Evaluating Changes to Fake Account Verification Systems

Fedor Kozlov[†], Isabella Yuen[†], Jakub Kowalczyk[†], Daniel Bernhardt[†], David Freeman[†],
Paul Pearce[†‡], and Ivan Ivanov[†]
[†]*Facebook, Inc*
[‡]*Georgia Institute of Technology*

Abstract

Online social networks (OSNs) such as Facebook, Twitter, and LinkedIn give hundreds of millions of individuals around the world the ability to communicate and build communities. However, the extensive user base of OSNs provides considerable opportunity for malicious actors to abuse the system, with fake accounts generating the vast majority of harmful actions and content. Social networks employ sophisticated detection mechanisms based on machine-learning classifiers and graph analysis to identify and remediate the actions of fake accounts. Disabling or deleting these detected accounts is not tractable when the number of false positives (i.e., real users disabled) is significant in absolute terms. Using challenge-based verification systems such as CAPTCHAs or phone confirmation as a response for detected fake accounts can enable erroneously detected real users to recover their access, while also making it difficult for attackers to abuse the platform.

In order to maintain a verification system's effectiveness over time, it is important to iterate on the system to improve the real user experience and adapt the platform's response to adversarial actions. However, at present there is no established method to evaluate how effective each iteration is at stopping fake accounts and letting real users through. This paper proposes a method of assessing the effectiveness of experimental iterations for OSN verification systems, and presents an evaluation of this method against human- labelled ground truth data using production Facebook data. Our method reduces the volume of necessary human labelled data by 70%, decreases the time necessary for classification by 81%, has suitable precision/recall for making decisions in response to experiments, and enables continuous monitoring of the effectiveness of the applied experimental changes.

1 Introduction

Online Social Networks (OSNs) enable people to build communities and communicate effortlessly. With the proliferation of social media usage, OSNs now play a role in the lives of billions of people every day. The largest social networks—Facebook, Twitter, LinkedIn, and Instagram—provide a broad set of features enabling more than two billion people to share news, media, opinions, and thoughts [12, 49]. The scale and scope of these OSNs in turn attract highly motivated attackers, who seek to abuse these platforms and their users for political and monetary gain [3].

The prevalence, impact, and media coverage of harmful social media accounts has increased commensurately with the growth of the platforms [8, 28]. A key contributor to this problem is *fake accounts*—accounts that do not represent an authentic user, created for the express purpose of abusing the platform or its users.

Recent research estimates as much as 15% of all Twitter accounts to be fake [51], and Facebook estimates as much as 4% of their monthly active users to fall into this category [11]. These fake accounts post spam, compromise user data, generate fraudulent ad revenue, influence opinion, or engage in a multitude of other abuses [14, 15, 38, 44, 48].

The variety of behaviours exhibited by fake accounts—especially those controlled by humans—makes building accurate detection systems a challenge. On a platform with billions of active users, a detection system with even 99% precision would incorrectly identify hundreds of thousands of users *every day* as malicious. It follows that OSNs require remediation techniques that can tolerate false positives without incurring harm, while still providing significant friction for attackers.

A common OSN remediation technique is to enroll fake accounts detected by a detection system into a *verification system* [17, 33] aimed at blocking access to the OSN for fake accounts and providing a way to recover an account for legitimate users. These systems are composed of *challenges* which prompt identified users to provide some additional information such as phone numbers, recent activity, or identity verification. These challenges—of which the best known example is a CAPTCHA [53]—take the form of challenge-response tests that are designed to be easy for real users to pass, but difficult for attackers to solve. Verification systems

have numerous advantages over direct disabling of accounts. They provide a soft response that is tolerant of false positives: a real user classified as potentially fake has semi-automated means of changing the classification result without substantial impact on their engagement. The challenges themselves provide an opportunity to collect additional signals about the user (e.g., time-to-solve), which can aid in further investigation, re-classification, and remediation. The strength (friction) of the challenge can be scaled based on initial classification confidence of the detection system.

Despite these advantages, attackers can adapt to overcome the friction posed by verification system challenges [27, 35, 41]. It follows that continuously iterating on the design of those challenges and being able to measure the *effectiveness* of the iterations over time is an important component of improving fake account defences, which has not yet been addressed in the research literature.

We seek to understand *iteration effectiveness*: the degree to which a new or improved challenge is more successful in both stopping fake accounts and letting real users through. To compare effectiveness, we subject pools of accounts to two different experiences in an A/B experiment and compute the change in the proportion of fake and real accounts that managed to successfully pass the verification process. This computation is particularly challenging as it involves determining the true nature of a set of users that were already identified as fake (with high-probability) by an in-production detection framework. To aid in classification one could leverage human labelling of accounts at various stages within and after the verification process. However, relying on human labelling limits the scale and speed of experiments, especially when we require that: many experiments can be run at the same time; we support *backtests*, a technique where some flows are withheld from a small proportion of the population after they have become the default experience, in order to gauge adversarial response; experiments must be powerful enough to show results on different user segments (e.g., platform, locale).

To enable such classification at scale and across such requirements, our approach is to replace the majority of human labelling with automated techniques having precision/recall suitable for both making decisions on the experiments and continuously monitoring the effectiveness of the applied experimental changes.

Our contribution: In this work we develop an automated, scalable method of assessing the effectiveness of experimental iterations for OSN verification systems. A important insight is that we only need weak labels (i.e., "likely" labels) in order to enable rapid experimentation.

Our approach, which we call the *Post Authentication State Model* (PAS), reproduces in an automated way the process that human investigators use to determine the authenticity of an account. PAS requires accounts to be observed for a certain period of time after the verification process in order to collect additional signals, after which they are evaluated against a continuously retrained machine-learned ensemble decision tree of account behaviours. Using this model to evaluate test and control groups of accounts that pass the verification system allows us to determine the change in post-verification fake/real distributions and ultimately how successful an introduced change is at improving the system's effectiveness. Section 3 provides an overview of Facebook's verification system and relevant background. Section 4 discusses the design of this model and several variants. We assess our approach with experiments conducted on Facebook's production verification system, described in Section 5. Our system: enables rapid A/B experimentation; supports an arbitrary number of backtests of the experimental changes, allowing us to continuously monitor the effectiveness of the improvements and adversarial response over time; supports a variety of verification system challenges.

We deployed our approach in a real-world setting at Facebook to assess its potential effectiveness. Our approach, PAS, provided useful signal on whether accounts clearing the verification system were real or fake; it vastly out-performed random assignment, achieving precision over 70% and recall over 60% for all three classes. This approach reduced the volume of human labelling for the life cycle of an experiment by 70%, and the labelling frequency from continuous to a single post-experiment operation. Practically, we showed that our approach could reduce the time necessary for classification by up to 81%. This reduction in human effort allowed Facebook to run more experiments in parallel, improving the agility and scale of their experimentation methods.

Furthermore, the deployed model completely automated the backtests of successfully launched experiments. Thanks to automated backtesting, three instances of adversarial adaptation to the experimental changes were discovered, allowing the Facebook team to quickly find appropriate mitigations.

