Preemptive Detection of Fake Accounts on Social Networks via Multi-Class Preferential Attachment Classifiers

Adam Breuer
abreuer@dartmouth.edu

Nazanin Khosravani
nazaninkt@fb.com

Michael Tingley
tingley@fb.com

Bradford Cottel
bcottel@fb.com

ABSTRACT

In this paper, we describe a new algorithm called **Pre**ferential **Attac**hment **k**-class Classifier (PreAttacK) for detecting fake accounts in a social network. Recently, several algorithms have obtained high accuracy on this problem. However, they have done so by relying on information about fake accounts' friendships or the content they share with others—the very things we seek to prevent.

PreAttacK represents a significant departure from these approaches. We provide some of the first detailed distributional analyses of how new fake (and real) accounts first attempt to make friends by strategically targeting their initial friend *requests* after joining a major social network (Facebook). We show that even before a new account has made friends or shared content, these initial friend *request* behaviors evoke a natural multi-class extension of the canonical Preferential Attachment model of social network growth.

We leverage this model to derive a new algorithm, PreAttacK. We prove that in relevant problem instances, PreAttacK near-optimally approximates the posterior probability that a new account is fake under this multi-class Preferential Attachment model of new accounts' (not-yet-answered) friend requests. These are the first provable guarantees for fake account detection that apply to new users, and that do not require strong homophily assumptions.

This principled approach also makes PreAttacK the only algorithm with provable guarantees that obtains state-of-the-art performance at scale on the global Facebook network, allowing it to detect fake accounts before standard methods apply and at lower computational cost. Specifically, PreAttacK converges to informative classifications (AUC ≈0.9) after new accounts send + receive a total of just 20 not-yet-answered friend requests. For comparison, state-of-the-art network-based algorithms do not obtain this performance even after observing additional data on new users' first 100 friend requests. Thus, unlike mainstream algorithms, PreAttacK *converges before the median new fake account has made a single friendship (i.e.* accepted *friend request)* with a human.

KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599471

CCS CONCEPTS

• **Computing methodologies → Machine learning algorithms**; • **Networks → Online social networks**; • **Security and privacy**;

KEYWORDS

Social network analysis and graph algorithms; security, privacy, and trust; fake accounts & fake news; preferential attachment; sybils.

ACM Reference Format:
Adam Breuer, Nazanin Khosravani, Michael Tingley, and Bradford Cottel. 2023. Preemptive Detection of Fake Accounts on Social Networks via Multi-Class Preferential Attachment Classifiers. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599471

1 INTRODUCTION

Fake user accounts are the primary source of fake news and other malicious phenomena on social networks such as Facebook and Twitter. Organized campaigns of fake accounts have recently been used to influence public opinion, push propaganda, infiltrate political discourse, manipulate stock markets, steal personal data, and propagate scams [7, 13, 14, 16–18, 26, 34, 35, 37, 38, 40]. Detecting these fake accounts and limiting their ability to interact maliciously with humans are core tasks for modern social networks [11, 23, 47].

The scale of fake accounts has increased commensurately with the rapid growth of online social networks. In the last year alone, Facebook disabled 6.1 billion fake accounts—more than double the number of active users on the Facebook network [16]. This figure reflects immense recent progress in fake account classification—for example, Facebook disabled the vast majority of these fakes during account registration. Nonetheless, the fraction of *active* social network users who are fake has remained at roughly 4-5% (for Facebook) or 8-15% (for Twitter) for the last several years [16, 40].

The early detection paradox. These active fakes that evade registration-time classifiers and join a social network raise what we call the early detection paradox: *Mainstream algorithms to detect active fake accounts rely on information about their friends or the content they share with others, yet these friendships and shared content are the very things we seek to prevent.* Our goal in this paper is to design algorithms that overcome this paradox by classifying active fake accounts *before* they make friends or share content.

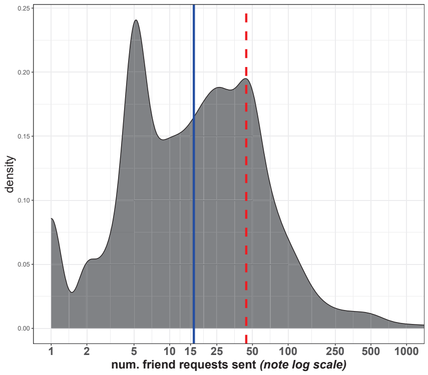

Figure 1: Distribs. of counts of Facebook friend requests *sent* by new fake accounts before a single real user accepts any *among new fake accounts who eventually befriend a real user.* Median at blue solid line; mean at dashed red line.

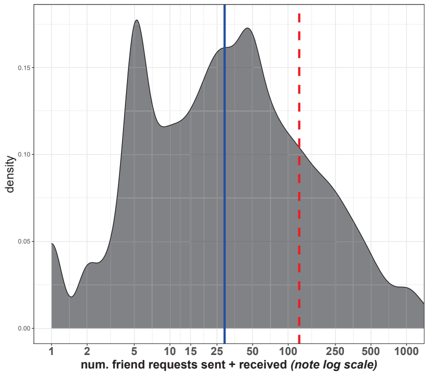

Figure 2: Distribs. of counts of friend requests *sent + received.*

Recent algorithms. This paradox is captured by the two mainstream approaches to fake account detection:

- *Network-structural algorithms.* Network-structural algorithms classify long-tenured accounts via the *Homophily Assumption*, which states that users *eventually* tend to cluster together (i.e. make the majority of their steady-state friendships) with other users who share their same $\{fake, real\}$ label [20, 44, 49, 53]. Based on this assumption, network-structural algorithms attempt to propagate a small number of known users' $\{fake, real\}$ labels across the friendship network to unknown users via either Random Walks [9, 15, 21, 49, 52, 53] or Belief Propagation [19, 20, 43–45].
- *Feature-based classifiers.* Recently, a variety of research has detected fake accounts in a supervised learning setting. State-of-the-art algorithms such as DEC [47], Jodie [25] and Ties [30] accomplish this via embeddings of tens of thousands of features that capture sophisticated properties of a user's friendship network, such as the average account age of a user's friends-of-friends, or temporal trends in the content a user shares over time [22, 24, 25, 30, 36, 41, 47]. While these algorithms have no theoretic guarantees, they are performant: Facebook now uses them to obtain high quality $\{fake, real\}$ labels (AUC>0.98) for virtually all of their long-tenured users [11, 23, 47].

Notwithstanding these impressive results, neither approach is ideally suited to the *early detection* of new fake accounts that have not yet made many (or any) friendships: Because such accounts have just passed registration-time feature-based classifiers, they cannot be detected by other feature-based classifiers until their features evolve significantly. Also, many informative features are unknown until after a new user has made several friends or shared content with others. Similarly, it is well-known that mainstream network-structural algorithms do not apply, as their theoretic guarantees rely critically on the Homophily Assumption, which only applies to long-tenured users who have had sufficient 'stabilization time' to make the majority of their eventual friendships [1, 9, 33, 44, 49]. For this reason, evaluations of network-structural algorithms have often excluded new users with less than e.g. 1 to 6 months of tenure on the social network [9, 10, 49]. Recent evaluations of these algorithms on the Facebook network suggest they perform poorly (AUC<0.6) on new users who have not yet made many friends [11].

Overcoming the paradox. To address this early detection paradox, we use data from the Facebook social network to provide some of the first distributional analyses of how fake (and real) accounts target their friend *requests* after joining a major social network (Figs. 1-4). This focus on friend requests is motivated by the fact that new fake accounts can only meaningfully interact with real users after they have sent friend requests to real users (or received requests from real users) *and* those requests have been seen and accepted. Fig. 1 shows that *among the subset of new fake accounts that eventually obtain a friendship with a real user,* the median new fake account sends 16 friend requests before obtaining a single friendship (*accepted* request) with a real user (note log-scale). If we also include the requests these new fake accounts receive from others (Fig. 2), the count increases to 29 requests (sent+received).

Can we leverage this small number of not-yet-answered friend requests to distinguish new fake accounts from new real users?

k-Class Directed Preferential Attachment model (kCDPA). On the Facebook network, we observe that while fake and real users do differ slightly as a class in terms of the degree to which they send and receive requests from fakes and reals (*red vs. blue distributions in Figs. 3 and 4, next pg.—see also Sec. 5*), these class-level differences are small in comparison to individual-level differences (*spread of distributions*). Specifically, some users are exponentially more likely to request (or be requested by) a real user (*mass right of red lines in Figs. 3 and 4, resp.*); Other users are exponentially more likely to request (or be requested by) a fake account (*mass left of red lines*).

This observation evokes the canonical Preferential Attachment (PA, i.e. rich-get-richer) generative model of social network growth [2, 5, 6, 8, 32]. In a traditional PA model, each new user joins a social network and sends friend requests to recipients who are selected with probability proportional to the counts of requests that they have already received. This process results in a power-law distribution of users' in-degrees such that a small number of recipients become vastly more popular than others. PA models and their associated dynamic processes continue to motivate a variety of recent results across several machine learning subfields.

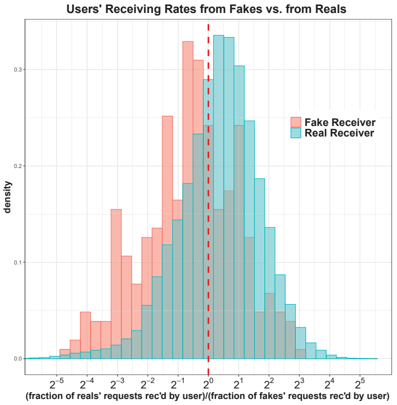

Figure 3: Mass right of x=1 line represents $\{fake, real\}$ Facebook users who *receive* disproportionately more of real users' requests vs. fakes' requests by the factor on the x axis.

