.11	0.97 ± 0.05	0.51 ± 0.24
	F-score	0.73 ± 0.05	0.68 ± 0.04	0.97 ± 0.02	0.48 ± 0.10
ET	Precision	0.62 ± 0.07	0.51 ± 0.04	**0.99** ± 0.01	0.26 ± 0.14
	Recall	**0.82** ± 0.14	0.81 ± 0.12	0.89 ± 0.06	**0.74** ± 0.26
	F-score	0.66 ± 0.10	0.59 ± 0.08	0.93 ± 0.04	0.38 ± 0.16
GB	Precision	**0.87** ± 0.25	0.87 ± 0.07	0.98 ± 0.01	**0.77** ± 0.49
	Recall	0.74 ± 0.05	0.70 ± 0.05	**0.99** ± 0.04	0.49 ± 0.06
	F-score	**0.78** ± 0.12	0.75 ± 0.07	**0.98** ± 0.02	**0.58** ± 0.22

Table 3. Evaluation of classifiers trained using 5-fold cross validation

Section 3. While the classification algorithms have additional data available to them, it is unrealistic for them to perform better than the individual reviewers who provided the ground truth. Table 3 summarizes the results of the evaluation. The key result is that even without extensive tuning, extra trees (ET) and gradient boosting (GB) perform surprisingly well, with accuracies comparable to those of individual reviewers.

5 Privacy-preserving learning

We now consider how to adapt the abuse detection algorithm to a decentralised privacy-preserving OSN, where we face an adaptive adversary who will change his behavior to evade detection. In this setting, we need to consider how to obtain the numeric value in a way that respects the privacy constraints, and how to make it difficult for an attacker to *forge or falsify* the value of a given feature.

5.1 Account properties

Various features reflect properties of the sender's account that are entirely under the control of the sender. This includes the number of lists the user has created and the set of subscriptions made by the sender. Given an adaptive adversary who knows how the abuse detection algorithm uses these features, we have to assume that the adversary can freely adapt these properties and thus deliberately manipulates all such features.

5.2 Message properties

This feature simply counts the number of times a message contains some of the special functions available in existing OSNs, such mentioning users (`@user`) or highlighting a topic (`#hashtags`) in Twitter.

These two are examples of message properties that are trivial to evaluate locally. The first one (mentions) seem to have negative implications for privacy when the computation is performed by the receiver, while the latter does not.

In case of mentions, adaptive adversaries may again shape their messages as to avoid a true positive in abuse classification, but possibly at the expense of being less effective at hurting the victim (e.g., not being able to mention her, thus not disrupting).

5.3 Message is invasive

The feature "message invasive" is a predicate that is false if sender and receiver of the message are mutual subscribers, that is both the sender subscribes to the receiver, and the receiver subscribes to the sender. If either party is not subscribed to the other, the message is considered "invasive". Table 4 shows that messages that are invasive are more likely to be abusive.

	acceptable	abusive
invasive	440	31
non-invasive	196	1

Table 4. Relationship between abusive behavior and invasiveness.

The predicate is trivial to evaluate locally, as both parties know their subscriptions and their subscribers. While an attacker can easily subscribe to the victim, it would be hard to convince a victim to subscribe to the attacker's feed.

5.4 Messages per day

The feature "messages over age" represents the number of public messages sent of average by a user to all of its subscribers each day. The CCDF shows no clear trend as to whether abusive users in our data set send fewer or more messages per day (Figure 1). To establish this value securely, a user could subscribe to the public feed and observe the message stream. As these are public messages, there is no privacy concern. Subscribing would—with some delay—provide an accurate count of the number of messages made per day.

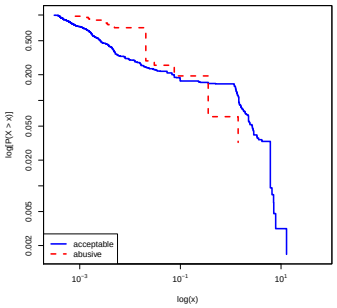


Fig. 1. CCDF of messages/day.