Out-of-scope: In this work, we focus on classification of fake and real accounts that were already detected by an in-production detection framework and were able to pass challenges in OSN verification systems, such as CAPTHA and phone confirmation. Automated classification of these accounts enables an assessment of experimental iterations for OSN verification systems in order to improve real user experience and increase friction for fake accounts. Based on description above, we consider the following areas out of scope of this work: improvements to efficiency and accuracy of existent fake account detection systems and methods; measurement of recall and precision of fake account detection systems; and improvements made to verification systems.

2 Related Work

There is a large literature examining fake accounts in social networks. This work touches on understanding what the ac-

counts are doing (e.g., scamming, impersonation, etc.), methods for detecting fake accounts, and providing techniques (e.g. CAPTCHA) to effectively address detected fake accounts.

2.1 Types of Fake Accounts

Fake accounts (sometimes called *sybils* [56]) can be divided into three broad classes: *automated*, *manual*, and *hybrid* [7, 21]. Automated fake accounts—*social bots*—are software-controlled profiles that can post and interact with legitimate users via an OSNs's communication mechanisms, just like real people [38]. Usually, social bots are created at scale via automated registration of new accounts in OSNs. The types of abuse caused by social bots varies. There have been instances of social bots that participate in organised campaigns to influence public opinion, spread false news, or gain personal benefits [2, 44]. Recently, social bots have targeted and infiltrated political discourse, manipulated the stock market, stolen personal data, and spread misinformation [15].

In contrast, manually driven fake accounts (MDFA) are set up for a specific purpose without using automation, and are then operated manually by attackers to gain personal benefit [20], push propaganda [28], or otherwise harm users of the platform. The close similarity between actual users and MDFAs breaks traditional at-scale detection techniques which focus on identifying automated behaviours.

Hybrid fake accounts (sometimes called *cyborgs* [7]) include fake accounts driven by bot-assisted humans or human-assisted bots. In practice, sophisticated attackers may choose a mix of tactics for running cyborg fake accounts. Cyborgs are often used for the same purposes as social bots, such as spam and fake news [39].

2.2 Detecting Fake Accounts

The topic of detection of fake accounts is actively explored in recent literature. Research has mostly focused on the design and measurement of detection systems with the purpose of increasing precision and recall. Detection frameworks can be based on different methodologies.

Graph-based and sybil detection focuses on exploring connections between identities and their properties inside social graph to detect fake accounts [9, 23, 56]. A typical example of graph-based sybil detection framework is Sybilguard [58]. The detection protocol of this framework is based on the graph among users, where an edge between two users indicates a human-established trust relationship. Malicious users can create many identities but few trust relationships. Therefore, there is a disproportionately-small "cut" in the graph between the sybil nodes and the honest nodes. Other examples of the detection frameworks based on this methodology that use various algorithms and assumptions about social graph are Sybillimit [57], Sybilinfer [10], SybilRank [5].

Behaviour-based and spam detection employs rule-base heuristics to detect fake accounts. An example of such heuristic is rate limits on specific user activity such as comments and posts and anomalies of such activities. This methodology focuses on high precision to avoid high false positive rate in detection and usually shows low recall [45,52,54,59]. Another example of behaviour-based cetection system is SynchroTrap. This system employs clustering of accounts according to the similarity of their actions to detect large groups of abusive accounts [6].

Machine learning detection frameworks use machine learning models to detect fake accounts [16, 24, 47, 55]. Machine learning models are usually trained based on human labelled data or high precision proxies and utilize an extracted set of user's behavioral features. One of the first examples of such machine learning detection frameworks was proposed by Stein et al. [43]. There are two main downsides of this methodology: it is challenging to properly design features that are resistant to adversarial response, and the process of collecting high precision training data based on humal labelling is expensive.

Digital footprint detection employs digital footprints of OSN users to detect fake and malicious accounts across different social networks. A digital footprint is generated based on publicly available information about a user, such as username, display name, description, location, profile image and IP address [29, 46].

Described methodologies of fake account detection and detection frameworks can't be directly used to measure effectiveness of the improvements in verification systems for fake accounts because users in verification systems are already classified as fakes by detection frameworks. However, in the proposed approach, we use learnings and techniques from machine-learning, graph-based and behaviour-based detection methodologies.

2.3 Remediating Fake Accounts

Once fake accounts are detected, social networks must decide how to respond. Typical actions that a social network might take on detected fake accounts include disabling or deletion. Such responses might be appropriate in some particular cases, where the approximate cost of abusive actions taken by fake accounts and the cost of disabling a real user can both be established. In such cases, the detection framework owner can use this information to make a trade-off between recall and precision [36]. However, representing user actions and cost in financial terms typically will only apply to very narrow scenarios like e-commerce transactions.

In order to allow incorrectly detected real users to regain access to the system, OSNs employ verification systems and challenges. There are numerous types of challenges, including email verification, CAPTCHA resolution, phone confirmation, time and IP address restrictions, challenge questions

and puzzles, manual review of the account, ID verification, facial/voice recognition, and challenges based on liveness detection systems [1, 25, 33, 42, 50]. Most prior work related to verification systems for fake accounts covers new types of verification challenges [22, 30–32] or ways to bypass these systems [26, 60]. This paper is focused on the effectiveness measurement of the improvements in verification systems for fake accounts, for which there is no prior exploration.

3 Background

In this section we frame the overall space of fake account verification systems, outline the metrics used to evaluate the effectiveness of such systems, and discuss prior systems used at Facebook.

3.1 Verification Systems and Clearance

The purpose of OSN fake account verification systems is to block access to the OSN for accounts detected by fake account detection systems; present those accounts with various challenges that allow them to identify themselves as legitimate; collect additional signals by means of those challenges; and ultimately make a determination if an account is real or fake.

An account that is determined to be real is said to "clear" the challenge. Figure 1 shows the structure of an OSN fake account verification system such as the one used at Facebook. A particular path an account takes through the system, which involves passing one or more challenges, is called a *flow*. A flow is divided into *flow points*, or *steps*, which describe the current state of the account within the verification system. Each step can have a number of outcomes, which result in transitions to different steps in the flow or back to the same step. Thus the verification system is essentially a set of possible flows on a directed (possibly) cyclic graph, where the nodes are the steps and the edges are the possible step transitions.

A step is most often associated with a user interface (UI) screen that either requires user input or contains some information for the user, for example an introduction step that explains the reason for being enrolled into the verification system. Some steps contain only a piece of business logic and are invisible to the user. An example of such a step is the *challenge chooser*, which contains rules to decide whether the user has provided sufficient information to determine the authenticity of their account; if the answer is negative, this step will also decide which challenge to show the user. In the context of the flow graph, a challenge is represented as group of one or more steps that need to be completed to proceed forward through the flow.