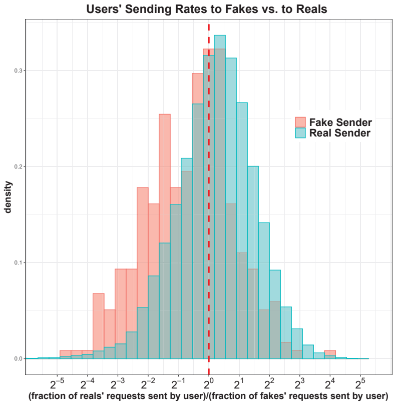

Figure 4: Mass right of x=1 line represents FB users who *send* disproportionately more requests to real users vs. to fakes.

In our problem setting, fake and real users' 'preferential attachment' to *different* individuals inspires a natural multi-class extension of the PA model, which we call kCDPA:

- Suppose we observe an *arbitrary preexisting* directed network of friend requests between existing users. Then, suppose some new fake and real users join this network.
- New fakes and reals each send and receive friend requests to/from existing users who are chosen proportional to how many *of the new user's $fake/real$ class* already did so.

The kCDPA model provides a principled foundation for a classifier that applies to new accounts. Specifically, recent research has highlighted various similar multi-class PA models as a theoretical mechanism for the emergence of homophily in social networks [3, 4, 27, 29, 54]. As such, our kCDPA model forms a natural antecedent to standard homophily-based fake account detection methods that are used to detect long-tenured fake accounts.

We emphasize that we use kCDPA to model the friend request networks of a small batch of new users; we do *not* assume that the entire network emerged from this process (which would be a far stronger assumption), nor do we assume that the (distinct) network of *accepted* friend requests (i.e. friendships) adheres to PA.

Main contribution. Our main result is an algorithm, PreAttack, that determines the posterior probability that a new user is a fake account based on the kCDPA model of her (not-yet-answered) friend requests. Specifically, PreAttack updates the probability that a new user is fake to the extent she (1) 'preferentially attaches' to specific recipients in keeping with their probabilities of being requested by fake accounts vs. by real ones, and also (2) to the extent existing users 'preferentially attach' to her in keeping with their probabilities of sending requests to fake accounts vs. real ones.

- **Theoretic contribution.** We derive instance-specific bounds that show PreAttack near-optimally approximates each new user's posterior probability of being fake in relevant problem instances at lower computational cost than alternatives. These are the first provable guarantees for fake account detection that apply to new users, and that do not require strong homophily assumptions. Indeed, despite the enormous popularity of Preferential Attachment models, to our knowledge PreAttack is the first time that the corresponding classifier has been derived.
- **Real-world effectiveness.** This principled approach makes PreAttack the only algorithm with provable guarantees that obtains state-of-the-art performance at scale on the global Facebook network. Specifically, we implement PreAttack at scale at Facebook and show it obtains high AUC≈0.9 after new users sent/received a total of just 20 not-yet-answered friend requests. For comparison, state-of-the-art network-based algorithms do not obtain this performance even after observing additional data on new users' first 100 friend requests. This means that unlike existing algorithms, PreAttack converges to detect fakes before the median new fake account makes a single friendship (i.e. *accepted* request) with a real user (see Figs. 1 and 2 on the previous page).
- **General applicability.** While we focus on fake accounts on Facebook, PreAttack applies generally to networks where directed edges convey information about users' latent labels, such as Twitter or Instagram 'follows', LinkedIn 'connects', etc. PreAttack may also be used to infer new users' other latent class labels beyond fake/real (e.g. political party, etc.), which offers a means to address other cold-start problems.

Paper organization. Section 2 specifies the kCDPA model. Sections 3 and 4 derive PreAttack and its instance-specific approximation bounds. Section 5 extends PreAttack by incorporating observed homophily to obtain faster convergence. Section 6 shows PreAttack's performance on the Facebook network.

2 MULTI-CLASS DIRECTED PA (kCDPA)

Our core generative model is a simple but powerful extension of the canonical directed Preferential Attachment (*rich-get-richer*) model to the setting where there are k=2 classes of new users, *fakes* and *reals*, who join a preexisting social network. Whereas traditional PA models capture how new users tend to seek out already-popular users, kCDPA captures how new users tend to seek out (request and/or be requested by) users who are already popular with those of the new user's fake/real class:

- **New users' outgoing friend requests:** We model that new *fake* users send friend requests to existing users drawn in proportion to the counts of requests existing users already received *from fakes only*, and new *real* users send friend requests to existing users drawn in proportion to the counts of requests existing users already received *from reals only*.
- **New users' incoming friend requests:** Similarly, each new *fake* [or *real*] user *receives* requests from existing users who are drawn in proportion to the counts of requests existing users already sent *to fakes* [or *reals*].

The kCDPA model is formally described by the following generative process. Suppose we have a preexisting directed social network $G(V, E_0, L_V)$ where edges E_0 capture friend *requests* (*not* friendships/accepted requests). We consider k=2 classes: users $v \in V$ have known fake/real labels $L_V : L_V \in \{F, R\}^{|V|}$. We will denote a single user v's label by lowercase ℓ_v. Finally, we have a small set of *new users* $U : U = \{u_1 \dots u_m\}$ who are each fake with probability π. Some new users u are more likely than others to send a friend request and/or receive a friend request. To be as general as possible, suppose we have some distribution $\mathcal{D}$ that captures these probabilities (so $\mathcal{D}$'s domain includes $2|U|$ entries—two for the probability that each new user u will $[send, receive]$ a friend request). The kCDPA model is then:

k-Class Directed Preferential Attachment (kCDPA)

```
input Preexisting network of requests $G(V, E_0, L_V)$;  new users $U$
  Draw new users' fake/real labels $L_U = \{\ell_u \sim Bernoulli(\pi)\}$
  for $i \in 1, \dots, n$
    Draw new user & direction $\{u \in U,\ d \in [send, receive]\} \sim \mathcal{D}$
    if $d = send$
      Draw $v \in V;\ P(v) \propto \alpha + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[(y = v) \wedge (\ell_x = \ell_u)]$
      $E_i = E_{i-1} \cup \{u \longrightarrow v\}$
    else
      Draw $v \in V;\ P(v) \propto \alpha + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[(x = v) \wedge (\ell_y = \ell_u)]$
      $E_i = E_{i-1} \cup \{v \longrightarrow u\}$
  return $G(V \cup U,\ E_i,\ L_V \cup L_U)$
```

Here, α is a small constant that captures e.g. the probability that a preexisting user v receives or sends her first-ever request (in Section 5 below, we consider a 'homophily-incorporating' extension where α depends on the sender and receivers' real/fake labels). By $\mathbb{1}$ we denote the indicator function that takes value 1 if the argument is true and 0 otherwise, so the sum under the *if* $d = send$ statement counts the number of friend requests that existing user v has already received from users who have the same $\{fake, real\}$ label as new user u. Note that this includes requests from the preexisting network (E_0) as well as requests from new users in previous iterations.[1]

While we are interested in k=2 classes, note that kCDPA easily extends to the case where there are k>2 classes of users, $L_V \in \{1, \dots, k\}^{|V|}$ just by replacing *Bernoulli(π)* with *Multinom*$(\pi_1, \dots, \pi_k)$. This captures (for example) settings where there are multiple types of fake users: sockpuppets, false news bots [42], etc., and each has different preferences in terms of existing users they seek to befriend.

Very recently, similar 2-class (and multi-class) PA models have received much attention due to their ability to explain the generative process by which homophily and related properties emerge in social networks [3, 4, 27, 29, 54]. However, for our purposes, we do not require the model to explain the full evolution of a social network; we merely require it to capture the friend request behavior of new users who join a long-established network (e.g. Facebook).

3 THE PREATTACK ALGORITHM

In this section, we derive a new algorithm, **Pre**ferential **Attach**ment **k**-class Classifier (PreAttacK) that near-optimally approximates the posterior probability that each new social network user is a fake account under the kCDPA model. Intuitively, PreAttacK updates the probability that a new user is fake to the extent she (1) 'preferentially attached' to specific recipients in keeping with their probabilities of being requested by fake accounts vs. by reals, and also (2) to the extent existing users 'preferentially attached' to her in keeping with their probabilities of sending requests to fake accounts vs. reals. Because kCDPA models friend requests rather than friendships (accepted requests), PreAttacK can classify new users even before they make a single friendship. Surprisingly, despite the complex properties of kCDPA (and PA processes in general), we show that PreAttacK is also computationally efficient on mature social networks containing billions of users.

PreAttacK considerations. We are interested in the k=2 case where users are $\{real, fake\}$, but also show in Appendix B that PreAttacK also accommodates k>2 to classify multiple types of fakes, such as sockpuppets and false news bots. Importantly, we will assume that the count of requests that each new user u sends and receives are independent of her label. This precludes the undesirable scenario where PreAttacK e.g. penalizes new real users who send many requests by increasing the posterior probability that they are fake. Finally, note that kCDPA generates no requests between new accounts. It is easy to modify kCDPA to generate such requests[2], but excluding them precludes a scenario where the posterior probability that one new account is fake depends *only* on other new accounts. This prevents malicious adversaries from manipulating PreAttacK by generating many new accounts at once (see Section 4.3).