By supporting anonymous subscriptions and gossip-based message distribution, an OSN could make it difficult for an adversary to give the victim an inaccurate view of the public message stream of the adversary.

Naturally, the adversary may be able to adapt by sending fewer or more messages, but this may have an adverse and indirect impact into other features, particularly the adversary subscriber base. A similar analysis holds for features like "retweets" and "favorited messages".

5.5 Age of account

The "age of account" feature considers how many days ago the account was created. The classifiers generally assume that older accounts are less likely to exhibit abusive behavior (which is supported by the CCDF in Figure 2). Thus, an adversary has an interest in making his accounts look old. Using the age of an account is not privacy sensitive, as it hardly can be considered to be sensitive personal information about the user.

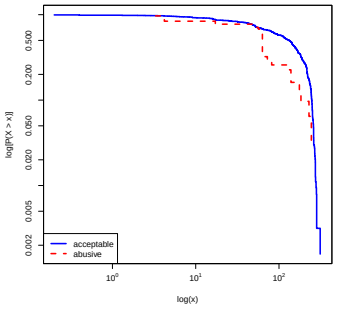


Fig. 2. CCDF of age of account.

In a fully decentralized network, a time-stamping service [6] can be implemented to prevent malicious participants from backdating the time at which their account was created. Naturally, a time-stamping service does not prevent an adversary from creating dormant accounts to be used at a later time for attacks. However, time-stamping raises the bar in terms of required planning, and is thus unlikely to be defeated by non-professional trolls.

5.6 Number of subscribers

The feature "subscribers count" represents the number of subscribers of the user sending the message. Figure 3 shows that there is no clear trend in our data set between abusive and non-abusive senders. It is conceivable that this is because the feature is trivial to manipulate: creating new accounts is generally relatively cheap, and there are even existing blackmarkets for Twitter [16].

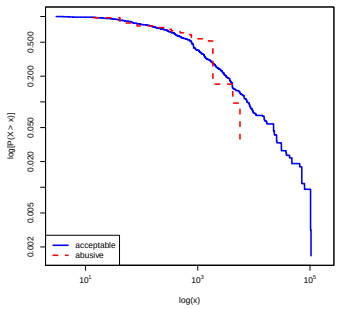


Fig. 3. CCDF of # of subscribers.

Assuming that abusive accounts do need to artificially inflate their subscriber base, one could use proof-of-work based group size estimation methods [5] to increase the cost of faking a large subscriber base. However, the network size estimation method presented in [5] would reveal the public keys of some of the subscribers. Still, this is easily mitigated by having each subscriber use a fresh pseudonym for each subscription, limiting the use of this special pseudonym to the group size estimation protocol. This has the drawback that the proof-of-work computation would have to be performed again for each subscription.

In any case, we do not expect such methods to work particularly well: an adversary can typically be expected to be willing to spend significant energy to create fake accounts. As a result, preventing fake accounts from being created by increasing the complexity is likely to deter normal users from using the system long before this would become an effective deterrent for a determined adversary.

5.7 Subscription ∩ subscription

The "subscription ∩ subscription" feature is measuring the size of the intersection among the set of subscriptions of the sender and the receiver in relation; it is normalized by dividing it by the sum of the number of subscriptions of the receiver and the sum of subscriptions of the sender. Subscriptions are likely private information, and thus neither sender nor receiver can be expected to simply provide this information in a privacy-preserving OSN set up. In our data set, the resulting number of this feature is substantially less for messages classified as abusive (Figure 4), thus an adversary would attempt to increase the value. This requires the adversary to guess which subscriptions the victim may have, and then to create (or pretend to have made) the same subscriptions. We expect this to be costly, but not computationally hard: by watching the victim's public activity, it is likely possible to deduce quite a bit of information about the victim's subscriptions.

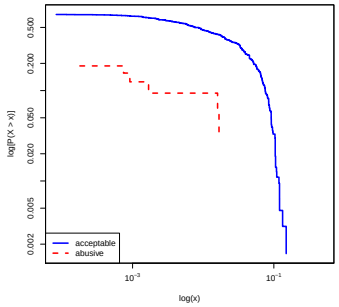


Fig. 4. CCDF of subscription intersection.