Each challenge and the steps within it present variable friction to the user, defined as the degree of difficulty in solving the challenges or proceeding through a given step. This friction causes two observable phenomena in the flow graph. The first phenomena is *churn*, defined as the number of users which do not proceed further through the flow in a given step, which reveals how restrictive a step is for a user. The second phenomena is *anomalies in step completion*, such as spikes or long term drift, which reveal, for example, that bad actors have become proficient at solving the challenge or that there is a loophole in the system being exploited by attackers.

To measure these phenomena, Facebook uses a "funnel logging" system. This system tracks transition events through the flow graph—when a user proceeds from one step to another step, receives a challenge, starts or finishes the verification system flow. Figure 1 shows such events as dots labelled with dashed boxes. Along with those transition events, event metadata such as country, device, or user age are logged in order to be able to understand how clearance rates vary across user segments.

Funnel logging allows us to calculate clearance rate metrics that quantify the overall friction for the step, challenge, or verification system as a whole. We can also calculate these metrics for different sub-populations or segments of users. For a specific subpopulation segment Y, enrolled on day d_e, which cleared step s on day d_c, we define the *step clearance rate* C as:

$$C(d_e, d_c, s, Y) = \frac{|d_e, d_c, s, Y|}{|d_e, Y|},$$

where $|\cdot|$ denotes the number of users in a population defined by the given variables. The step clearance rate can be used to calculate the end-to-end challenge or system clearance rate by using the last step of the challenge or flow, respectively, as the input to s.

Using data from the funnel logging system, it is possible to monitor churn for each step and detect anomalies in the clearance rate metrics for specific user segments. The spikes or drops in clearance rate metrics can be an early signal of a bot attack or a bug in the verification system.

However, since our goal is be able to identify fake accounts that pass verification challenges and can be ultimately operated by attackers with a range of skills, clearance rate alone is not sufficient to fully capture the effectiveness of a set of challenges or the verification as a whole. We need further techniques which have the power to distinguish between real and manually driven fake accounts clearing verification system flows.

3.2 Label and Metric Definitions

We examine the performance of our classification models for distinguishing between fake and real accounts by comparing our classifications to *expert manually labelled accounts*. In order to establish if an account is fake, Facebook uses a team of specialists to review accounts. The reviewers look for specific signals that can indicate whether a account is real or fake, and using these signals ultimately label each account. For the

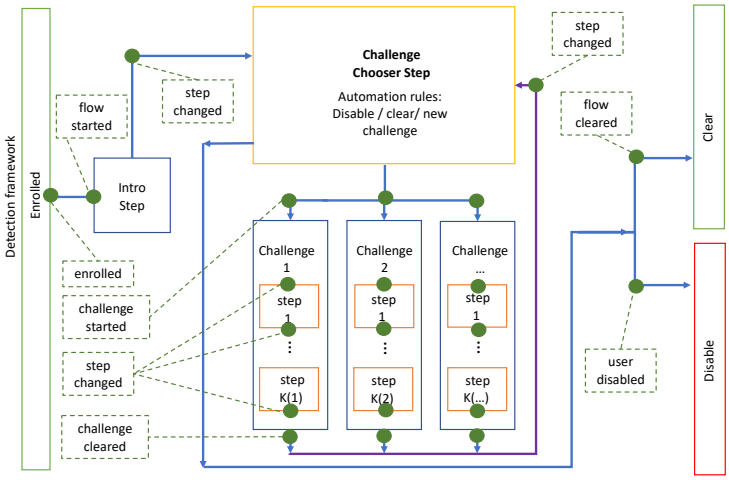

Figure 1: Flow graph showing Facebook's fake account verification system. Logging events ("funnel logging") are indicated as dots labelled with the name of the event in the dashed box.

purposes of this work we treat such labelling as *ground truth*. We define three account labels:

- **Abusive**: The account has intent to abuse the platform, including human-driven abuse.
- **Benign**: The account is authentic/real.
- **Empty**: There is *not yet* enough information to classify.

The definition of what constitutes abusive and benign behaviour is specific to the OSN. For example, at Facebook, these labels are defined by the Community Standards document [13].

Human labels are robust and reliable, but not perfect. For example, it is possible that an account's label might change over time, e.g. empty accounts could be created en masse and then sold days/weeks/months later to individuals who operate the account manually for abusive purposes.

The terms *fake* and *abusive* both refer to fake accounts. The terms *benign*, *authentic*, and *real* all refer to real users. The *prevalence* of a class $Pv(t_i,Y)$ is defined as the true proportion of accounts of class t_i in the overall population Y. Prevalence is typically measured through human labelling on a random sample of population Y, taking care to account for bias in the dataset (e.g., orders of magnitude more good than bad).

The ultimate goal of this work is to enable more rapid and computationally cost effective experiment iteration, and our strategy is to develop systems that can approximate expert human labelling. Section 4 describes several candidate models for classifying users clearing verification flows. The outputs of our models are called *proxy labels*. We evaluate our models based on the *precision* and *recall* [40] of these labels; specifically, for model m which classifies users into classes $t_1,t_2,...,t_n$, we denote the precision and recall of m for class t_i over population Y by $P(m,t_i,Y)$, $R(m,t_i,Y)$, respectively.

We also use the F_1 *score* [37] of m for class t_i, over population Y, denoted $F_1(m,t_i,Y)$. This score is defined as the harmonic mean of precision and recall:

$$F_1(m,t_i,Y) = \frac{2 \cdot P(m,t_i,Y) \cdot R(m,t_i,Y)}{P(m,t_i,Y) + R(m,t_i,Y)}$$

Both precision and recall are important for classifying users clearing verification systems. High recall across all classes is important as there is limited utility in precisely identifying authentic users if the identified set is only a small fraction of the population. This consideration is equally important for the abusive population, as we will demonstrate in Section 3.3. On the other hand, low precision is unacceptable as it could lead us to believe we are helping authentic users to clear when actually we are helping both authentic and fake users.

The F_1 score gives an overall quality indicator in cases where there is an unequal distribution of fake/real classes, and/or the relative costs of false positives and false negatives are different; both of these conditions hold in fake account verification problems.

A key insight in our examination of this space is that any model that performs better than random assignment will provide useful insight. However, higher precision and recall means we can be more confident in the model thus reducing classification time and human labelling volume. For example, a model with near perfect precision and recall could replace human labelling altogether, whereas a model that is only slightly better than chance could be used in data analysis to support hypotheses but could not be used to accurately measure the effects of changes to real or fake users.

The methods described in this work also use some time delay to accrue signal. We use *time to classification* to refer to the time delay between a user clearing the verification system and enough signals being collected for a label to be assigned.

3.3 Prior Art: BOT Classification Model

The goal of this work is to enable rapid iteration of verification challenge systems, and to that end, we require metrics to

Classification	Label	Precision	Recall	F_1 score
Bot	abusive	86%	6%	12%
Non bot	benign/empty	59%	99%	74%

Table 1: BOT model classification results for the verification system flow.

quickly assess account clearance rates with limited human labelling overhead.