PreAttacK part I: A new user's outgoing friend requests. The conditional probability $P^F_{v_i+}$ that new *fake* user u who *sends* a friend request at iteration i of kCDPA draws preexisting user v for the recipient is proportional to the count of requests that v already received from fakes before iteration i:

$$P^F_{v_i+} := P[v_i | \ell_\mathbf{u} = \mathbf{F},\ E_{i-1}, u, d, L_V \cup L_U] \tag{1}$$

$$= \frac{\alpha + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[y = v \wedge \ell_x = F]}{\alpha|V| + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[\ell_x = F]} \tag{2}$$

Similarly, if new user u is real, this probability becomes:

$$P^R_{v_i+} := P[v_i | \ell_u = R, \cdot] = \frac{\alpha + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[y = v \wedge \ell_x = R]}{\alpha|V| + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[\ell_x = R]} \tag{3}$$

Given all new users' $\{F, R\}$ labels and the sequence of all other new users' friend requests $E_1, \dots$, then the joint conditional probability of observing u's *sequence* of outgoing friend request recipients

[1] As in the original PA model, this generative process may result in a multigraph (e.g. if the same edge is drawn twice). This is suitable for our setting, as social network users can send multiple friend requests to the same recipient (e.g. if the first is rejected).

[2] To make this change, add a line: $V = V \cup u$ at the end of the for-loop.

v_i is just the product of their individual probabilities (eqn. 2 or 3). Denote this sequence of u's recipients by $\mathcal{N}_u^+$. If u is fake:

$$P^F_{\mathcal{N}_u^+} := P[\mathcal{N}_u^+ | \ell_u = F, \cdot] = \prod_{v_i : \{u \to v\}_i \in \mathcal{N}_u^+} P^F_{v_i +} \tag{4}$$

And similarly, if u is real, this conditional probability is:

$$P^R_{\mathcal{N}_u^+} := P[\mathcal{N}_u^+ | \ell_u = R, \cdot] = \prod_{v_i : \{u \to v\}_i \in \mathcal{N}_u^+} P^R_{v_i +} \tag{5}$$

PreAttacK part II: A new user's incoming requests. Noting the symmetry of the kCDPA model with respect to requests that new users send and receive, we can also derive the cond. probability $P^F_{v_i -}$ that a new user u who *receives* a friend request at iteration i draws preexisting user v for the request's sender. Similar to above, this probability is proportional to the count of requests that v has already sent to users who share the same label as u. If u is fake:

$$P^F_{v_i -} := P[v_i | \ell_u = F, \cdot] = \frac{\alpha + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[x = v \wedge \ell_y = F]}{\alpha |V| + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[\ell_y = F]} \tag{6}$$

And if new user u is real, this conditional probability is:

$$P^R_{v_i -} := P[v_i | \ell_u = R, \cdot] = \frac{\alpha + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[x = v \wedge \ell_y = R]}{\alpha |V| + \sum_{e_{x \to y} \in E_{i-1}} \mathbb{1}[\ell_y = R]} \tag{7}$$

Similar to above, the joint conditional probability of u's sequence of incoming friend request senders (denoted by $\mathcal{N}_u^-$) if u is fake is:

$$P^F_{\mathcal{N}_u^-} := P[\mathcal{N}_u^- | \ell_u = F, \cdot] = \prod_{v_i : \{v \to u\}_i \in \mathcal{N}_u^-} P^F_{v_i -} \tag{8}$$

And similarly, if u is real, this conditional probability is:

$$P^R_{\mathcal{N}_u^-} := P[\mathcal{N}_u^- | \ell_u = R, \cdot] = \prod_{v_i : \{v \to u\}_i \in \mathcal{N}_u^-} P^R_{v_i -} \tag{9}$$

Posterior probability that a new user is fake. We are now able to derive the full posterior probability that new user u is fake as a function of the observed sequence of preexisting users to whom she sent friend requests and from whom she received requests. Leveraging Bayes' rule and the law of total probability we have:

$$\mathbf{P}^*_\mathbf{u} : = P[\ell_u = F | \mathcal{N}_u^+, \mathcal{N}_u^-, E_0, E_1 \dots E_n, L_V \cup L_{U \setminus u}] \tag{10}$$

$$= \frac{P^F_{\mathcal{N}_u^+} \cdot P^F_{\mathcal{N}_u^-} \cdot \pi}{P^F_{\mathcal{N}_u^+} \cdot P^F_{\mathcal{N}_u^-} \cdot \pi \ + \ P^R_{\mathcal{N}_u^+} \cdot P^R_{\mathcal{N}_u^-} \cdot (1 - \pi)} \tag{11}$$

$$= \Big(1 \ + \ (P^F_{\mathcal{N}_u^+} \cdot P^F_{\mathcal{N}_u^-})^{-1} (P^R_{\mathcal{N}_u^+} \cdot P^R_{\mathcal{N}_u^-}) \cdot \pi^{-1} (1 - \pi)\Big)^{-1} \tag{12}$$

This posterior captures the idea that u is relatively more likely to be fake to the extent she 'preferentially' sent requests to recipients who are more preferred by fakes, and also to the extent she received requests from senders who are more likely to send to fakes.

3.1 Intractability

Unfortunately, this expression for the posterior probability $\mathbf{P}^*_\mathbf{u}$ that a new user u is fake is intractable, as it requires knowledge of the (latent) real/fake label of all new users who sent requests before u. Moreover, computing this posterior in expectation becomes infeasible as we consider more than a handful of new users, as this requires integrating over all possible label combinations.

A standard approach at this point would be to apply either linearized belief propagation or MCMC techniques. However, both are computationally expensive in large networks due to the need to e.g. iterate between inferring new users' posterior labels and updating *all* existing users' sending and receiving preferential attachment weights (i.e. sums within $P^F_{v_i +}, P^R_{v_i +}, P^F_{v_i -}, P^R_{v_i -}$) until (possible) convergence. They also typically lack convergence guarantees [20, 51], or obtain guarantees only at the expense of the approximation (e.g. via linearization) [43, 45] or significant complexity [50].

3.2 Fast approximation

In contrast to these approaches, we consider a fast approximation for $\mathbf{P}^*_\mathbf{u}$ based on the following idea: PA probabilities in mature social networks are stable over small batches of new entrants. *So, rather than account for small and intractable changes to one new user's posterior that accrue due to other new users' edges $E_1, \dots,$ we ignore them and then bound their worst-case impact.* Consider that given a large preexisting network, a small batch of new accounts who send and receive friend requests (probably) do *not* significantly change existing users' PA probabilities (i.e. sums in the *Draw* steps of kCDPA). At a high level, there are three reasons why this is so:

1. **Collisions are (probably) rare.** Given a large preexisting network of 300 million (Twitter) or 2 billion (Facebook) users, a small batch of new users are unlikely to 'draw' the same recipients multiple times. When a new user sends a request to a recipient who was not previously requested by a new user, the numerators in $P^F_{v_i +}$ and $P^R_{v_i +}$ are equal to their (known) original values in E_0. The same is true of numerators in $P^F_{v_i -}$ and $P^R_{v_i -}$ when a new user receives a request from a not-previously-drawn sender.
2. **Collisions (probably) have negligible impact.** In cases where multiple new accounts *do* send friend requests to the same preexisting recipient, that recipient was probably already very popular (i.e. already had a large PA probability) due to PA's 'rich-get-richer' dynamics. In that case, this preexisting recipient's PA probability only undergoes a small percentage change after each new request, so it is well-approximated by its original value in E_0. This argument also applies when multiple new accounts *receive* requests from the same preexisting recipient.
3. **New users have a small number of friend requests.** A large preexisting social network of billions of users results from on the order of 10^{11} friend requests. The new requests sent by a relatively small batch of new fake and real accounts has only a negligible impact on this preexisting count. Therefore, the denominators in $P^F_{v_i +}, P^R_{v_i +}, P^F_{v_i -}, P^R_{v_i -}$ are well-approximated by their original values in E_0.

These three key intuitions, which we formalize in Section 4, suggest we can obtain a good approximation for the posterior $\mathbf{P}^*_\mathbf{u}$ by holding all PA probabilities fixed at their values in the preexisting requests network $G(V, \mathbf{E_0}, L_V)$. With this change, we can approximate the probability of observing the i'th request that a new (fake or real) account sends or receives, $P^F_{v_i+}, P^R_{v_i+}, P^F_{v_i-}$, and $P^R_{v_i-}$, *without* knowing the labels of other new accounts. For example, for the 'sending' probabilities $P^F_{v_i+}$ and $P^R_{v_i+}$:

$$P^F_{v_i+} = P[v_i | \ell_u = F, \cdot] \tag{13}$$

$$\approx \hat{P}^F_{v+} := \frac{\alpha + \sum_{e_{x \to y} \in \mathbf{E_0}} \mathbb{1}[y = v \wedge \ell_x = F]}{\alpha |V| + \sum_{e_{x \to y} \in \mathbf{E_0}} \mathbb{1}[\ell_x = F]} \tag{14}$$

And similarly:

$$P^R_{v_i+} = P[v_i | \ell_u = R, \cdot] \tag{15}$$

$$\approx \hat{P}^R_{v+} := \frac{\alpha + \sum_{e_{x \to y} \in \mathbf{E_0}} \mathbb{1}[y = v \wedge \ell_x = R]}{\alpha |V| + \sum_{e_{x \to y} \in \mathbf{E_0}} \mathbb{1}[\ell_x = R]} \tag{16}$$

We obtain approximations for the remaining PA probabilities ('receiving' probabilities) $\hat{P}^F_{v-}$, and $\hat{P}^R_{v-}$ by making the identical substitution of E_0 for E_{i-1} in eqns. 6, and 7 (note that these four approximations are constant for all new edges to/from the same preexisting user v, so we drop i subscripts accordingly).