Our protocol part 1 We provide a new privacy-preserving protocol to compute the size of the set intersections, which is a variation of the PSI-CA protocol of [4]. Suppose each user has a private key c_i and the corresponding public key is $C_i := g^{c_i}$ where g is some generator. Let $\mathcal{A}$ be the set of public keys representing Alice's subscriptions and $\mathcal{B}$ be the set of keys representing Bob's subscriptions. Fix a cryptographic hash function h. For any list or set Z, define $Z' := \{h(x)|x \in Z\}$. We also assume a fixed system security parameter $\kappa \geq 1$ has been agreed upon.

Suppose Alice wishes to know $n := |\mathcal{A} \cap \mathcal{B}|$. First, she generates an ephemeral private scalar $x_A \in \mathbb{Z}/p\mathbb{Z}$ and sends Bob

$$\mathcal{X}_{\text{Alice}} := \texttt{sort}\left[C^{x_A} \mid C \in \mathcal{A}\right] \tag{1}$$

Second, Bob picks ephemeral private scalars $t_{\text{Bob},j} \in \mathbb{Z}/p\mathbb{Z}$ for $j \in 1, \ldots, \kappa$ and computes

$$\mathcal{X}_{\text{Bob},j} := \texttt{sort}\left[C^{t_{\text{Bob},j}} \;\middle|\; C \in \mathcal{B}\right] \tag{2}$$

$$\mathcal{Y}_{\text{Bob},j} := \texttt{sort}\left[\overline{C}^{t_{\text{Bob},j}} \;\middle|\; \overline{C} \in \mathcal{X}_{\text{Alice}}\right] \tag{3}$$

He then sends commitments $\mathcal{X}'_{\text{Bob},i}$ and $\mathcal{Y}'_{\text{Bob},i}$ for $i \in 1, \ldots, \kappa$ to Alice. Third, Alice picks a non-empty random $J \subseteq \{1, \ldots, \kappa\}$ and sends J to Bob. Fourth, Bob sends Alice his scalar $t_{\text{Bob},j}$ for $j \notin J$, as well as $\mathcal{X}_{\text{Bob},j}$ for $j \in J$. Fifth, Alice checks the $t_{\text{Bob},j}$ matches the commitment $\mathcal{Y}'_{\text{Bob},j}$ for $j \notin J$. She also verifies the commitment to $\mathcal{X}_{\text{Bob},j}$ for $j \in J$. She then computes for $j \in J$

$$\mathcal{Y}_{\text{Alice},j} := \left\{ \hat{C}^{x_A} \;\middle|\; \hat{C} \in \mathcal{X}_{\text{Bob},j} \right\} \tag{4}$$

Finally, Alice computes the result from $|\mathcal{Y}'_{\text{Alice},j} \cap \mathcal{Y}'_{\text{Bob},j}| = n$ for $j \in J$, checking that all $|J| \geq 1$ values agree.

We note that the same privacy-preserving protocol also applies for computing the overlap between the sender's subscriptions and the receiver's subscribers. However, in this case it is even easier for the adversary to manipulate the outcome, as the adversary can simply create fake accounts to subscribe to the victim, and it is trivial for the adversary to subscribe to these fake accounts. As a result, the adversary can increase the overlap for the "subscriber^r-subscription^s" feature limited only by the number of fake accounts. As with the "number of subscribers" (Section 5.6), this attack can again be slightly mitigated by making account creation expensive.

5.8 Subscriber ∩ subscriber

The "subscriber ∩ subscriber" feature is measuring the size of the intersection among the set of subscribers of the sender and the receiver; it is again normalized by the sum of the number of subscribers of sender and receiver. Unlike their subscription set, a user cannot freely determine the set of their subscribers: A user needs to actually convince other users that they should subscribe to their

public channel. We assume the channel owner knows its subscribers, and that the subscribers are willing to cryptographically sign a message saying that they are subscribed to the user's channel.

