Towards Automatic Real Time Identification of Malicious Posts on Facebook

Prateek Dewan, Ponnurangam Kumaraguru
Indraprastha Institute of Information Technology, Delhi (IIIT-D)
Cybersecurity Education and Research Centre (CERC), IIIT-Delhi
Email: {prateekd,pk}@iiitd.ac.in

***Abstract*—Online Social Networks (OSNs) witness a rise in user activity whenever a news-making event takes place. Cyber criminals exploit this spur in user-engagement levels to spread malicious content that compromises system reputation, causes financial losses and degrades user experience. In this paper, we characterized a dataset of 4.4 million public posts generated on Facebook during 17 news-making events (natural calamities, terror attacks, etc.) and identified 11,217 malicious posts containing URLs. We found that most of the malicious content which is currently evading Facebook's detection techniques originated from third party and web applications, while more than half of all legitimate content originated from mobile applications. We also observed greater participation of Facebook pages in generating malicious content as compared to legitimate content. We proposed an extensive feature set based on entity profile, textual content, metadata, and URL features to automatically identify malicious content on Facebook in real time. This feature set was used to train multiple machine learning models and achieved an accuracy of 86.9%. We performed experiments to show that past techniques for spam campaign detection identified less than half the number of malicious posts as compared to our model. This model was used to create a REST API and a browser plug-in to identify malicious Facebook posts in real time.**

I. Introduction

Social network activity rises considerably during events that make the news, like sports, natural calamities, etc. [31]. For example, the 2014 FIFA World Cup final inspired more than 618,000 tweets per minute, a new record for Twitter. Facebook also saw 350 million users generating over 3 billion posts, comments and likes during the 32 days of the world cup. [1] This enormous magnitude of activity during sports and other news-making events makes OSNs a lucrative venue for malicious entities to seek monetary gains and compromise system reputation. Facebook, being the most preferred OSN for users to get news [20], is potentially the most attractive platform for malicious entities to launch cyber-attacks. Recently, cyber criminals exploited the context of various news events to spread hoaxes and misinformation on Facebook, luring victims into scams, phishing attacks, malware infections, etc. [21], [35] It has been claimed that Facebook spammers make $200 million just by posting links [32]. Such activity not only degrades user experience but also violates Facebook's terms of service. Facebook has acknowledged spam and hoaxes as a serious issues, and taken steps to reduce malicious content in users' newsfeed [23], [24].

Researchers have used various supervised learning models to detect spam and other types of malicious content on OSNs and achieved good results [3], [15]. However, existing approaches to detect malicious content on other OSNs like Twitter, YouTube etc. cannot be directly ported to Facebook because these approaches heavily rely on features that aren't publicly available from Facebook. These features include profile and network information, age of the account, total number of messages posted, social connections, etc.

In this paper, we address the problem of automatic real-time detection of malicious content generated during news-making events, that is currently evading Facebook's detection techniques [28]. To this end, we collect 4.4 million public posts generated by 3.3 million unique entities during 17 news-making events that took place between April 2013 and July 2014. We first study the effectiveness of existing techniques used by Facebook to counter malicious content. Then, we identify some key characteristics of malicious content spread on Facebook, which distinguish it from legitimate content. We propose an extensive feature set consisting of 42 features to automatically distinguish malicious content from legitimate content in real time. This feature set is used to train multiple machine learning models to identify malicious posts on Facebook, and attains a maximum accuracy of 86.9% using the Random Forest classifier. Our experiments show that prior clustering based spam detection techniques are able to detect less than half the number of malicious posts as compared to our model. We use our model to deploy a publicly available REST API [2] and a browser plug-in, that can be used to identify malicious content on Facebook in real time. Our broad contributions are as follows:

- Characterization of malicious content generated on Facebook during news-making events. Our dataset of 4.4 million public posts is one of the biggest datasets of Facebook posts in literature.
- Extensive feature set for identifying malicious content in real time, excluding features like *likes*, *comments*, *shares*, etc. which are absent at post creation time.
- Publicly available end-user solution (API and browser plugin) to identify malicious posts in real time.