We now obtain an approximation $\hat{\mathbf{P}}_\mathbf{u}$ of the posterior probability $\mathbf{P}^*_\mathbf{u}$ that new user u is fake by using these approximations in eqns. 4, 5, 8, and 9 to approximate the joint probabilities of all of user u's outgoing & incoming edges conditional on her real/fake label, $\hat{P}^F_{N^+_u}$, $\hat{P}^R_{N^+_u}$, $\hat{P}^F_{N^-_u}$, and $\hat{P}^R_{N^-_u}$, then computing her posterior (eqn. 12). This approach is formalized in the PreAttacK algorithm:

PreAttacK

```
input  Preexisting $G(V, E_0, L_V)$;  new users $U$; new requests $E_n \backslash E_0$
  for $v \in V$ who receives a new request, $\bigcup(v : \{x \longrightarrow v\} \in E_n \backslash E_0)$
    Compute $\hat{P}^F_{v+}$ and $\hat{P}^R_{v+}$
  for $v \in V$ who sends a new request, $\bigcup(v : \{v \longrightarrow x\} \in E_n \backslash E_0)$
    Compute $\hat{P}^F_{v-}$ and $\hat{P}^R_{v-}$
  for new user $u \in U$
    Compute $\hat{P}^F_{N^+_u}$, $\hat{P}^R_{N^+_u}$, $\hat{P}^F_{N^-_u}$, and $\hat{P}^R_{N^-_u}$
    Compute posterior $\hat{\mathbf{P}}_\mathbf{u}$
  return $[\hat{\mathbf{P}}_\mathbf{1}, \ldots, \hat{\mathbf{P}}_{|\mathbf{U}|}]$
```

Below, we show that in our setting PreAttacK obtains near-optimal approximations for the posterior probabilities $\mathbf{P}^*_\mathbf{u}$ at low computational cost. We also show in Section 5 that it can be naturally extended to capture *homophily* or even *monophily*—scenarios where $\alpha = f(\ell_v, \ell_u)$. These extensions incur no cost in terms of complexity, and they slightly improve the approximation bounds.

4 ANALYSIS OF PREATTACK

Our goal in this section is to show that PreAttacK results in improved computational complexity over alternatives, and that it admits instance-specific approximation bounds that confirm near-optimal posterior inference for our problem instance. We note that these are some of the first theoretic guarantees for this problem that do not rely on homophily assumptions.

4.1 Complexity of PreAttacK

Computing all existing users' preferential attachment weights requires $|E_0|+4(|V^+|+|V^-|) \leq |E_0|+8(|V|)$ simple operations, where $V^+, V^- \subseteq V$ respectively refer to the subset of preexisting users who receive and send requests in preexisting network E_0. Then, computing PreAttacK's posterior for all new accounts U requires $2|E_n \backslash E_0| + 2|U|$ operations. Importantly, unlike state-of-the-art algorithms, PreAttacK can be computed for all new accounts in a single pass through all edges [20, 43, 45, 49]. This yields $O(|E_n|)$ asymptotic complexity, which is $O(|V \cup U|)$ in (sparse) social networks [28]. This improves on state-of-the-art algorithms such as SybilBelief, SybilRank, and SybilSCAR, which require $O(m|E'|)$, where m is the number of iterations (at least $O(\log(|V \cup U|))$) and E' is the set of all accepted friend requests [20, 45, 49].

4.2 Instance-specific approximation guarantee

We formalize the three key intuitions from Section 3.2 to derive instance-specific and new-user-specific approximation guarantees. This is advantageous because it allows researchers to also obtain an upper- and lower-bound of the *exact* posterior for each new user, and also to determine the batch size (or subset) of new users that can be classified while maintaining a desired worst-case approximation bound for a specific problem instance. We give the key intuition for the proof here and defer full analysis to Appendix A.

One-sided approximation errors. It is acceptable for PreAttacK to overestimate the posterior probability that a new fake is fake and underestimate the probability that a new real is fake, but not the opposite. Therefore we seek, for each new user u, two bounds: a worst-case approximation factor (underestimate factor) $f^F \leq \hat{\mathbf{P}}_\mathbf{u}/\mathbf{P}^*_\mathbf{u}$, which is useful if u is fake, and a factor (overestimate factor) $f^R \geq \hat{\mathbf{P}}_\mathbf{u}/\mathbf{P}^*_\mathbf{u}$ that is useful in case u is real.

Avoiding the combinatorial problem of new users' labels. Consider f^F. The main difficulty is that we cannot know (without trying all combinations) the worst-case configuration of new users' latent labels that results in the worst underestimate $\hat{\mathbf{P}}_\mathbf{u}/\mathbf{P}^*_\mathbf{u}$. This is because each new user before u may have sent multiple requests to recipients v, some of which result in increases to $\mathbf{P}^*_\mathbf{u}$ (e.g. if the other new user is also fake and targets some of the same recipients as u) and some in decreases (e.g. if the other new user is also fake and targets some recipients who are not among u's recipients).

We sidestep this combinatorial problem by imagining that each new edge to/from a new account prior to u's is sent by a *unique* 'phantom' new account p whose label is the worst-case label for the bound of interest. Thus, for f^F we assume ℓ_p=F if p's single new request is to/from the same preexisting recipient v as one of u's requests, and ℓ_p=R otherwise. Compute u's 'worst case underestimate if u is fake' posterior $\mathbf{P}^\mathbf{F}_\mathbf{u,WC}$ using these 'phantom labels' to obtain $f^F = \hat{\mathbf{P}}_\mathbf{u}/\mathbf{P}^\mathbf{F}_\mathbf{u,WC} \leq \hat{\mathbf{P}}_\mathbf{u}/\mathbf{P}^*_\mathbf{u}$. To then obtain the 'worst case overestimate if u is real' factor f^R, compute $\mathbf{P}^\mathbf{R}_\mathbf{u,WC}$ assuming the opposite: ℓ_p=R if p's single new request is to/from the same preexisting user v as one of u's requests, else ℓ_p=F (see Appendix A).

In Section 6, we show this yields useful approximation bounds for millions of new accounts in real data ($f^F \approx 0.85, f^R \approx 1.1$).

4.3 Adversarial robustness in practice

We also highlight an important property that PreAttacK shares with recent advances in practical adversarial robustness for this problem. The most performant recent algorithms for fake account detection at Facebook obtain adversarial robustness in practice by leveraging so-called 'deep network features' [47], which are features that capture aggregate properties of each user's friends-of-friends. Such aggregates have been shown to be practically difficult for even coordinated campaigns of fake accounts to manipulate, particularly when befriending (at least some) real users. PreAttacK similarly works by aggregating over the features (i.e. counts) of e.g. friend-requesters-of-friend-requestees. As such, PreAttacK's preferential attachment probabilities may also be considered 'deep network features'. Manipulating PreAttacK's prediction for a certain user would require an adversary to manipulate the counts of fake and real senders who send requests to the user's recipients, as well as the counts of known fake and real users to whom the user's requesters also send requests.[3] See also Appendix C.

Below, we also consider a variant of PreAttacK called PreAttacK++ that also prevents sophisticated adversaries from avoiding detection by targeting only very unpopular (and thus uninformative) real users who have sent and received few friend requests.

Finally, we note that in practice on large scale social networks, new approaches to this problem that are practically vulnerable to attack (such as modifying a fake account classifier by adding a new and informative feature that can be manipulated by users) tend to prompt an observable response from sophisticated adversaries (see e.g. [47]). We have observed no such response to PreAttacK.

5 PREATTACK++ AND HOMOPHILY

We also consider a variant of PreAttacK, PreAttacK++, that incorporates *homophily* and/or *monophily*[4] to more rapidly detect fakes. PreAttacK++ captures scenarios where the kCDPA prior probabilities[5] α that an existing user v receives a request from (or sends a request to) a new user u depend on u and v's real/fake labels, and also on whether the new account is the sender or the recipient. This captures e.g. a typical case where a new real account is *a priori* much less likely to send a request to a preexisting fake account vs. a preexisting real account (even if neither has previously received any requests). It can also capture *monophilic* networks where e.g. new fakes prefer to target real users rather than other fakes.

Incorporating these label-dependent probabilities is advantageous because they allow the posterior to update even when a new user sends requests to (or receives requests from) preexisting recipients who have not received any requests, but whose label is known. This also prevents sophisticated fake accounts from avoiding detection by targeting only unpopular recipients.

In the most general case, α can take 8 values: 4 probabilities that a new $\{fake, real\}$ user u *sends* a request to any preexisting $\{fake, real\}$ user v, which we denote by $\alpha^+_{\ell_u\to\ell_v}$ and 4 probabilities that a new $\{fake, real\}$ user u *receives* a request from any preexisting $\{fake, real\}$ v, denoted by $\alpha^-_{\ell_v\to\ell_u}$. Estimates of these probabilities are known or easily obtainable from historical data. PreAttacK++ uses them (per kCDPA) in the approximate probabilities $\hat{P}^F_{v+}$, $\hat{P}^R_{v+}$, $\hat{P}^F_{v-}$, and $\hat{P}^R_{v-}$ of observing each new edge in the first 2 loops in PreAttacK. For example, in PreAttacK++, the probability $\hat{P}^F_{v+}$ that a new fake user sends a request to v becomes:

$$\hat{P}^F_{v+} = \frac{\alpha^+_{\mathbf{F}\to\ell_v} + \sum_{e_{x\to y}\in\mathbf{E_0}} \mathbb{1}[y = v \wedge \ell_x = F]}{\sum_{v\in V}\alpha^+_{\mathbf{F}\to\ell_v} + \sum_{e_{x\to y}\in\mathbf{E_0}} \mathbb{1}[\ell_x = F])} \tag{17}$$

And the probability a new fake receives a request from v becomes:

$$\hat{P}^F_{v-} = \frac{\alpha^-_{\ell_v\to\mathbf{F}} + \sum_{e_{x\to y}\in\mathbf{E_0}} \mathbb{1}[x = v \wedge \ell_y = F]}{\sum_{v\in V}\alpha^-_{\ell_v\to\mathbf{F}} + \sum_{e_{x\to y}\in\mathbf{E_0}} \mathbb{1}[\ell_y = F])} \tag{18}$$

Note that PreAttacK++'s new expressions for $\hat{P}^R_{v+}$ and $\hat{P}^R_{v-}$ can be obtained by substituting R for F everywhere in eqns. 17 and 18.