Given this, we create a stronger version of the protocol from Section 5.7, which uses signatures that allow Bob to prove to Alice that his input consists really of his subscribers. The tricky part here is that the identities of the subscribers are still sensitive private information, so we need to use a particular signature scheme for our privacy-preserving computation of the overlap in subscriber sets. The fact that subscribers provide the signatures and not a certification authority is a key difference to the private set intersection with certificate authority (PSI-CA) of [4].

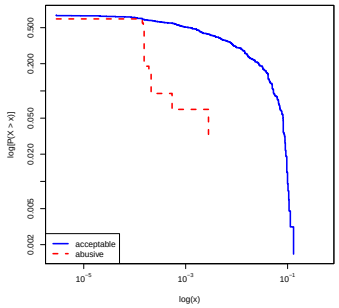


Fig. 5. CCDF of subscriber intersection.

The Boneh-Lynn-Shacham (BLS) signature scheme We first outline the BLS signature scheme [2], which begins with a Gap co-Diffie-Hellman group pair (G_1, G_2) of order p with an efficiently-computable bilinear map $e\colon G_1 \times G_2 \to G_T$, a generator g_2 of G_2, and a cryptographic hash function $H : \{0,1\}^* \to G_1$.

In the BLS scheme, a private key consists of a scalar $c \in \mathbb{Z}/p\mathbb{Z}$, while the corresponding public key is $C := g_2^c$, and a signature on a message m by C is $\sigma := H(m)^c$.

A signature σ is verified by checking that $e(H(m), C) = e(\sigma, g_2)$. If $\sigma = H(m)^c$ then this holds by bilinearity of e.

Our protocol part 2 We again define $Z' := \{h(x) | x \in Z\}$ whenever Z is some set under discussion, and assume a fixed system security parameter $\kappa \geq 1$ has been agreed upon. Each participant is identified by a public key pair $C = g_2^c$ for the BLS signature scheme. Each participant A has a subscriber list L_A consisting of tuples $(C, \sigma_{A,C})$ where $\sigma_{A,C} := H(A, \texttt{date})^c$ is a BLS signature affirming that $C = g_2^c$ was subscribed to A until some expiration `date`, the specifics of which depend on the application. We envision these signatures being provided in advance so that Bob's subscribers need not be online when running the protocol.

Suppose Alice wishes to know $n := |L_{\text{Alice}} \cap L_{\text{Bob}}|$. First, she generates an ephemeral private scalar $x_A \in \mathbb{Z}/p\mathbb{Z}$ and sends Bob

$$\mathcal{X}_{\text{Alice}} := \texttt{sort}\left[\, C^{x_A} \mid (C, \sigma_{A,C}) \in L_{\text{Alice}} \,\right] \tag{5}$$

Second, Bob picks ephemeral private scalars $t_{\text{Bob},j} \in \mathbb{Z}/p\mathbb{Z}$ for $j \in 1, \ldots, \kappa$ and computes

$$\mathcal{X}_{\text{Bob},j} := \texttt{sort}\left[(C^{t_{\text{Bob},j}}, \sigma_{B,C}^{t_{\text{Bob},j}}) \;\middle|\; (C, \sigma_{B,C}) \in L_{\text{Bob}} \right] \tag{6}$$

$$\mathcal{Y}_{\text{Bob},j} := \texttt{sort}\left[\overline{C}^{t_{\text{Bob},j}} \;\middle|\; \overline{C} \in \mathcal{X}_{\text{Alice}} \right] \tag{7}$$