Prior to this work Facebook employed a high precision bot identification model to generate proxy labels and divide users clearing the challenge into "bot" and "non-bot" classes (in addition to numerous other detection and classification systems). This model, which we denote BOT, uses as features metadata collected from fake account detection. In particular, it is often possible to detect a subset of abusive accounts through very high precision rules. When such a rule is triggered the BOT model predicts a fake account, and in all other cases it predicts a non-bot account. Because of this definition, the non-bot class can include a significant proportion of bots that were not detected by the high-precision rules. Applying this model to the clearance rate definition yields the bot proxy clearance $C_b(d_e, d_c, s, Y)$.

Given our goals and requirements, the BOT clearance rate C_b is a potentially attractive option for our proxy metric. In order to verify this hypothesis we sampled tens of thousands of accounts that successfully passed the verification system flow in August 2018 and used human labelling to find the volume of abusive, empty and benign accounts for the resulting class. Table 1 shows the label distribution over the BOT model. While $P(\text{bot}, \text{abusive})$ is fairly high, the model would be of limited value because $R(\text{bot}, \text{abusive}) = 6\%$. The majority of users that cleared the verification system flow are ambiguous, as shown by the precision of the non-bot class, $P(\text{bot}, \text{benign} \cup \text{empty}) = 59\%$. Section 5 evaluates BOT further.

The clear downside of C_b is that the non-bot class has low recall for abusive accounts. The "non-bot clearance" label is thus not accurate enough to measure verification system improvements targeted at real users. The rest of this work explores methods that better approximate human labelling ground truth, quickly, and with limited human input.

4 Post Authentication State Model

When running a large number of A/B experiments it quickly becomes prohibitively resource intensive to use human labelling to classify enough accounts clearing various challenges in each variant to get statistically significant results. Requiring expert human labellers also slows down iteration as such labelling jobs take time. A/B experiments are also often segmented by populations of interest (e.g., platform used, country, locale), which again increases the volume of necessary human labelling and reduces iteration frequency. To understand subtle changes in account clearing performance and metrics, thousands of labels are required per experiment, and possibly also for each population of interest.

In this section we present the Post Authentication State (PAS) model, a method for generating weak (i.e., likely) labels which enable rapid A/B experimentation. PAS can be scaled and is able to classify users more accurately than prior low computational cost high volume solutions (e.g., BOT classification), while allowing both faster classification and far fewer human labels than full-scale human labelling would require. PAS classifies benign users as well as abusive users, and has significantly higher recall of abusive accounts than other methods.

4.1 Overview

OSNs enroll accounts suspected of being fake into a verification system in order to gain further information about their state. The verification system needs to evolve to match the adversarial response of attackers, so OSNs need to run A/B experiments. PAS classifies accounts clearing the verification system, after a time delay, so that we can understand how the A/B experiment affected the clearance rate of each population (Figure 2). Based on results of A/B experiment OSN can evolve its response to the adversarial adaption of detected fake accounts.

PAS is a decision tree model which aims to emulate human labelling decisions, ultimately assigning an account a proxy label [4]. We denote such labels as "states." PAS is trained and validated against sets of human labelled accounts using out of the box classifiers based on the CART recursive partitioning algorithms such as SciKit Learn *DecisionTreeClassifier* [18, 34]. The model assigns one of three possible states to the classified account: *Good Post Authentication State (GPAS)* for likely real accounts with authentic signals; *Bad Post Authentication State (BPAS)* for likely fake accounts with intent to abuse the platform; *Empty Post Authentication State (EPAS)* for accounts with too little signal post-clearing to yet make a determination.

PAS predicts the labelling outcome based on signals we can automate, for example number of friends. Gupta et al. [19] showed that decision tree models, based on user level signals and behavioural signals, can be effective in classifying real and fake images in OSNs; we extend this approach to possible fake accounts clearing verification challenges in OSNs. We note that *the PAS model is not designed to be a precise classifier*; instead it buckets users clearing into "probably good" and "probably bad" which gives direction to A/B experiments with higher precision/recall.

Adversarial Adaptation: A common problem in the space of abuse detection systems is adversarial adaptation—can attackers learn what signals are used for detection, and evade them? This is not a direct concern for PAS, since this method is not used to take direct actions on accounts clearing verification system flows; rather it is used to aid in A/B experimentation

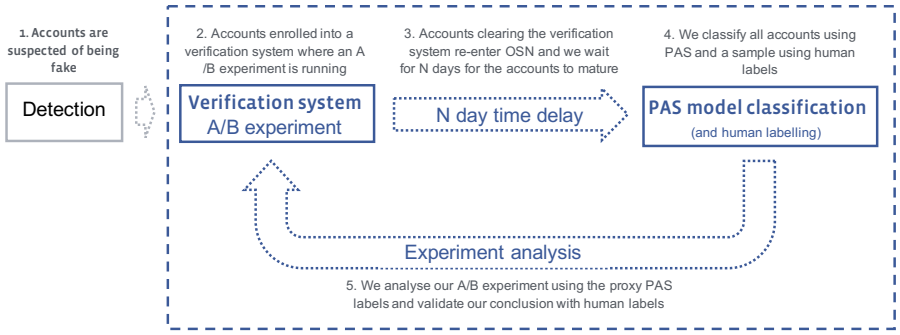

Figure 2: The PAS model as a component of the process to iterate on fake account verification systems.

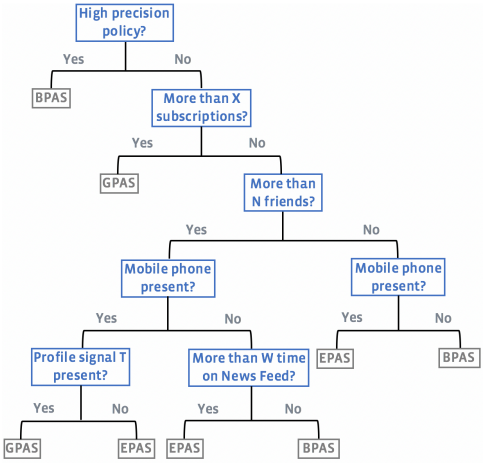

Figure 3: PAS V1 decision tree generated with recursive partitioning (CART). Accounts sent through this flow are ultimately classified with weak labels for A/B experimentation. Threshold values X, N, W, and T are operationally dependent.

and thus product evolution. This means there is no direct mechanism for adversaries to discover which signals to manipulate.

To generate proxy labels we created multiple PAS models iteratively. We started with a simple proof of concept, which showed that we could create a classifier that was better than random assignment but it had flaws in the features selection (Section 4.2). Our next iteration, still a simple proof of concept, used more robust features and was used to understand how the time to classification, or latency, could be improved (Section 4.3). Finally, we created a more accurate model, implemented it in Facebook's production verification system and showed that it could maintain good performance and allow rapid iteration of the verification system over a 6 month period (Section 4.4).