Note this change does not incur a penalty in terms of complexity. Also, because more informative $\alpha^+_{\ell_u\to\ell_v}$ and $\alpha^-_{\ell_v\to\ell_u}$ values reduce the marginal change in posterior that can accrue due to new edges in each existing user's PA weights, PreAttacK++ admits slightly improved instance-specific bounds compared to PreAttacK for identical problem instances (see Appendix A).

6 EVALUATIONS

Our goal in this section is to show that beyond its provable guarantees, PreAttacK performs well in practice on new fake accounts on the global Facebook network. Our goal is *not* to measure performance on *all* fake accounts, as the current generation of production classifiers already detect the vast majority of fakes during account registration [47, 49]. Similarly, PreAttacK is not an alternative to other production classifiers that detect longer-tenured fake accounts based on their longer timelines of friendships and shared content [30]. Rather, we seek to overcome the *early detection paradox* by rapidly obtaining a good classification after an account passes registration, but before it can engage with real users. Thus, rather than measure performance on all new accounts (including those easily detected by existing means), we instead evaluate the degree to which PreAttacK *improves upon state-of-the-art defenses already in place* [11, 30, 47] by detecting new fake accounts that are not yet detected by those methods. This 'hardest-to-detect' class [11, 23, 47] of new fakes motivates our evaluations.

Our main empirical result is that PreAttacK converges to informative classifications (AUC $\approx$0.9) after new accounts send + receive a total of 20 not-yet-answered friend requests.[6] For comparison, state-of-the-art network-based algorithms do not obtain this performance even after observing additional data on new users' first 100 friend requests. This means that unlike many state-of-the-art algorithms, PreAttacK converges before the median fake account makes a single friendship (accepted request) with a real user.

[3] Alternatively, a sophisticated adversary might attempt to learn and then target the set of real users who are primarily targeted by real users and not fakes (i.e. who have small $\hat{P}^F_{v+}/\hat{P}^R_{v+} < 1$). However, even if this were possible, selection bias dictates that these real users may be less receptive to accepting fakes' friend requests, and the adversary would have to severely limit its fake accounts' friend requests to each real user v to avoid increasing $\hat{P}^F_{v+}$ (which would result in future detection by PreAttacK).

[4] Recall that monophily occurs where one type of user prefers to connect to a specific other type of user, e.g. if fake users send requests to reals rather than other fakes.

[5] We refer to α's as 'probabilities' for readability, but note that in kCDPA, PA probabilities are *proportional* to α, so it is possible to choose parameters $\alpha \in [0, \inf)$.

[6] As is standard, we use the ROC AUC as our metric because real/fake account labels are highly imbalanced (~95% of users are real) [11, 44, 47]. Recall that a perfect classifier has AUC=1, whereas AUC=0.5 denotes 'no better than random'.

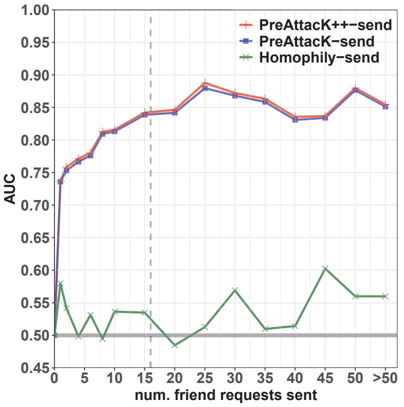

Figure 5: Eval. 1 -send version AUC vs. # friend requests sent.

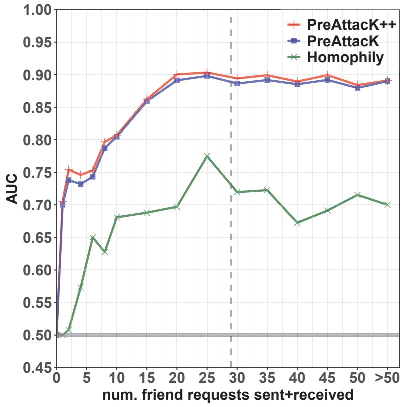

Figure 6: Eval. 1 AUC vs. # friend requests (sent+received).

To accomplish this, we conduct two sets of evaluations. In the first set, we evaluate PreAttacK and its variants on new accounts that joined the global Facebook network, and we show how PreAttacK converges to AUC ≈0.9 as each new account sends and receives its first handful of friend requests. In our second set of evaluations, we compare PreAttacK to four state-of-the-art network-based benchmarks. Because these benchmarks are significantly more computationally intensive than PreAttacK, we restrict our data in this 2nd evaluation to a single country of ~1 million users.

6.1 Evaluation 1 framework

To evaluate PreAttacK's performance on new fake accounts on the global Facebook network, we adopt the evaluation framework of [11]. Specifically, we consider the set of all ($n>10^6$) new accounts that joined the global Facebook network during a particular week last year, along with the time-ordered set of friend requests that they sent and received during that week. Our goal is to determine whether PreAttacK could have accurately classified these new accounts *using just their initial* $1, 2, \ldots 50$ *initial friend requests from this first week after they joined the network*, based on the counts of requests that preexisting accounts had sent and received from real and fake accounts prior to the start of this week (i.e. preexisting users' PA probabilities). Because several months have passed since this 'historical evaluation week', we can now measure the accuracy of PreAttacK's 'early' classifications against high-confidence labels subsequently obtained from production classifiers. [23, 47].

We also confirm PreAttacK guarantees near-optimal approximations ($f^F \geq 0.85, f^R \leq 1.1$) for >90% of these new accounts by computing the instance-specific bounds (see Section 4).

***Homophily* benchmark.** We also consider a simplified variant of PreAttacK: Homophily. Homophily is identical to PreAttacK++ but with existing users' PA probabilities zeroed out except for α terms, such that the probability of each new user's edge to/from *any* existing user is proportional to the overall within- or cross-class rate $\alpha^{+}_{\ell_u \to \ell_v}$ or $\alpha^{-}_{\ell_v \to \ell_u}$ (see Appendix D). By comparing PreAttacK to Homophily, we ascertain the degree to which PreAttacK's performance is homophily-based (i.e. driven by real vs. fake users' different preferences for in-class vs. cross-class friends) versus the degree to which it is driven by differences between real and fake users' preferences for individuals (i.e. our 2-Class PA model).

***PreAttacK-send, PreAttacK++-send, & Homophily-send*.** For each variant, we also compute a '-send' version that only considers the friend requests that new users *sent* (and ignores requests they received). By comparing (for example) PreAttacK-send to PreAttacK, we measure how PreAttacK's performance is driven by the requests that new users send vs. the requests they receive.

***Fast implementation and practical scaling*.** We implement PreAttacK and its variants in PyTorch [31]. On a 40-core 2GHz production virtual machine and even without GPUs, PreAttacK classifies more than a million new accounts-per-second. This efficiency permits us to recompute PreAttacK's posterior after each user's first friend request, second request, and so on in order to obtain real-time-updated classifications for all new accounts.

6.2 Evaluation 1 Results

Fig. 5 plots the AUC of PreAttacK-send versus the count of friend requests *sent* by new accounts. Each (x, y) point in the plot represents the AUC of the corresponding variant of PreAttacK run on just the first x friend requests sent by new accounts during the 'evaluation week'. Here, we observe that PreAttacK-send and PreAttacK++-send already obtain an informative posterior (AUC>0.75) after a new account sends 2 friend requests—well less than the 16 requests it takes the median new fake to make a friendship (i.e. *accepted* request) with a real user. Note that the x-axis of Fig. 5 corresponds to our motivating plot, Fig. 1 in Section 1. PreAttacK-send and PreAttacK++-send then converge to approx. AUC≈0.85 after a new account sends ≈25 friend requests.

Fig. 6 plots the AUC of the full (send+receive) version of PreAttacK versus the total count of friend requests sent+received by new accounts. Here, the additional information regarding the friend requests that new accounts *receive* permits PreAttacK and PreAttacK++ to obtain AUC≈0.9 after each new account sends + receives a total of 20 requests. Thus, they converge before the median fake account makes a friendship (i.e.*accepted* request) with a single real user (which requires a *total* of 29 requests—see Fig. 2).

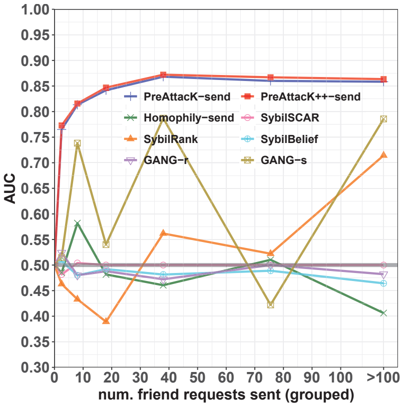

Figure 7: Eval. 2 AUC vs. # friend requests sent.

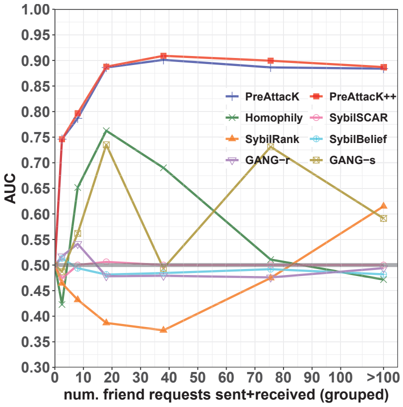

Figure 8: Eval. 2 AUC vs. # friend requests (sent+received).