II. Related work

Detection of malicious content on Facebook: Gao et al. presented an initial study to quantify and characterize spam

[1] http://edition.cnn.com/2014/07/14/tech/social-media/world-cup-social-media/

[2] http://multiosn.iiitd.edu.in/fbapi/endpoint/?fid=⟨post_id⟩

campaigns launched using accounts on Facebook [13]. They studied a large anonymized dataset of 187 million asynchronous "wall" messages between Facebook users, and used a set of automated techniques to detect and characterize coordinated spam campaigns. Authors detected roughly 200,000 malicious wall posts with embedded URLs, originating from more than 57,000 user accounts. Following up their work, Gao et al. presented an online spam filtering system that could be deployed as a component of the OSN platform to inspect messages generated by users in real time [12]. Their approach focused on reconstructing spam messages into campaigns for classification rather than examining each post individually. They were able to achieve a true positive rate of slightly over 80% using this technique on their dataset of 187 million Facebook wall posts. Authors reported an average throughput of 1,580 messages/sec and an average processing latency of 21.5ms. However, the clustering approach used by authors always marked a new cluster as legitimate and was unable to detect malicious posts if the system had not seen a similar post before. We overcome this drawback by eliminating dependency on post similarity and using classification instead of clustering.

In an attempt to protect Facebook users from malicious posts, Rahman et al. designed a social malware detection method which took advantage of the social context of posts [25]. Authors were able to achieve a maximum true positive rate of 97%, using a SVM based classifier trained on 6 features. The classifier required 46ms to classify a post. This model was then used to develop MyPageKeeper [3], a Facebook app to protect users from malicious posts. Similar to Gao et al's work [13], this work was also targeted at detecting spam campaigns, and relied on message similarity features. Such techniques are efficient in detecting content which they have seen in the past, for example, campaigns. However, none of the these techniques are capable of detecting malicious posts in real time, which their systems haven't seen in the past.

Detection of malicious content on other OSNs: Multiple machine learning models have been proposed in the past to detect malicious content on other social networks such as Twitter and YouTube [3], [15], [33]. The efficiency of such models comes from features like age of the account, number of social connections, past messages of the user, etc. [3], which are not available on Facebook publicly. Other techniques make use of OSN specific features like user replies, user mentions, retweets (Twitter) [15], views and ratings (YouTube) [4], which are absent in Facebook. Blacklists have been shown to be ineffective, capturing less than 20% URLs at zero-hour [26].

For efficient real time detection of malicious posts on Facebook, our approach utilizes various Facebook-specific features (for example, page category, and post type) which are publicly available at post creation time. This approach also eliminates any dependency on post similarity.

III. Methodology

There exists a wide range of malicious content on OSNs today. These include phishing, advertising campaigns, content originating from compromised profiles, artificial reputation gained through fake likes, etc. We do not intend to address all such attacks. We focus our analysis on identifying posts with a malicious URL and creating automated means to detect such posts in real time, without looking at the landing pages of the URLs. We emphasize on not visiting the landing pages of URLs since this process induces time lag and increases the time taken by real-time systems to make a judgment on a post. For the scope of this work, we refer to a post as malicious if it contains one or more malicious URLs.

A. Data collection

We collected data using Facebook's Graph API Search endpoint [11] during 17 news-making events that took place between April 2013 and July 2014. We used event specific terms for each of the 17 events (see Table I) to collect relevant public *posts*. All events we picked for our analysis, made headlines in international news. To maintain diversity, we selected events covering various domains of news events like political, sport, natural hazards, terror strikes and entertainment news. For all 17 events, we started data collection from the time the event took place, and stopped about two weeks after the event ended.

A drawback of the search method is that if a post is deleted or removed (either by the user herself, or by Facebook) before our data collection module queries the API, it would not appear in the search results. We repeated the search every 15 minutes to overcome this drawback as much as possible. Table II shows the descriptive statistics of our final dataset.

B. Labeled dataset creation

To create a labeled dataset, we filtered out all posts containing one or more URLs. These URLs were added to the set of URLs present in the *link* field (if available) for each post. These extracted URLs were then visited using Python's Requests package. [4] In case the Requests package failed, the URLs were visited using LongURL API. [5] Visiting the landing pages of the URLs helped us to eliminate invalid URLs and capture the correct final destination URLs corresponding to shortened URLs, if any. After the extraction and validation process, we were left with a total of 480,407 unique URLs across 1,222,137 unique posts (Table II). Each URL was then subjected to six blacklist lookups, viz. SURBL [30], Google Safebrowsing [14], PhishTank [22], VirusTotal [19], SpamHaus [27], and Web of Trust [34] in October 2014. This methodology of identifying malicious content using URL blacklists has also been used multiple times in past research [7], [13].