4.2 PAS V1 and PAS V2: Simple decision trees

The inputs to the PAS model are attributes and behaviours we can associate with the account. Account-level attributes include features such as the number of friends or email domain the account signed up with. Behaviours include post-clearance activity such as the number of friend requests sent or number of times other users reported the account. For each potential input, we first observed how prominent it was in each labelling population, to understand its potential impact in the construction of a decision tree.

Figure 3 depicts the first PAS decision tree we developed to classify users clearing fake account verification challenges at Facebook. This was a simple decision tree that remained static rather than being retrained. We wanted to understand how this tree performed initially and how it degraded over time. Behavioural signals such as "More than W time on News Feed" correlate to how engaged and how manual the account is, which in turn increases the likelihood that the account is a real user. We leave a specified time period post-clearing to allow these behavioural signals to accrue; it aligns with the period we use to allow labelling signals to accrue before human labelling, and so there is no decrease in the time to classification (Section 5). The specific features in this construction can vary based on OSN use case. For example, "News Feed" could be swapped for another product users engage with in other OSNs. Profile information such as "mobile phone present" could be replaced with other engagement signals such as employment status or current city.

During our evaluation of the first simple PAS model we saw a clear decline in performance of the PAS V1 model over time. This resulted from an important signal (the "high precision policy" in Figure 3) having lost its discriminating power due to changes in the prevalence of the signal in the fake population. We also identified that decision points which are also prerequisites for challenges (e.g., the "having a mobile phone number" signal is a prerequisite for the SMS challenge) create bias in A/B experimentation; since experiments that change the distribution of challenges offered would a priori skew the resulting proxy labels. As a result of these observations we developed a subsequent PAS model, PAS V2, which addressed these limitations.

PAS V2 is structured similarly to PAS V1, constructed again using CART. In this iteration, the "high precision policy" signal is replaced with signals we identified experimentally to be longitudinally stable and have high distinguishing power (Figure 4). Two new signals were added to the tree: one based on how many times the user logged in (behavioural) and one

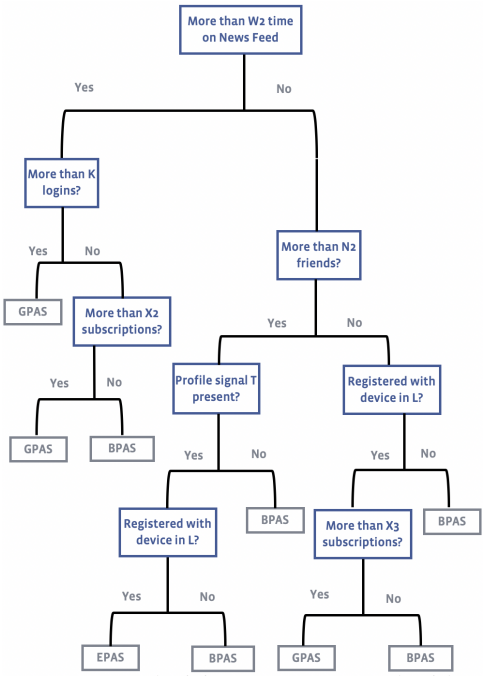

Figure 4: PAS V2 decision tree generated with recursive partitioning (CART). Accounts sent through this flow are ultimately classified with weak labels for A/B experimentation. Threshold values X_2, X_3, N_2, W_2, K, L, and T are operationally dependent.

based on the device they registered with (account attribute). The delay period post-clearing, used to allow signals to accrue and to calibrate thresholds used for signals such as "more than N friends," remained the same between the two models.

Section 5 contains a detailed evaluation of PAS V1 and PAS V2 performance.

4.3 Quick PAS: Decreasing time to classification

The simple PAS decision trees use fewer signals than the human labelling trees, and the signals are not contextual. Given this, we hypothesised the time to classification (delay post-clearing) is less critical to PAS than human labelling i.e., decreasing it would not significantly impact precision and recall.

There are two natural ways to decrease the time to classification. The first is artificially limiting the time to classification, running the same model sooner. We assessed the precision and recall of these models when run at truncated delays post-clearing; between 40% and 80% of the full time to classification before human labelling. As hypothesised, reducing the time to classification did not yield significant reductions in precision and recall, even at the shortest time to classification tested.

The second method explored to limit the time to classification was to train a new decision tree with a shorter delay post-clearing and a feature set pruned of time sensitive signals. We created "Quick PAS," a reduced-time version of PAS V2 that provides signals more than 5 times faster than PAS V2. Quick PAS has lower thresholds for behavioural signals, such as time on News Feed, and omits some of the signals that take more time to collect, such as having subscriptions. It is important to note that the trade-off in using Quick PAS is not just precision/recall; we are also biasing towards accounts that return to the platform faster than others.

Section 5 evaluates Quick PAS in the context of other PAS models. It also shows the performance of PAS V2 when time to classification is reduced by just over 50%, "Truncated PAS V2."

4.4 PAS Production: Ensemble decision tree with retraining

The simple PAS decision tree models showed promise in terms of accuracy and latency (time to classification). However, fake account detection and response is an adversarial space; attackers adapt their approach over time to try to evade detection and deceive response verification systems. The consequence is that a simple decision tree model, trained at a particular point in time, will degrade in accuracy as fake accounts evolve. Moreover, training just once makes the model vulnerable to anomalies in the training data.

The next iteration, PAS Production, was developed to address these limitations. PAS Production uses an ensemble decision tree model, to avoid overfitting; it is also retrained every day using a rolling window of training labels from the last few weeks, to retain freshness. This model uses SciKit Learn *BaggingClassifer* combined with *DecisionTreeClassifier*. Like PAS V1 and PAS V2, this model was trained with time to classification the same as the post-clearance delay to human labelling. The goal of PAS Production was to make a more accurate and reliable model, rather than a faster one. A "Quick PAS" could be developed in the same way as described in Section 4.3, by trimming the feature set and training the model with a shorter delay post-clearing.

Additionally, we explored using SciKit Learn probability outputs to gauge uncertainty of the predicted label. Averaging these probabilities for each class in each experiment group can give more signal than taking the most likely class. For example, test groups A and B might have the same number of GPAS (real account) predictions, but group A's GPAS accounts might all have higher probabilities associated with them than group B's. Averaging the probabilities would reveal this where summing class labels wouldn't. It's important to note that probability of a predicted label class can be only be interpreted as confidence of that prediction if the model is well calibrated. SciKit Learn offers calibration functions, such as *CalibratedClassifierCV*, to achieve this.

Section 5 evaluates PAS Production in the context of our other PAS models; for this purpose we restrict our analysis to class predictions and ignore the associated probabilities.

5 Evaluation

In order to assess the effectiveness of the PAS iterations, we evaluated their classification performance against human labelling data on hundreds of thousands of accounts over a period from March 2018 to May 2019. In addition to results, we also identify insights that led to further improvements throughout the evaluation.

The goal of these models is to produce *weak* labelling for use in A/B experimentation, not to produce classification for operational in-production abuse detection. Given this goal we can tolerate medium levels of precision, recall, and F_1, provided the models perform significantly better than random assignment.