PreAttacK vs. Homophily. Interestingly, Homophily-send performs only slightly better[7] than random (Fig. 5), and Homophily (Fig. 6) is only moderately informative. The large gap between Homophily vs. PreAttacK suggests that PreAttacK's performance is driven by differences between real and fake users' preferences for individuals (i.e. kCDPA), rather than by real and fake users' different preferences for in-class vs. cross-class friends (i.e. homophily).

PreAttacK vs. PreAttacK++. In both Fig. 5 and Fig. 6, PreAttacK++ (or PreAttacK++-send) offers a small-but-consistent performance improvement of ~0.01-0.02 AUC over PreAttacK (or PreAttacK-send), which is considered nontrivial in this competitive domain [30, 47]. We compared them and found that '++' versions detected additional fakes that were targeting only 'unpopular' existing users whose PA probabilities for both reals and fakes were both small (and thus less informative).

6.3 Evaluation 2 framework

Evaluation 2 compares PreAttacK and its variants to four state-of-the-art network-based fake account detection algorithms: GANG [43], SybilRank, [49], SybilBelief [20], and SybilSCAR [44]. These benchmarks are significantly more computationally intensive than PreAttacK, so we follow [11] and restrict the network to a single country of ~1 million users. This makes it practically feasible to run benchmarks using their papers' original C++ code and parameters. We provide details in Appendix E.

It is computationally impractical to run benchmarks multiple times to compute AUC after each new user's 1st, 2nd, etc. request, so we instead partition new users in Figs. 7 & 8 by #requests they sent (or sent+received): $[0, 5]$, $[6, 10]$, $[11, 25]$, $[26, 50]$, $[51, 100]$, $[101, \infty]$.

6.4 Evaluation 2 Results

Fig. 7 plots the AUC of PreAttacK-send and benchmarks vs. the count of friend requests that new accounts *send*, and Fig. 8 plots the AUC of full PreAttacK vs. the total count of requests that new accounts send+receive. Consistent with their performance on the global Facebook network (Figs. 5 & 6), PreAttacK-send and PreAttacK obtain an informative signal of new accounts' authenticity before the median fake obtains a friendship (*accepted* request) with a single human. In contrast, benchmarks perform poorly on new users, consistent with [11]. We theorize this is because the current generation of new fakes do not exhibit sufficient homophily. GANG-s is a partial exception: it uses the directed network of friend requests (like PreAttacK) to obtain a useful AUC of 0.75-0.85, albeit with high variance (Fig. 7). However, unlike PreAttacK, GANG-s often misclassifies new users that receive many requests (Fig. 8).

7 CONCLUSION

In this paper, we have studied a principled algorithmic approach to address what we call the early detection paradox: mainstream algorithms to detect fake accounts rely on the same behaviors they seek to prevent, such as fake accounts' friendships and the content they share with others. To overcome this paradox, we show some of the first distributional analyses of how fake (and real) accounts send and receive friend requests after joining a major social network, before they have made friends or shared content. We show that these friend request behaviors evoke a natural multi-class extension to the preferential attachment model of social network growth. We leverage this model to derive a new algorithm PreAttacK, and we show that in relevant problem instances, PreAttacK near-optimally approximates the posterior probability that a new user is fake. This approach also provides some of the first theoretic guarantees for fake account detection that do not rely on homophily assumptions. We conduct a variety of evaluations on the global Facebook network, and we consistently find that PreAttacK obtains informative classifications of new accounts before the median fake account succeeds in making a single friendship (i.e. *accepted* friend request) with a real user. We note that, while impressive, PreAttacK's AUC does not match state-of-the-art feature-based classifiers such as DEC, which *eventually* obtains AUC>0.98 on the set of all active accounts by leveraging ~20,000 user-features that describe users' friendships and shared content [47]. Instead, PreAttacK complements such methods by obtaining informative and interpretable early classifications before fake accounts can populate a user-feature vector, share content, or interact with others.

[7] This suggests that the 'hardest-to-detect' new fake accounts in our evaluation set are savvy enough to avoid 'suspicious' friendships with other fakes.

REFERENCES

[1] Muhammad Al-Qurishi, Mabrook Al-Rakhami, Atif Alamri, Majed Alrubaian, Sk Md Mizanur Rahman, and M Shamim Hossain. 2017. Sybil defense techniques in online social networks: a survey. *IEEE Access* 5 (2017), 1200–1219.

[2] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74, 1 (2002), 47.

[3] Chen Avin, Hadassa Daltrophe, Barbara Keller, Zvi Lotker, Claire Mathieu, David Peleg, and Yvonne-Anne Pignolet. 2020. Mixed preferential attachment model: Homophily and minorities in social networks. *Physica A: Statistical Mechanics and its Applications* 555 (2020), 124723.

[4] Chen Avin, Barbara Keller, Zvi Lotker, Claire Mathieu, David Peleg, and Yvonne-Anne Pignolet. 2015. Homophily and the glass ceiling effect in social networks. In *Proceedings of the 2015 conference on innovations in theoretical CS.* 41–50.

[5] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. 2012. Four degrees of separation. In *Proc. of the 4th Annual ACM WEBSCI.* 33–42.

[6] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[7] Alessandro Bessi and Emilio Ferrara. 2016. Social bots distort the 2016 US Presidential election online discussion. *First Monday* 21, 11-7 (2016).

[8] Béla Bollobás, Oliver Riordan, Joel Spencer, and Gábor Tusnády. 2011. The degree sequence of a scale-free random graph process. In *The Structure and Dynamics of Networks*. Princeton University Press, 384–395.

[9] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Lería, Jose Lorenzo, Matei Ripeanu, and Konstantin Beznosov. 2015. Integro: Leveraging Victim Prediction for Robust Fake Account Detection in OSNs.. In *NDSS*, Vol. 15. 8–11.

[10] Yazan Boshmaf, Matei Ripeanu, Konstantin Beznosov, and Elizeu Santos-Neto. 2015. Thwarting fake OSN accounts by predicting their victims. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security.* ACM, 81–89.

[11] Adam Breuer, Roee Eilat, and Udi Weinsberg. 2020. Friend or faux: Graph-based early detection of fake accounts on social networks. In *Proceedings of The Web Conference 2020.* 1287–1297.

[12] Lu Cheng, Ruocheng Guo, Kai Shu, and Huan Liu. 2021. Causal understanding of fake news dissemination on social media. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 148–157.

[13] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. 2010. Who is tweeting on Twitter: human, bot, or cyborg?. In *Proceedings of the 26th annual computer security applications conference.* 21–30.

[14] Nicholas Confessore, Gabriel JX Dance, Richard Harris, and Mark Hansen. 2018. The follower factory. *The New York Times* 27 (2018).

[15] George Danezis and Prateek Mittal. 2009. SybilInfer: Detecting Sybil Nodes using Social Networks.. In *NDSS*. San Diego, CA, 1–15.

[16] Facebook. 2021. *Community Standards Enforcement Report.* https://transparency.fb.com/data/community-standards-enforcement/

[17] Nicholas Fandos and Kevin Roose. 2018. Facebook identifies an active political influence campaign using fake accounts. *The New York Times* 7 (2018).

[18] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Commun. ACM* 59, 7 (2016), 96–104.

[19] Peng Gao, Binghui Wang, Neil Zhenqiang Gong, Sanjeev R Kulkarni, Kurt Thomas, and Prateek Mittal. 2018. Sybilfuse: Combining local attributes with global structure to perform robust sybil detection. In *2018 IEEE conference on communications and network security (CNS).* IEEE, 1–9.

[20] Neil Zhenqiang Gong, Mario Frank, and Prateek Mittal. 2014. Sybilbelief: A semi-supervised learning approach for structure-based sybil detection. *IEEE Transactions on Information Forensics and Security* 9, 6 (2014), 976–987.

[21] Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. 2017. Random walk based fake account detection in OSN's. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on.* IEEE, 273–284.

[22] Sarah Khaled, Neamat El-Tazi, and Hoda MO Mokhtar. 2018. Detecting fake accounts. In *2018 IEEE Intl. Conf. on Big Data (Big Data).* IEEE, 3672–3681.

[23] Fedor Kozlov, Isabella Yuen, Jakub Kowalczyk, Daniel Bernhardt, David Freeman, Paul Pearce, and Ivan Ivanov. 2020. Evaluating Changes to Fake Account Verification Systems. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020).* 135–148.

[24] Sneha Kudugunta and Emilio Ferrara. 2018. Deep neural networks for bot detection. *Information Sciences* 467 (2018), 312–322.

[25] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD Intl. Conference on Knowledge Discovery & Data Mining.* 1269–1278.

[26] Kate Lamb. 2018. I felt disgusted": inside Indonesia's fake Twitter account factories. *The Guardian [Internet]* (2018).

[27] Jay Lee, Manzil Zaheer, Stephan Günnemann, and Alex Smola. 2015. Preferential Attachment in Graphs with Affinities. In *Proceedings of the Eighteenth Intl. Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 38)*, Guy Lebanon and S. V. N. Vishwanathan (Eds.). PMLR, San Diego, California, USA, 571–580. http://proceedings.mlr.press/v38/lee15b.html

[28] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement & analysis of OSN's. In *Proceedings of the 7th ACM SIGCOMM conf. on Internet measurement.* ACM, 29–42.

[29] Buddhika Nettasinghe, Nazanin Alipourfard, Vikram Krishnamurthy, and Kristina Lerman. 2021. A Directed, Bi-Populated Preferential Attachment Model with Applications to Analyzing the Glass Ceiling Effect. *arXiv preprint arXiv:2103.12149* (2021).

[30] Nima Noorshams, Saurabh Verma, and Aude Hofleitner. 2020. TIES: Temporal Interaction Embeddings For Enhancing Social Media Integrity At Facebook. In *26th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining.* 3128–3135.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, Vol. 32.