He then sends commitments $\mathcal{X}'_{\text{Bob},i}$ and $\mathcal{Y}'_{\text{Bob},i}$ for $i \in 1, \ldots, \kappa$ to Alice. Third, Alice picks a non-empty random $J \subseteq \{1, \ldots, \kappa\}$ and sends J to Bob. Fourth, Bob sends Alice his scalar $t_{\text{Bob},j}$ for $j \notin J$, as well as $\mathcal{X}_{\text{Bob},j}$ for $j \in J$. Fifth, Alice checks the $t_{\text{Bob},j}$ matches the commitment $\mathcal{Y}'_{\text{Bob},j}$ for $j \notin J$. She also verifies the commitment to as well as the signatures in $\mathcal{X}_{\text{Bob},j}$ for $j \in J$. The signatures in $\mathcal{X}_{\text{Bob},j}$ validate because we employ the BLS pairing based signature scheme where:

$$\begin{aligned} e(C^{t_{\text{Bob},j}}, H(m)) &= e(C, H(m))^{t_{\text{Bob},j}} \\ &= e(P_1, \sigma_{B,C})^{t_{\text{Bob},j}} = e(P_1, \sigma_{B,C}^{t_{\text{Bob},j}}) \end{aligned}$$

Alice then computes for $j \in J$

$$\mathcal{Y}_{\text{Alice},j} := \left\{ \hat{C}^{x_A} \;\middle|\; \hat{C} \in \mathcal{X}_{\text{Bob},j} \right\} \tag{8}$$

Finally, Alice obtains the result from $|\mathcal{Y}'_{\text{Alice},j} \cap \mathcal{Y}'_{\text{Bob},j}| = n$ for $j \in J$, checking that all $|J| \geq 1$ values agree.

An attack on this blinded signature scheme translates into an attack on the underlying BLS signature scheme. If Bob tries to manipulate to increase the overlap, the cut-and-choose part detects this with probability $1 : 2^{\kappa}$.

Assessment In our data set, the size of the subscriber set intersection is again substantially lower for messages classified as abusive (Figure 5), thus an adversary would attempt to increase the value. It is hard for an adversary to try to get the subscribers of the victim to subscribe to the adversary's feed, especially given that the subscribers are typically unknown to the adversary as subscriptions are private information.

It is again possible for the adversary to create fake accounts which subscribe to both the adversary and the victim. While these accounts may be relatively new, the "age of account" feature only considers the age of the sender's account, not the age of the accounts of subscribers. As with the "subscribers count" feature, proof-of-work techniques may increase the cost of this attack.

5.9 Subscriber^s ∩ subscription^r

Finally, we consider the size of the intersection among the set of subscribers of the sender and the subscriptions of the receiver. Figure 6 shows that, an adversary would try to increase the intersection of their subscribers (subscriber^s) with the subscriptions of the receiving victim (subscription^r). This feature is particularly interesting, as the sending attacker cannot easily influence set of subscriptions of the receiver, and will similarly have a hard time obtaining subscriptions from the user's to whom the victim is subscribed to. Unlike "subscriber ∩ subscriber", creating fake accounts is ineffective unless the receiver subscribes to these fake accounts.

Naturally, computing the subscriber^s-subscription^r overlap is again dependent on privacy-sensitive information. However, the protocol from the previous section can be trivially adapted to the situation where Alice uses her set of subscriptions instead of her set of subscribers.

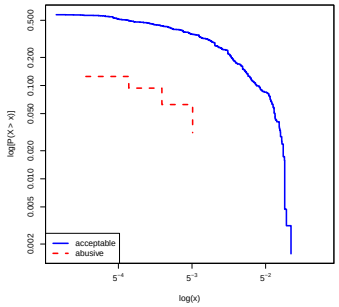


Fig. 6. CCDF of subscriber^s-subscription^r intersection.

6 Evaluation

We have shown how to obtain some of the key features from our original abuse detection heuristic even in a privacy-preserving decentralised OSN. While many of the features can be inherently manipulated by a sophisticated adversary, others can be made robust even against strong and adaptive attacks.