Table 2 presents the results of experiments carried out for each version of the model. The table is divided into three groupings: Baseline results 1-3 (random assignment, BOT, human labelling), iterative developments 4-9 (PAS V1, PAS V2, Truncated PAS V2, Quick PAS), and current deployment 10-11 (PAS Production). The last grouping represents the final iteration of the system and shows significant decreases human labelling volume and improvements over previous models.

5.1 Baselines: Random Assignment, BOT, and Human Labelling

Since we take human labelling to be our ground truth, human labelling provides the benchmark and optimal result for models intending to classify users clearing our verification system (Table 2, Row 3). If we classified users with random assignment, then recall would be 1/3 for each class and precision would be the prevalence of that class in the population of accounts sent for verification (Table 2, Row 1). Random assignment provides a lower bound to compare models against; any model with lower precision and/or recall than random assignment would be detrimental in evaluating experiments.

The BOT model provides a second comparison point. This model uses a high precision signal available from detection to classify users as fake. The signal used is a binary signal which predicts an account to be fake (or BPAS), if it exists for the account. It cannot predict whether an account is authentic (GPAS) or empty (EPAS). Table 2, Row 2 provides the precision, recall, and F_1 scores for BOT. As a result of the signal existing prior to the account clearing fake account verification systems, there is no time delay needed to use it for prediction. We observe that the BOT model's BPAS precision is high, at 86%, but its recall and thus F_1 are low at 6% and 12% respectively. Given the low recall for BPAS and its inability to distinguish the other two classes, we cannot use this model for weak labelling. We require a model that predicts both fake and authentic users because our experiments are designed to prevent fake users from clearing verification systems and help authentic users to do so. Moreover, low recall for fake users means that this model is not suitable for even the subset of experiments that try to prevent fake accounts from clearing, because it is able to classify too few of them.

5.2 PAS V1

Table 2, Row 4 shows the performance of PAS V1 in March 2018, during its first iteration. The PAS V1 decision tree performed better than random assignment in terms of both precision and recall and was an initial improvement in classification. EPAS ("empty"), the proxy label for accounts with too little signal to mark as authentic or fake, had the poorest precision and recall but represents the population of accounts we are less motivated by in this use case—our primary objectives are to help increase authentic user clearance (GPAS) and decrease clearance of abusive users (BPAS). PAS V1 has a much better precision-recall trade-off for abusive accounts than the bot/non-bot classification. We did not measure the decrease in human labelling as the limitations of PAS V1 necessitated PAS V2.

Table 2, Row 5 shows the performance for PAS V1 in June 2018, three months after implementation. The precision of benign classifications decreased significantly, from 76% to 25%, and recall across both abusive and empty classifications also similarly decreased. F_1 scores dropped for all classes. As discussed in Section 4.2, the "high precision policy" signal had lost its discriminating power due to changes unrelated to our work. These changes motivated the design of PAS V2.

5.3 PAS V2

Table 2, Row 6 shows PAS V2 performance in July 2018, when it was first evaluated. Compared to the degraded scores of PAS V1 from June 2018, PAS V2 shows large improvement in F_1 scores for all classes. In comparison to the initial PAS V1, F_1 score increased for BPAS class and decreased for GPAS classes. Additionally, we've observed that more of its signals have stable distribution over time.

To explore the stability of the system, we reran the evaluation of PAS V2 in September 2018, several months after it was first implemented (Table 2, Row 7). Unlike PAS V1, we did not notice a substantial reduction in performance over time. The main change was that the F_1 score for abusive accounts dropped from 72% to 53%, primarily from abusive precision dropping from 66% to 42%. The drop is caused by changes in the abusive clearance population; fewer accounts were being labelled as abusive, and more were labelled as empty—potentially due to attackers choosing to let accounts "sleep" in response to concurrent, independent work on improved detection.

PAS V2 does not have the same issues as PAS V1 with respect to signals that can be skewed by the verification system itself and none of the underlying signals changed in definition. However the reduction in abusive precision highlights the fact it is necessary to monitor and retrain the PAS decision

Row	Method	Time Period	BPAS Abusive			GPAS Benign			EPAS Empty			Decrease Class. Time	Decrease Human Label Vol.
			Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1		
1	Rand. Assign.	Sep 2018	33%	33%	33%	25%	33%	28%	42%	33%	36%	–	–
2	BOT	Aug 2018	86%	6%	12%	–	–	–	–	–	–	–	–
3	Human Label.	All	100%	100%	100%	100%	100%	100%	100%	100%	100%	0%	0%
4	PAS V1	Mar 2018	65%	61%	63%	76%	78%	77%	47%	51%	49%	0%	–
5	PAS V1	Jun 2018	74%	32%	45%	25%	82%	38%	40%	32%	36%	0%	–
6	PAS V2	Jul 2018	66%	80%	72%	53%	64%	58%	76%	35%	48%	0%	70%
7	PAS V2	Sep 2018	42%	70%	53%	57%	62%	59%	79%	33%	46%	0%	70%
8	PAS V2 Trunc.	Jul 2018	63%	77%	70%	52%	60%	56%	73%	36%	48%	56%	–
9	Quick PAS	Jul 2018	61%	76%	68%	59%	36%	45%	55%	45%	50%	81%	70%
10	PAS Production	Nov 2018	73%	61%	66%	71%	71%	71%	78%	86%	82%	0%	70%
11	PAS Production	May 2019	68%	62%	65%	61%	61%	61%	74%	81%	78%	0%	70%

Table 2: Comparison of PAS models broken down by classification method and validated against human labelling. The first grouping of rows shows idealised and prior methods. The second grouping shows results of intermediate techniques. The third grouping shows results of the final design.

tree model at regular intervals to mitigate risks of changing behaviours in the clearance population.

5.4 Truncated PAS V2 and Quick PAS

To verify our hypothesis about the trade-offs associated with shortened post-clearing delay (Section 4.3), Table 2 (Row 8) measures performance of PAS V2 after truncating the post-clear calculation delay by just over 50%. Compared with PAS V2 evaluated over the same period, the performance of Truncated PAS V2 is only very slightly lower for each class. This experiment confirmed that the post-clearing delay can be reduced without compromising accuracy, which allows us to introduce lower thresholds for behavioural signals and train a decision tree optimised for those changed thresholds and shortened delay. Such changes were codified (beyond a simple reduced threshold) into Quick PAS (Section 4.3).

Table 2, Row 9 shows the performance of Quick PAS in July 2018. Quick PAS has lower F_1 scores in all classes compared to PAS V2. However, benign recall drops and empty recall increases, since the reduced time window limits our ability to collect authentic engagement signals which would ultimately disambiguate an "empty" account from benign for expert human labellers.