[32] Derek de Solla Price. 1976. A general theory of bibliometric and other cumulative advantage processes. *Journal of the Am. society for Info. Sci.* 27, 5 (1976), 292–306.

[33] Devakunchari Ramalingam and Valliyammai Chinnaiah. 2018. Fake profile detection techniques in large-scale online social networks: A comprehensive review. *Computers & Electrical Engineering* 65 (2018), 165–177.

[34] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Alessandro Flammini, and Filippo Menczer. 2017. The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592* 96 (2017), 104.

[35] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media. *ACM SIGKDD explorations* 19, 1 (2017), 22–36.

[36] Tao Stein, Erdong Chen, and Karan Mangla. 2011. Facebook immune system. In *Proceedings of the 4th workshop on social network systems.* 1–8.

[37] Stefan Stieglitz, Florian Brachten, Björn Ross, and Anna-Katharina Jung. 2017. Do social bots dream of electric sheep? A categorisation of social media bot accounts. *arXiv preprint arXiv:1710.04044* (2017).

[38] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. 2011. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference.* 243–258.

[39] Christopher Torres-Lugo, Manita Pote, Alexander C Nwala, and Filippo Menczer. 2022. Manipulating Twitter Through Deletions. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 16. 1029–1039.

[40] Onur Varol, Emilio Ferrara, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2017. Online human-bot interactions: Detection, estimation, and characterization. In *Proc. of the Intl. AAAI Conf. on Web and Social Media*, Vol. 11.

[41] Svitlana Volkova and Eric Bell. 2017. Identifying effective signals to predict deleted and suspended accounts on twitter across languages. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 11.

[42] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *science* 359, 6380 (2018), 1146–1151.

[43] Binghui Wang, Neil Zhenqiang Gong, and Hao Fu. 2017. GANG: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs. In *Data Mining (ICDM), 2017 IEEE Intl. Conference on.* IEEE, 465–474.

[44] Binghui Wang, Jinyuan Jia, Le Zhang, and Neil Zhenqiang Gong. 2019. Structure-based sybil detection in social networks via local rule-based propagation. *IEEE Transactions on Network Science and Engineering* (2019).

[45] Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. 2017. SybilSCAR: Sybil detection in online social networks via local rule based propagation. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE.* IEEE, 1–9.

[46] Yuchung J Wang and George Y Wong. 1987. Stochastic blockmodels for directed graphs. *J. Amer. Statist. Assoc.* 82, 397 (1987), 8–19.

[47] Teng Xu, Gerard Goossen, Huseyin Kerem Cevahir, Sara Khodeir, Yingyezhe Jin, Frank Li, Shawn Shan, Sagar Patel, David Freeman, and Paul Pearce. 2021. Deep Entity Classification: Abusive Account Detection for Online Social Networks. In *Proceedings of the 30th (USENIX) Security Symposium ((USENIX) Security 21).*

[48] Jilong Xue, Zhi Yang, Xiaoyong Yang, Xiao Wang, Lijiang Chen, and Yafei Dai. 2013. Votetrust: Leveraging friend invitation graph to defend against social network sybils. In *2013 Proceedings IEEE INFOCOM.* IEEE, 2400–2408.

[49] X Yang, Q Cao, and M Sirivianos. 2012. SybilRank: Aiding the detection of fake accounts in large scale social online services.

[50] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. 2000. Generalized belief propagation. In *NIPS*, Vol. 13. 689–695.

[51] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. 2003. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium* 8 (2003), 236–239.

[52] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. 2008. Sybillimit: A near-optimal social network defense against sybil attacks. In *2008 IEEE Symposium on Security and Privacy (sp 2008).* IEEE, 3–17.

[53] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. 2006. Sybilguard: defending against sybil attacks via social networks. In *ACM SIGCOMM Computer Communication Review*, Vol. 36. ACM, 267–278.

[54] Chenwei Zhang, Yi Bu, Ying Ding, and Jian Xu. 2018. Understanding scientific collaboration: Homophily, transitivity, and preferential attachment. *Journal of the Association for Information Science and Technology* 69, 1 (2018), 72–86.

APPENDIX

A DEFERRED ANALYSIS FOR INSTANCE-SPECIFIC BOUNDS

***Lower bound* f^F.** We seek a worst-case factor $f^F \leq \hat{\mathbf{P}}_{\mathbf{u}}/\mathbf{P}^*_{\mathbf{u}}$ that bounds PreAttacK's underestimation of the posterior probability that u is fake, which is useful in case u is fake (Section 4). The main difficulty is that a new user who sent/received requests before u may have had both positive and negative effects (i.e. via its different edges) on u's posterior. To sidestep the problem of trying all combinations of new users' latent labels, we bound the worst-case by supposing each new edge before u's edges contained a *unique* new 'phantom' user whose latent label was the worst-case for its respective edge. Thus, any new user's edge before u's edges that contained the same preexisting user as the one in u's edge gets a *fake* phantom user; any other such edge gets a *real* phantom user. We compute the exact posterior (eqn. 12) using these phantom users' labels (in place of the latent ones) to obtain the desired bound. The probabilities for each of u's observed edges become:

$$P^F_{v_i+} \coloneqq P[v_i|\ell_u{=}F,\cdot] = \frac{\alpha + \sum_{e_{x\to y}\in E_{i-1}} \mathbb{1}[y{=}v \wedge \ell_x{=}F]}{\alpha|V| + \sum_{e_{x\to y}\in E_{i-1}} \mathbb{1}[\ell_x{=}F]} \leq P^F_{v_i+,WCF} \coloneqq \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[y{=}v \wedge \ell_x{=}F] + \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[y{=}v]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_x = F] \;+\; \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[y = v]} \tag{19}$$

$$P^R_{v_i+} \coloneqq P[v_i|\ell_u{=}R,\cdot] = \frac{\alpha + \sum_{e_{x\to y}\in E_{i-1}} \mathbb{1}[y{=}v \wedge \ell_x{=}R]}{\alpha|V| + \sum_{e_{x\to y}\in E_{i-1}} \mathbb{1}[\ell_x{=}R]} \geq P^R_{v_i+,WCF} \coloneqq \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[y{=}v \wedge \ell_x{=}R]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_x{=}R] + \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[y{\neq}v]} \tag{20}$$

$$P^F_{v_i-} \coloneqq P[v_i|\ell_u{=}F,\cdot] = \frac{\alpha + \sum_{e_{x\to y}\in E_{i-1}} \mathbb{1}[x{=}v \wedge \ell_y{=}F]}{\alpha|V| + \sum_{e_{x\to y}\in E_{i-1}} \mathbb{1}[\ell_y{=}F]} \leq P^F_{v_i-,WCF} \coloneqq \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[x{=}v \wedge \ell_y{=}F] + \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[x = v]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_y{=}F] \;+\; \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[x = v]} \tag{21}$$

$$P^R_{v_i-} \coloneqq P[v_i|\ell_u{=}R,\cdot] = \frac{\alpha + \sum_{e_{x\to y}\in E_{i-1}} \mathbb{1}[x{=}v \wedge \ell_y{=}R]}{\alpha|V| + \sum_{e_{x\to y}\in E_{i-1}} \mathbb{1}[\ell_y{=}R]} \geq P^R_{v_i-,WCF} \coloneqq \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[x{=}v \wedge \ell_y{=}R]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_y{=}R] + \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[x \neq v]} \tag{22}$$

$$\hat{\mathbf{P}}_{\mathbf{u}}/\mathbf{P}^*_{\mathbf{u}} \coloneqq \hat{\mathbf{P}}_{\mathbf{u}}/\Big(\frac{P^F_{\mathcal{N}^+_u}\cdot P^F_{\mathcal{N}^-_u}\cdot \pi}{P^F_{\mathcal{N}^+_u}\cdot P^F_{\mathcal{N}^-_u}\cdot \pi \;+\; P^R_{\mathcal{N}^+_u}\cdot P^R_{\mathcal{N}^-_u}\cdot (1-\pi)}\Big) \tag{23}$$

$$= \hat{\mathbf{P}}_{\mathbf{u}}/\Big(\frac{\prod_{v_i\in\mathcal{N}^+_u} P^F_{v_i+} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^F_{v_i-} \cdot \pi}{\prod_{v_i\in\mathcal{N}^+_u} P^F_{v_i+} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^F_{v_i-} \cdot \pi \;+\; \prod_{v_i\in\mathcal{N}^+_u} P^R_{v_i+} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^R_{v_i-} \cdot (1-\pi)}\Big) \tag{24}$$

$$\geq f^F \coloneqq \hat{\mathbf{P}}_{\mathbf{u}}/\Big(\frac{\prod_{v_i\in\mathcal{N}^+_u} P^F_{v_i+,WCF} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^F_{v_i-,WCF} \cdot \pi}{\prod_{v_i\in\mathcal{N}^+_u} P^F_{v_i+,WCF} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^F_{v_i-,WCF} \cdot \pi \;+\; \prod_{v_i\in\mathcal{N}^+_u} P^R_{v_i+,WCF} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^R_{v_i-,WCF} \cdot (1-\pi)}\Big) \tag{25}$$

This expression for f^F requires no knowledge of new users' latent labels, and conveniently, it can be computed during the same single pass through new edges that we use to compute PreAttacK with no penalty in asymptotic complexity (Note also that the expression for f^F can be further factored as in eqn. 12).

