BlueSky Scam Labeler

CS 5342: Trust and Safety: Platforms, Policies, Products

Members: Konrad Kopko kk2239, Yang Ji yj586, Alex Xiong hx329, Hanqi Guo hg493

Github Repository Link: BlueSky Labeler

Video Link + Slides Link: Video Slides

Motivation

Decentralized platforms like Bluesky face a "cold start" problem in Trust and Safety compared to mature, centralized systems [1]. Bluesky's open architecture attracts malicious actors who use URL scams and crypto fraud, often leveraging AI-generated content and mass-following tactics [2]. While Bluesky effectively moderates hate speech, malicious URLs remain a vulnerability difficult to filter at the protocol level without risking censorship. We pivoted from hate speech to URL scams because Bluesky already moderates the former effectively, whereas malicious URLs remain a vulnerability difficult to filter without risking censorship. These scams cause financial loss and erode trust. Phishing uses social engineering to steal credentials, and decentralized environments heighten impersonation risks where custom domains may be mistaken for verification [3]. Research suggests blacklists alone are reactive; however, Cao et al. (2015) showed *behavioral analysis* (e.g., follower ratios) detects malicious accounts with up to 86% accuracy [4]. For our research question, we want to investigate whether a client-side labeler can mitigate URL scams by combining content heuristics with behavioral metrics. Our "Potential URL Scam" labeler empowers users without relying on central censorship.

Policy Design

This proposal outlines the design and implementation of PolicyLabeler, an automated moderation tool that detects and labels potential URL scams in the Bluesky "Firesky" feed. Unlike traditional binary moderation (keep/delete), our approach uses the AT Protocol's labeling framework to contextualize risk for users, preserving BlueSky's decentralized ethos of user choice. The PolicyLabeler aggregates risk signals across three primary dimensions, informed by the literature on social media abuse:

1. **Profile Heuristics:** Leveraging research on bot detection, we analyze the author's social graph. Accounts with extreme following-to-follower ratios (indicating mass-following scripts), low follow-back rates, or high posting volumes with zero audience engagement are flagged as high-risk.

2. **Content Semantics:** The system scans for "urgency cues" and high-pressure sales language (e.g., "guaranteed profit", "act now"), excessive emoji usage (a common obfuscation tactic in spam), and hashtag flooding.

3. **URL Analysis:** We implement a dual-layer URL check that flags known malicious domains (via the `malicious_phish.csv` dataset) and the use of URL shorteners (e.g., bitly, tinyurl, etc.), which are frequently used to mask phishing destinations.

Our implementation assigns a weighted "scam score" to each post containing a URL. If the cumulative score exceeds a threshold of 5 points, indicating convergence between suspicious profile behavior and malicious content characteristics, the post is assigned the "Potential URL scam post" label. This summary proposal demonstrates that a heuristic-based, transparent labeling system can effectively reduce the visibility of low-effort scams and protect user safety without requiring invasive surveillance or centralized takedowns.

Technical Approach

Our code implements a `PolicyLabeler` class that evaluates Bluesky posts to determine whether they should be flagged as a “Potential URL scam post.” When given a post URL, it retrieves the post and runs several heuristic checks on both the author and the content.

```
(1) scam_checks += self.check_profile_for_potential_scam(post)
(2) scam_checks += self.check_post_for_emojis(post)
(3) scam_checks += self.check_post_for_sus_language(post)
```

```
(4) scam_checks += self.check_post_for_malicious_urls(post)
(5) scam_checks += self.check_post_for_shortened_urls(post)
(6) has_url = self.check_post_for_any_url(post)
```

These functions examine the author's profile for characteristics typical of scam or bot accounts (1), such as extreme following-to-follower ratios, unusually high posting volume with very few followers, and low follow-back rates.It analyzes the post's text for excessive emojis (2), scam-related language from a CSV list of suspicious phrases (3), and heavy hashtag usage (3).The system extracts all URLs in the post, including those embedded in Bluesky facets (4), and checks whether any matching domains are on a known malicious phishing list (4) or are standard URL-shortening services (5).If the combined "scam score" from these checks reaches a threshold (1–5) and the post contains at least one URL (6), the method labels it as a potential scam; otherwise, it produces no label.