5.5 PAS Production

Table 2, Row 10 shows the performance of PAS Production in November 2018. PAS Production strikes the best performance balance between classes: it is the only model to have F_1 scores above 60% for every class. In particular, the Empty (EPAS) F_1 score is much higher than other models, 82% compared with 50% or less from previous models, due to increased recall. This could be a result of the retraining, allowing thresholds to adapt. The Benign (GPAS) F_1 score is also higher than PAS V2's, 71% compared with 59% or less, due to increased precision. This could be a result of using an ensemble model and not overfitting on the training data. The Abusive (BPAS) F_1 score is slightly lower that the F_1 score of PAS V2 when it was first developed, 66% compared with 72%. However, this is a much smaller drop than the gain in accuracy for the other two classes and still much higher than random assignment, so we find this acceptable. To verify our hypothesis that PAS Production is more robust than previous PAS models that didn't retrain, we reran the evaluation of PAS Production six months later in May 2019 (Table 2, Row 11). The precision, recall and F_1 scores of all three classes remained above 60%. The largest drop was for the F_1 score of the Benign (GPAS) class, from 71% to 61%, changing equally in precision and recall. These drops might result from attackers increasing their efforts to appear real over those six months, as far as the automatable signals used in PAS can tell. Our human labelling process relies on more signals, some of which are contextual, and it adapts over time. We are still confident that our human labels represent ground truth.

Our ensemble decision tree, PAS Production, which was been implemented to retrain daily, shows more consistent performance between the three classes and more robustness to time compared with previous models. It has the same time to classification as the labelling process. A lower latency "Quick PAS Production" could be developed to complement PAS Production, to provide earlier signal for A/B experiments.

5.6 PAS Impact

We integrated PAS Production into Facebook's environment to assess their usefulness in the experimentation process. When a change was introduced into a verification system through an experiment, we used PAS to understand how the change impacted real and abusive accounts clearing the system. In order to understand how experiments impact how accounts flow within a verification system, we used funnel logging event aggregations within challenges to identify the number of accounts attempting and passing challenges, and the time taken. We used the proxy labels assigned by the PAS models, combined with the funnel logging metrics, to support

or refute our hypotheses. If the proxy labels and the additional metrics supported the experiment hypothesis, we would then supplement with additional human labelling to validate results before launching the change.

Decreased Classification Time: Quick PAS showed that we are able to get directional signal on experiments with significant reductions in the time to classification that human labelling requires. This early signal enables us to stop failing experiments earlier or request human labelling validation so that we can launch a change sooner. Quick PAS decreased classification time by 81% whilst keeping accuracy for each class well above random assignment.

Decreased Human Labelling Volume: As outlined in Section 3, the purpose of an OSN fake account verification system is to block access for abusive accounts; and permit benign accounts to re-enter the OSN. Experiments on verification systems will aim to achieve one of these objectives, without harming the other. It is thus necessary to understand how an experiment affects each population and not rely on just the overall clear rates. For example, without further breakdown, an increase in the volume of accounts clearing the verification system cannot be interpreted as achieving the objective of helping benign accounts; as these incremental accounts might be overwhelmingly abusive. A significant amount of labels are required to understand the effects of an experiment at different stages. Accounts have to be labelled early to catch failing experiments sooner. In addition, accounts clearing in subsequent days have to be labelled to mitigate effects of selection bias of the early-stage labelling. Finally, labelling may be required to measure adversarial response several weeks after shipping a feature, using a holdout.

Using the Wald method of binomial distribution, in order to estimate the proportion of accounts in each group that are abusive, benign and empty, to within a 5% error bound, we would need 400 labels per group. Doing this several times per experiment, for multiple experiments per week, would mean tens of thousands of labels are required each week. Human labels are a scarce resource and can't be scaled to support experiments. Pairing the PAS model-produced proxy labels with just one set of validation human labels per experiment, for only those experiments we believe are successful, reduces total human label volume. This method saves early-stage labels on all experiments and it saves all label requirements in clearly negative experiments; as PAS proxy labels give this information. We evaluated labelling volume from July to May 2019. Over this period, Facebook launched and analysed more than 120 experiments. In total, 20,000 human labels were required to be confident about shipping iterations to the fake account verification system. Facebook saved an estimated 50,000 human labels that would have otherwise been required to monitor these experiments. PAS models reduced the volume of human labelling required for experiment analysis by 70% (Table 2). Additionally, as each of the launched experiments required substantially fewer labels, Facebook could run many more experiments in parallel.

Adversarial Adaptation: In addition to improving efficiency, the models were successfully used for automated monitoring in backtests of launched features. With this framework, Facebook discovered three cases in which the adversaries eventually adapted to the new feature, which would manifest itself as a shift in BPAS prevalence in the population exposed to that feature. This measurement allowed the team working on the verification system to quickly discover the underlying reasons for adaptation and mitigate the problem appropriately.

6 Conclusion and Future Work

We have presented a method for evaluating changes to fake account verification systems, the Post Authentication State (PAS) method. PAS uses a continuously retrained machine-learned ensemble decision tree model that proxies human labelling to classify accounts as abusive and benign faster and with less human labelling than prior approaches. PAS can be used to measure the effectiveness of changes in a verification system over time and to analyse A/B experiments which aim to prevent abusive accounts clearing or help benign accounts to clear the system. At Facebook, PAS reduced the volume of human labelling required for experiment analysis by 70% and decreased the classification time of accounts by 81%. The presented method achieved precision over 70% and recall over 60% for all three classes. PAS has allowed Facebook engineering and data science teams to iterate faster with new features for verification challenges, scale experimentation launch and analysis, and improve the effectiveness of verification systems at remediating fake accounts.

In this paper we have mentioned that *fake account* is a generic term that can cover several types of abusive accounts; a high-level taxonomy would be bots and manually driven fake accounts (MDFA). Being able to further divide our abusive labels and further divide BPAS (our proxy label) into abusive bot and abusive MDFA would greatly help to optimise challenge selection in a verification system. For example, there could be challenges that are trivial for humans and difficult for bots (e.g., a well designed CAPTCHA), and there could be challenges that may be solved by bots but deter humans (e.g., a time-consuming verification). If we were able to measure whether a fake account was a bot or a MDFA then we could assign challenges appropriately.

Finally, we note that our implementation and experiments use the data and infrastructure of a single large online social network, Facebook, and therefore the experimental results might be different for other OSNs. We encourage the research community to apply our approach more broadly to determine to what extent the results and conclusions we have presented in this paper transfer to other areas.

References

[1] Noura Alomar, Mansour Alsaleh, and Abdulrahman Alarifi. Social authentication applications, attacks, defense strategies and future research directions: a systematic review. *IEEE Communications Surveys & Tutorials*, 99, 2017.

[2] Alessandro Bessi and Emilio Ferrara. Social bots distort the 2016 us presidential election online discussion. 2016.

[3] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. The socialbot network: when bots socialize for fame and money. In *ACM CCS*, 2011.

[4] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[5] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, pages 197–210, 2012.

[6] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 477–488, 2014.

[7] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Who is tweeting on twitter: human, bot, or cyborg? In *ACM CCS*, 2010.