***Upper bound* f^R.** We also seek a worst-case overestimate factor $f^R \geq \hat{\mathbf{P}}_{\mathbf{u}}/\mathbf{P}^*_{\mathbf{u}}$ that bounds PreAttacK's overestimation of the posterior probability that u is fake, which is useful in case u is real. Similar to before, we bound the worst-case by computing the exact posterior (eqn. 12) supposing each new edge before u's edges contained a unique new 'phantom' user whose latent label was the worst-case for u's posterior:

$$P^F_{v_i+} \geq P^F_{v_i+,WCR} \coloneqq \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[y = v \wedge \ell_x = F]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \big(\alpha + \mathbb{1}[\ell_x = F]\big) \;+\; \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[y \neq v]} \tag{26}$$

$$P^R_{v_i+} \leq P^R_{v_i+,WCR} \coloneqq \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[y = v \wedge \ell_x = R] + \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[y = v]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_x = R] + \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[y = v]} \tag{27}$$

$$P^F_{v_i-} \geq P^F_{v_i-,WCR} \coloneqq \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[x = v \wedge \ell_y = F]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_y = F] \;+\; \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[x \neq v]} \tag{28}$$

$$P^R_{v_i-} \leq P^R_{v_i-,WCR} \coloneqq \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[x = v \wedge \ell_y = R] + \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[x = v]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_y = R] + \sum_{e_{x\to y}\in E_{i-1}\setminus E_0} \mathbb{1}[x = v]} \tag{29}$$

$$\hat{\mathbf{P}}_{\mathbf{u}}/\mathbf{P}^*_{\mathbf{u}} \coloneqq \hat{\mathbf{P}}_{\mathbf{u}}/\Big(\frac{\prod_{v_i\in\mathcal{N}^+_u} P^F_{v_i+} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^F_{v_i-} \cdot \pi}{\prod_{v_i\in\mathcal{N}^+_u} P^F_{v_i+} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^F_{v_i-} \cdot \pi \;+\; \prod_{v_i\in\mathcal{N}^+_u} P^R_{v_i+} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^R_{v_i-} \cdot (1-\pi)}\Big) \tag{30}$$

$$\leq f^R \coloneqq \hat{\mathbf{P}}_{\mathbf{u}}/\Big(\frac{\prod_{v_i\in\mathcal{N}^+_u} P^F_{v_i+,WCR} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^F_{v_i-,WCR} \cdot \pi}{\prod_{v_i\in\mathcal{N}^+_u} P^F_{v_i+,WCR} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^F_{v_i-,WCR} \cdot \pi \;+\; \prod_{v_i\in\mathcal{N}^+_u} P^R_{v_i+,WCR} \cdot \prod_{v_i\in\mathcal{N}^-_u} P^R_{v_i-,WCR} \cdot (1-\pi)}\Big) \tag{31}$$

Note that f^F is strictly decreasing, and f^R strictly increasing in the number of new users' edges before u's.

B MULTI-CLASS PREATTACK

PreAttacK also applies to the case where we want to classify $k{>}2$ classes $\kappa \in \{1, \ldots, k\}$ of fake users, such as sockpuppets, false news bots [42], etc. and each has different preferences in terms of existing users they seek to befriend. Computing PreAttacK in this case, we first compute k conditional probabilities $\hat{P}^{\kappa}_{v_i+}$ of each observed edge that new user u sends—one for each class κ—and also k conditional probabilities $\hat{P}^{\kappa}_{v_i-}$ of each observed edge that new user u receives:

$$\hat{P}^{\kappa}_{v_i+} = \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[y = v \wedge \ell_x = \kappa]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_x = \kappa]} \; ; \tag{32}$$

$$\hat{P}^{\kappa}_{v_i-} = \frac{\alpha + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[x = v \wedge \ell_y = \kappa]}{\alpha|V| + \sum_{e_{x\to y}\in \mathbf{E_0}} \mathbb{1}[\ell_y = \kappa]} \tag{33}$$

We use these (approximated) conditional probabilities to compute k joint conditional probabilities $P^{\kappa}_{\mathcal{N}^+_u}$ of all u's outgoing requests, and k probabilities $P^{\kappa}_{\mathcal{N}^-_u}$ of all u's incoming requests. Finally we compute $k{-}1$ (approximated) posterior probabilities—one for each class that could describe the new account (the k'th is implied). Here, the posterior probability that u is a member of class κ that has prior probability π^{κ} is:

$$\hat{\mathbf{P}}^{\kappa}_{\mathbf{u}} = \frac{\hat{P}^{\kappa}_{\mathcal{N}^+_u} \cdot \hat{P}^{\kappa}_{\mathcal{N}^-_u} \cdot \pi^{\kappa}}{\hat{P}^{\kappa}_{\mathcal{N}^+_u} \cdot \hat{P}^{\kappa}_{\mathcal{N}^-_u} \cdot \pi^{\kappa} \;+\; \sum\limits_{\gamma=\{1,\ldots,\kappa\}\backslash\kappa}^{k} \hat{P}^{\gamma}_{\mathcal{N}^+_u} \cdot \hat{P}^{\gamma}_{\mathcal{N}^-_u} \cdot \pi^{\gamma}} \tag{34}$$

$$= \left(1 \;+\; (P^{\kappa}_{\mathcal{N}^+_u} \cdot P^{\kappa}_{\mathcal{N}^-_u} \cdot \pi^{k})^{-1} \sum_{\gamma=\{1,\ldots,\kappa\}\backslash\kappa}^{k} \hat{P}^{\gamma}_{\mathcal{N}^+_u} \cdot \hat{P}^{\gamma}_{\mathcal{N}^-_u} \cdot \pi^{\gamma}\right)^{-1} \tag{35}$$

C ADVERSARIAL ROBUSTNESS IN PRACTICE: ADDITIONAL DISCUSSION

We also note a relationship between our work and the very recent discussions in the fake account and fake news detection community that have highlighted causal considerations, and in particular, unobserved confounding and related biases (see e.g. Cheng et al. [12]). Specifically, algorithmic approaches to fake accounts (or fake news) will suffer from bias to the extent that they leverage users' counts of fake friends (or fake news articles shared) without accounting for the fact that some users had more exposure to fake accounts (e.g. by receiving more friend requests from fakes) than others *a priori*, and some users had more exposure to real accounts than others *a priori.* Such bias can explicitly accrue when, for example, algorithms increase the posterior belief that a new account is real by the same amount for each real account she befriends, failing to account for the fact that different users have varying propensities to receive friend requests from fake accounts. It is also well-known that adversaries can leverage this bias to avoid detection. For example, an adversary might avoid detection by mainstream algorithms by sending thousands of friend requests to real users in the hope of befriending a 'normal' number of real friends, knowing that certain algorithms are blind to rejected friend requests and unable to account for the fake account's unduly high *a priori* exposure to real users. Our distributional analyses in Section 1 confirm this 'varying exposure' phenomenon for our problem instance. Sophisticated adversaries may also attempt to learn the subset of real users who accept friend requests indiscriminately, or they may strategically delete certain friendships after making them in order to manipulate detection algorithms (see e.g. [39, 47, 48]).

Whereas several new research streams seek to address these sources of bias via propensity scoring and other inference techniques, PreAttacK sidesteps them entirely by aggregating over all friend requests (rather than the subset that are accepted). More importantly, PreAttacK does not require inferential propensity scoring corrections to address bias from to the fact that different users have varying exposure to fake accounts, as we are able to use social network data to explicitly compute each existing user's 'propensity' to receive a friend request from (or send a request to) fake and real accounts, which we compute as each user's preferential attachment probabilities, $\hat{P}^F_{v+}, \hat{P}^R_{v+}, \hat{P}^F_{v-}, \hat{P}^R_{v-}$. In this way, PreAttacK may be seen as robust to prevalent confounding bias vulnerabilities that are common among mainstream approaches in this application domain.

D DETAILS OF HOMOPHILY BENCHMARK

Homophily is equivalent to PreAttacK++ with $E_0 = \varnothing$. Thus, all request probabilities in Homophily can be summarized by an 8-tuple, including 4 probabilities $\alpha^+_{\ell_u \to \ell_v}$ that a new $\{fake, real\}$ *sends* a request to a preexisting $\{fake, real\}$ and 4 probabilities $\alpha^-_{\ell_v \to \ell_u}$ that a new $\{fake, real\}$ *receives* a request from a preexisting $\{fake, real\}$. Here, kCDPA reduces to a 2-class directed Stochastic Block Model (SBM) [46].

E DETAILS OF BENCHMARK ALGORITHMS

Section 6 compares PreAttacK and its variants to four state-of-the-art benchmarks. For each, we use their paper's code and parameters:

GANG. GANG [43] is a recent algorithm that leverages directed edges (requests) in a belief-propagation framework. We consider two variants: GANG-s, which uses the directed network of friendship requests *sent*, and GANG-r, which uses the directed network of requests *received.* This allows us to test GANG's performance when beliefs about the authenticity of a new user flow from senders to receivers, or alternatively, from receivers to senders. As with SybilBelief, we set parameters $\{\theta^+, \theta^-, \theta\}$ to $\{0.9, 0.1, 0.5\}$ per [43].

SybilRank. SybilRank [49] is currently the most widely used random walk based algorithm. SybilRank runs on the network of accepted friend requests and set of known real users. As in [49], we run SybilRank for log2($|V|$) iterations.

SybilBelief. SybilBelief [20] is a loopy belief propagation algorithm that is widely used in state-of-the-art applications. SybilBelief uses the network of accepted friend requests and both known real users and fakes. As in [20], we run SybilBelief with edge weights of 0.9 and set $\{\theta^+, \theta^-, \theta\}$ to $\{0.9, 0.1, 0.5\}$.

SybilSCAR. SybilSCAR [44] is a recent algorithm that uses the graph of accepted requests and both known real users and fakes. We run both versions: SybilSCAR-C with weights equal to half the inverse of the avg. degree per [44], and user-degree weighted SybilSCAR-D. Each point in Figs. 7 & 8 reports the higher of their two AUC's. Per [44], we set $\{\theta^+, \theta^-, \theta\}$ to $\{0.6, 0.4, 0.5\}$, and $\delta{=}10^{-3}$.