We now evaluate the abuse detection system in the context of an adaptive adversary. In particular, we assume that the adversary can *trivially* adapt all of the account properties of the sender's account, *possibly* create fake accounts

Feature	Falsification/Adaptation	Crypto helps?
# lists	trivial	n/a
# subscriptions	trivial	n/a
$\frac{\#\text{ subscriptions}}{\text{age}}$	trivial	n/a
$\frac{\#\text{subscriptions}}{\#\text{subscribers}}$	trivial	n/a
# mentions	costly	n/a
# hashtags	costly	n/a
$\frac{\#\text{ messages}}{\text{age}}$	costly	yes
$\frac{\#\text{ mentions}}{\text{age}}$	costly	yes
$\frac{\#\text{ mentions}}{\#\text{ messages}}$	costly	n/a
# retweets	costly	n/a
# favorited messages	costly	n/a
age of account	**hard**	yes
# subscribers	possible	minimally
$\frac{\#\text{ subscribers}}{\text{age}}$	possible	minimally
subscription $\cap$ subscription	costly	w. privacy
subscriber $\cap$ subscriber	possible	w. privacy
subscriber^s $\cap$ subscription^r	**very hard**	yes
subscription^s $\cup$ subscriber^r	possible	w. privacy
message invasive	**hard**	n/a

Table 5. Summary of how difficult it would be for an adversary to manipulate features.

Classifier	Metric	Arithmetic Mean	Geometric Mean	Only Acceptable	Only Abusive
DT	Precision	0.64 ± 0.09	0.54 ± 0.04	0.98 ± 0.01	0.30 ± 0.17
	Recall	0.78 ± 0.12	0.76 ± 0.14	0.91 ± 0.08	0.64 ± 0.26
	F-score	0.67 ± 0.11	0.62 ± 0.09	0.95 ± 0.05	0.40 ± 0.18
RF	Precision	0.67 ± 0.12	0.59 ± 0.05	0.98 ± 0.01	0.36 ± 0.24
	Recall	0.76 ± 0.08	0.74 ± 0.09	0.94 ± 0.09	0.58 ± 0.19
	F-score	0.69 ± 0.12	0.64 ± 0.10	0.96 ± 0.05	**0.43** ± 0.20
ET	Precision	0.58 ± 0.05	0.40 ± 0.04	**0.99** ± 0.02	0.16 ± 0.08
	Recall	**0.80** ± 0.17	0.79 ± 0.16	0.79 ± 0.08	**0.80** ± 0.33
	F-score	0.58 ± 0.08	0.49 ± 0.08	0.88 ± 0.05	0.27 ± 0.13
GB	Precision	**0.71** ± 0.10	0.66 ± 0.04	0.97 ± 0.01	**0.45** ± 0.20
	Recall	0.70 ± 0.07	0.64 ± 0.07	**0.97** ± 0.03	0.42 ± 0.15
	F-score	**0.70** ± 0.08	0.64 ± 0.05	**0.97** ± 0.02	0.42 ± 0.14

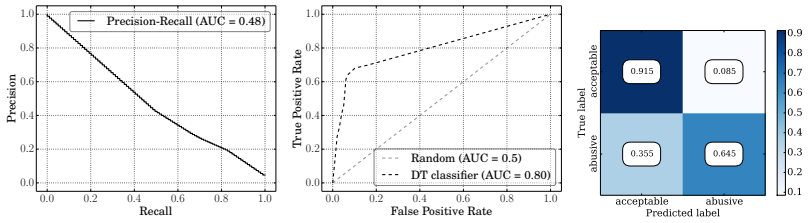
Table 6. Classifiers trained with 5-fold cross validation and hard to forge features

(Sybils) and fake subscriptions, and is willing to make *costly* behavioral adaptations, e.g. by adapting the text of messages to avoid message properties as mentions' 5.2 and the frequency at which messages of any type are sent (Table 5). However, the adversary is unable to manipulate the age of accounts (by breaking the timeline service) or to break the cryptographic primitives used in the protocols presented in this paper.

Given this adversary model, only three features remain: the age of the account, the $\text{subscriber}^r \cap \text{subscription}^s$ intersection size, and the invasive predicate. All other features need to be excluded from the classification algorithm's inputs, as we have to assume that the adversary will adapt to provide the worst-case input, thereby making abusive messages seem more benign.