[8] Nicholas Confessore, Gabriel J.X. Dance, Richard Harris, and Mark Hansen. The follower factory. *The New York Times*, 01 2018.

[9] Mauro Conti, Radha Poovendran, and Marco Secchiero. Fakebook: Detecting fake profiles in on-line social networks. In *Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2012.

[10] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, pages 1–15. San Diego, CA, 2009.

[11] Facebook. Community standards enforcement preliminary report, 2018.

[12] Facebook. Facebook reports second quarter 2018 results, 2018.

[13] Facebook. Community standards - integrity and authenticity, 2019.

[14] Nicholas Fandos and Kevin Roose. Facebook identifies an active political influence campaign using fake accounts. *The New York Times*, 07 2018.

[15] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 2016.

[16] M. Fire, G. Katz, and Y Elovici. Strangers intrusion detection - detecting spammers and fake profiles in social networks based on topology anomalies. *ASE Human Journal*, 2012.

[17] Hongyu Gao, Jun Hu, Tuo Huang, Jingnan Wang, and Yan Chen. Security issues in online social networks. *IEEE Internet Computing*, 2011.

[18] Raúl Garreta and Guillermo Moncecchi. *Learning scikit-learn: machine learning in python*. Packt Publishing Ltd, 2013.

[19] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy. In *World Wide Web Conference (WWW)*, 2013.

[20] JingMin Huang, Gianluca Stringhini, and Peng Yong. Quit playing games with my heart: Understanding online dating scams. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2015.

[21] Rodrigo Augusto Igawa, Sylvio Barbon Jr, Kátia Cristina Silva Paulo, Guilherme Sakaji Kido, Rodrigo Capobianco Guido, Mario Lemes Proença Júnior, and Ivan Nunes da Silva. Account classification in online social networks with lbca and wavelets. *Information Sciences*, 332:72–83, 2016.

[22] Anil K Jain, Karthik Nandakumar, and Arun Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79:80–105, 2016.

[23] Jing Jiang, Christo Wilson, Xiao Wang, Wenpeng Sha, Peng Huang, Yafei Dai, and Ben Y. Zhao. Understanding latent interactions in online social networks. *ACM Trans. Web*, 7(4), November 2013.

[24] Lei Jin, Hassan Takabi, and James B.D. Joshi. Towards active detection of identity clone attacks on online social networks. In *ACM Conference on Data and Application Security and Privacy*, 2011.

[25] Sam King. Stopping fraudsters by changing products, 2017.

[26] David Koll, Martin Schwarzmaier, Jun Li, Xiang-Yang Li, and Xiaoming Fu. Thank you for being a friend: an attacker view on online-social-network-based sybil defenses. In *Distributed Computing Systems Workshops (ICDCSW)*, 2017.

[27] Martin Kopp, Matej Nikl, and Martin Holena. Breaking captchas with convolutional neural networks. In *CEUR Workshop Proceedings*, volume 1885, pages 93–99, 2017.

[28] Kate Lamb. "i felt disgusted": inside indonesia's fake twitter account factories. *The Guardian*, 07 2018.

[29] Anshu Malhotra, Luam Totti, Wagner Meira Jr, Ponnurangam Kumaraguru, and Virgilio Almeida. Studying user footprints in different online social networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 1065–1070. IEEE Computer Society, 2012.

[30] Merylin Monaro, Luciano Gamberini, and Giuseppe Sartori. The detection of faked identity using unexpected questions and mouse dynamics. *PloS one*, 12(5):e0177851, 2017.

[31] Romklau Nagamati and Miles Lightwood. Audio challenge for providing human response verification, 2015. US Patent 8,959,648.

[32] Palash Nandy and Daniel E Walling. Transactional visual challenge image for user verification, 2008. US Patent App. 11/679,527.

[33] Avanish Pathak. An analysis of various tools, methods and systems to generate fake accounts for social media. Technical report, Northeastern University, Boston, Massachusetts, December 2014.

[34] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[35] Iasonas Polakis, Marco Lancini, Georgios Kontaxis, Federico Maggi, Sotiris Ioannidis, Angelos D Keromytis, and Stefano Zanero. All your face are belong to us: breaking facebook's social authentication. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 399–408, 2012.

[36] David Press. Fighting financial fraud with targeted friction, 2018.

[37] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 2nd edition, 1979.

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

[39] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 2017.

[40] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 2009.

[41] Saumya Solanki, Gautam Krishnan, Varshini Sampath, and Jason Polakis. In (cyber) space bots can hear you speak: Breaking audio captchas using ots speech recognition. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 69–80. ACM, 2017.

[42] David J Steeves. Client-side captcha ceremony for user verification, 2012. US Patent 8,145,914.

[43] Tao Stein, Erdong Chen, and Karan Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, page 8. ACM, 2011.

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

[45] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference*, pages 1–9. ACM, 2010.

[46] Gianluca Stringhini, Pierre Mourlanne, Gregoire Jacob, Manuel Egele, Christopher Kruegel, and Giovanni Vigna. Evilcohort: Detecting communities of malicious accounts on online services. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 563–578, 2015.

[47] Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang, and Yihong Zhao. Unik: Unsupervised social network spam detection. In *ACM International Conference on Conference on Information & Knowledge Management*, 2013.

[48] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: an analysis of Twitter spam. In *ACM Internet Measurement Conference (IMC)*, 2011.

[49] Twitter. Investor fact sheet. q2 2018 highlights, 2018.

[50] Erkam Uzun, Simon Pak Ho Chung, Irfan Essa, and Wenke Lee. rtcaptcha: A real-time captcha based liveness detection system. In *NDSS*, 2018.

[51] Onur Varol, Emilio Ferrara, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characterization. *arXiv preprint arXiv:1703.03107*, 2017.

[52] Bimal Viswanath, Muhammad Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *USENIX Security*, 2014.

[53] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. Captcha: Using hard ai problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 294–311. Springer, 2003.

[54] Alex Hai Wang. Don't follow me: Spam detection in twitter. In *2010 international conference on security and cryptography (SECRYPT)*, pages 1–10. IEEE, 2010.

[55] Cao Xiao, David Mandell Freeman, and Theodore Hwa. Detecting clusters of fake accounts in online social networks. In *ACM Workshop on Artificial Intelligence and Security*, 2015.

[56] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y. Zhao, and Yafei Dai. Uncovering social network sybils in the wild. *ACM Trans. Knowl. Discov. Data*, 8(1):2:1–2:29, February 2014.

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

[58] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36(4):267–278, 2006.

[59] Chao Michael Zhang and Vern Paxson. Detecting and analyzing automated activity on twitter. In *International Conference on Passive and Active Network Measurement*, pages 102–111. Springer, 2011.

[60] Binbin Zhao, Haiqin Weng, Shouling Ji, Jianhai Chen, Ting Wang, Qinming He, and Reheem Beyah. Towards evaluating the security of real-world deployed image captchas. In *ACM Workshop on Artificial Intelligence and Security*, 2018.