Test Design

We evaluated the labeler using a test set of 154 posts, each manually annotated so we could directly compare the expected label with the system's output. Our testing design intentionally combines 34 real Bluesky posts with 120 synthetic samples to ensure both authenticity and comprehensive coverage of scam patterns. Real posts capture natural user behavior, linguistic variation, and platform-specific context, allowing us to assess the labeler's performance under realistic conditions. In contrast, synthetic posts allow us to systematically construct edge-case scenarios, such as emoji-heavy scam messages, deliberately ambiguous wording, or rare phishing strategies, that are difficult to obtain in large numbers from real data but are essential for testing robustness.This combination offers broad coverage of scam variations and ensures that the 154 posts serve as a meaningful stress test.

Test Results and Analysis

The PolicyLabeler achieved **151 out of 154** correct classifications, corresponding to an accuracy of **0.98**. Only three posts were misclassified: *(1) a scam disguised as legitimate travel advertising that did not trigger suspicious-language rules; (2) a Telegram sexual scam hidden behind playful "dog / bark / playhouse" wording that bypassed semantic detection; and (3) a scam URL that was not included in the malicious_phish.csv dataset and therefore failed to reach the scoring threshold.* The results show that the system design is generally effective. By combining profile data, content semantics, and URL analysis into a weighted scoring mechanism, rather than relying on any single feature, the PolicyLabeler maintains stable performance across different scam patterns without being overly restrictive or producing excessive false negatives. However, several gaps remain. The current model evaluates posts individually and, beyond basic account data, does not analyze an account's broader posting style or behavioral patterns. This contributed to the first two misclassifications, where metaphorical or advertising-like language allowed scam content to evade detection. In addition, the phishing-URL list is not dynamically updated, causing newly emerging malicious domains to be unrecognized, as seen in the third error. Incorporating account-history analysis and automatically updating the malicious URL database would reduce such misclassifications.

Future Work

If we continue working on the project, we aim to increase the labeler's real-world effectiveness and expand beyond URL scams. The first thing we would like to do is implement an account age checker to handle scenarios where new accounts may not have insane follower/following ratios or many posts, allowing them to operate with seemingly "normal" metrics without being detected. Building on this, we will also introduce account history behavior analysis and automatically update the malicious URL database, which will help reduce false positives in the current test. Secondly, to expand past just URL scams, implementing image analysis would address a critical blind spot in our current text-only approach. As we've learned in class, deepfakes and AI-generated images are only on the rise, and scammers can also use this to their advantage. Images containing fake profit screenshots, malicious QR codes, or images with text will evade our current model. Lastly, developing dynamic phrase databases akin to an LLM that learns and adapts to emerging scam language patterns. This would ensure our system remains effective as scammers evolve their tactics. Being able to distinguish between known scam phrases and novel variations enables adaptation without human intervention. This would be valuable for detecting multilingual scams and culturally-specific fraud tactics that traditional keyword matching cannot capture.

References

[1] Y. Roth, "Decentralized Platforms Struggle with Trust and Safety," *WebProNews*, 2024.

[2] "Fake Crypto Accounts Push Scams on Bluesky," *Open Measures*, Feb 2025. [Online]. Available: https://openmeasures.io/fake-crypto-accounts

[3] "As Bluesky surpasses 20 million users, beware the fake accounts," *Mashable*, Nov 2024.

[4] C. Cao, J. Caverlee, and K. Lee, "Detecting Spam URLs in Social Media via Behavioral Analysis," *Proceedings of ECIR*, 2015.

[5] Patricia S. Abril and Robert Plant. 2007. The patent holder's dilemma: Buy, sell, or troll?. ACM 50, 1 (Jan. 2007), 36-44. https://doi.org/10.1145/1188913.1188915