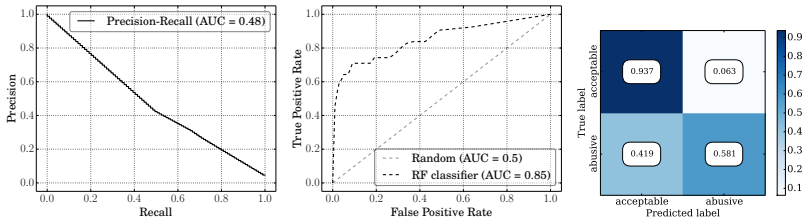
We evaluated the accuracy of the supervised learning techniques presented in Section 4 on this modified feature set. Table 6 summarizes the results for the various classifiers. As before, the ET and GB classifiers generally perform better than DT and RF for our data set; however, the high variance means that this comparison may not generalize. The reduced feature set largely impacts the precision for abusive messages, cutting it by a bit more than a third in the best case scenario, and more than two-thirds in a worst case one (e.g., DT). Still, even with this strong adaptive adversary, the GB classifier performs at slightly more than half the precision and nearly the same recall of a human reviewer for abusive messages.

Figures 7 to 10 provide the ROC curve, precision-recall (P-R) curves and the confusion matrix (CM). In terms of relative importance (RI), the age of account has always the highest importance (DT: 0.64%, RF: 0.59%, ET: 0.44%, GB: 0.80%) and the invasive predicate ranks pretty low in importance (DT: 0.00%, RF: 0.07%, ET: 0.27%, GB: 0.01%).



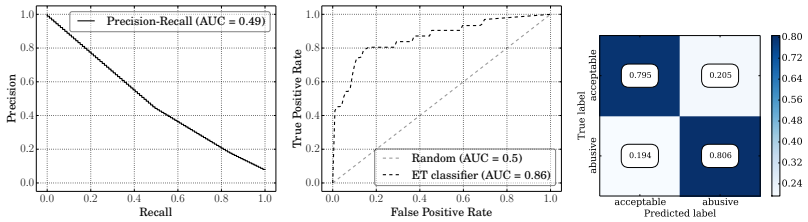
(a) P-R curve (b) ROC curve (c) CM

Fig. 7. Evaluation for decision trees (with strong adaptive adversary)



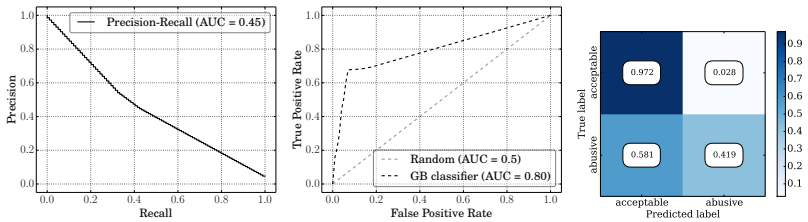
(a) P-R curve (b) ROC curve (c) CM

Fig. 8. Evaluation for random forest (with strong adaptive adversary)



(a) P-R curve (b) ROC curve (c) CM

Fig. 9. Evaluation for extra trees (with strong adaptive adversary)



(a) P-R curve (b) ROC curve (c) CM

Fig. 10. Evaluation for gradient boosting (with strong adaptive adversary)

7 Discussion

Many of the features we originally considered could not be effectively secured against an adversary creating fake accounts and fake subscriptions. It might be possible to use some of these features if we additionally considered the age of the accounts: given a time-stamping service, the adversary may be able to create fake accounts, but it would be very hard to back-date them. Combining timestamped public keys with the privacy-preserving set intersection protocols is thus an interesting open problem for future work.

That said, even if we included some of these features that could be secured, the performance of the privacy-preserving classifiers did not significantly improve. The more substantial gains seem to depend on features involving basic account properties and sender behavior which fundamentally cannot be secured against an adaptive adversary as they are under full control of the adversary. Real-world deployments will thus have to figure out whether including those features would help (because real-world adversaries are not that adaptive) or hurt (because real-world adversaries would adapt to use these features to their advantage).

We envision that future decentralised privacy-preserving OSNs will use the sort of abuse classifiers discussed here as part of ranking messages in the user's *timeline*, not for binary filtering of messages for an inbox. By *timeline*, we mean any interface that displays short message summaries ordered so that users never feel the desire to read all listed messages. After browsing only a brief portion of their timeline, a user should firstly feel they have skimmed enough summaries to be up to date on any topics about which they consult the application, and secondly not have spent time on matters they might later regret, such as responding to abusive messages.

We have treated abuse as a binary classification problem in this article, but actually one would prefer the different features to report back a numerical risk score for timeline construction. As a result, the concerns around bias one encounters with binary classifiers [12] seem unnecessary here. Instead, actual timeline constructions requires integrating an array of features with both positive and negative aspects.

In terms of concrete deployments, we envision that future OSNs would include a decision tree baked into the code and not expect users to train their own classifier. This will simplify the deployed software, improve usability and avoid users running expensive training algorithms.

8 Conclusion

Our results show how to combine local knowledge with private set intersection and union cardinality protocols (with masking of BLS signature to protect identity of signers/subscribers) to privately derive feature values from users in OSNs. Given an adaptive adversary that would be able to manipulate most features we propose in our supervised learning approach, it is surprising that with just three

features resistant to adversarial manipulation, the algorithms still provide useful classifications.

Acknowledgments We thank the Renewable Freedom Foundation for supporting this research, the volunteers who annotated abuse and the anonymous reviewers. Special thanks to Cristina Onete for pointing us towards PSI protocol literature.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. 1613-9011, Springer-Verlag New York, 1 edn. (2006)
2. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology. pp. 514–532. ASIACRYPT '01, Springer-Verlag, London, UK, UK (2001)
3. Breiman, L.: Arcing the edge. Tech. rep., Technical Report 486, Statistics Department, University of California at Berkeley (1997)
4. De Cristofaro, E., Gasti, P., Tsudik, G.: Fast and private computation of cardinality of set intersection and union. In: Cryptology and Network Security: 11th International Conference, CANS. pp. 218–231. Springer, Berlin, Heidelberg (2012)
5. Evans, N.S., Polot, B., Grothoff, C.: Efficient and secure decentralized network size estimation. IFIP International Conferences on Networking (2012)
6. Gipp, B., Meuschke, N., Gernandt, A.: Decentralized trusted timestamping using the crypto currency bitcoin. In: iConference. iSchools (2015)
7. Grothoff, C., Porup, J.M.: The NSA's SKYNET program may be killing thousands of innocent people. ARS Technica UK (2016), `https://hal.inria.fr/hal-01278193`
8. Hinduja, S., Patchin, J.W.: Bullying, cyberbullying and suicide. Archives of Suicide Research 14(3) (2010)
9. Kramer, A., Guillory, J., Hancock, J.: Experimental evidence of massive-scale emotional contagion through social networks. Proceedings of the National Academy of Sciences of the United States of America (2013)
10. Langos, C.: Cyberbullying: The challenge to define. Cyberpsychology, Behavior, and Social Networking 15, 285–289 (2012)
11. v. Loesch, C., Toth, G.X., Baumann, M.: Scalability & paranoia in a decentralized social network. In: Federated Social Web. Berlin, Germany (2011)
12. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences 250, 113–141 (2013)
13. Luxton, D., June, J., Fairall, J.: Social media and suicide: A public health perspective. American Journal of Public Health 102, 195–200 (May 2012)
14. Mandeep K. Dhami, P.: Behavioural Science Support for JTRIG's Effects and Online HUMINT Operations (March 2011), `http://www.statewatch.org/news/2015/jun/behavioural-science-support-for-jtrigs-effects.pdf`
15. Stein, T., Chen, E., Mangla, K.: Facebook immune system. In: Proceedings of the 4th Workshop on Social Network Systems. p. 8. ACM (2011)
16. Thomas, K., McCoy, D., Grier, C., Kolcz, A., Paxson, V.: Trafficking fraudulent accounts: the role of the underground market in twitter spam and abuse. In: USENIX Security Symposium (2013)