TABLE II. Descriptive statistics of complete dataset collected over April 2013 - July 2014.

Unique posts	4,465,371
Unique entities	3,373,953
- Unique users	2,983,707
- Unique pages	390,246
Unique URLs	480,407
Unique posts with URLs	1,222,137
Unique entities posting URLs	856,758
Unique posts with malicious URLs	11,217
Unique entities posting malicious URLs	7,962
Unique malicious URLs	4,622

[3] https://apps.facebook.com/mypagekeeper/

[4] http://docs.python-requests.org/en/latest/

[5] http://longurl.org/api

TABLE I. EVENT NAME, KEYWORDS USED AS SEARCH QUERIES, NUMBER OF POSTS, AND DESCRIPTION FOR THE 17 EVENTS IN OUR DATASET.

Event *(keywords)*	# Posts	Description
Missing Air Algerie Flight AH5017 *(ah5017; air algerie)*	6,767	Air Algerie flight 5017 disappeared from radar 50 minutes after take off on July 24, 2014. Found crashed near Mali; no survivors.
Boston Marathon Blasts *(prayforboston; marathon blasts; boston marathon)*	1,480,467	Two pressure cooker bombs exploded during the Boston Marathon at 2:49 pm EDT, April 15, 2013, killing 3 and injuring 264.
Cyclone Phailin *(phailin; cyclonephailin)*	60,016	Phailin was the second-strongest tropical cyclone ever to make landfall in India on October 11, 2013.
FIFA World Cup 2014 *(worldcup; fifaworldcup)*	67,406	20th edition of FIFA world cup, began on June 12, 2014. Germany beat Argentina in the final to win the tournament.
Unrest in Gaza *(gaza)*	31,302	Israel launched Operation Protective Edge in the Hamas-ruled Gaza Strip on July 8, 2014.
Heartbleed bug in OpenSSL *(heartbleed)*	8,362	Security bug in OpenSSL disclosed on April 1, 2014. About 17% of the world's web servers found to be at risk.
IPL 2013 *(ipl; ipl6; ipl2013)*	708,483	Edition 6 of IPL cricket tournament hosted in India, April-May 2013.
IPL 2014 *(ipl; ipl7)*	59,126	Edition 7 of IPL cricket tournament jointly hosted by United Arab Emirates and India, April-May 2013.
Lee Rigby's murder in Woolwich *(woolwich; londonattack)*	86,083	British soldier Lee Rigby attacked and murdered by Michael Adebolajo and Michael Adebowale in Woolwich, London on May 22, 2013.
Malaysian Airlines Flight MH17 shot down *(mh17)*	27,624	Malaysia Airlines Flight 17 crashed on 17 July 2014, presumed to have been shot down, killing all 298 on board.
Metro-North Train Derailment *(bronx derailment; metro north derailment; metronorth)*	1,165	A Metro-North Railroad Hudson Line passenger train derailed near the Spuyten Duyvil station in the New York City borough of the Bronx on December 1, 2013. Four killed, 59 injured.
Washington Navy Yard Shootings *(washington navy yard; navy yard shooting; NavyYardShooting)*	4,562	Lone gunman Aaron Alexis killed 12 and injured 3 in a mass shooting at the Naval Sea Systems Command (NAVSEA) headquarters inside the Washington Navy Yard in Washington, D.C. on Sept. 16, 2013.
Death of Nelson Mandela *(nelson; mandela; nelsonmandela; madiba)*	1,319,745	Nelson Mandela, the first elected President of South Africa, died on December 5, 2013. He was 95.
Birth of the fist Royal Baby *(RoyalBabyWatch; kate middleton; royalbaby)*	90,096	Prince George of Cambridge, first son of Prince William, and Catherine (Kate Middleton), was born on July 22, 2013.
Typhoon Haiyan *(haiyan; yolanda; typhoon philippines)*	486,325	Typhoon Haiyan (Yolanda), one of the strongest tropical cyclones ever recorded, devastated parts of Southeast Asia on Nov. 8, 2013.
T20 Cricket World Cup *(wt20; wt2014)*	25,209	Fifth ICC World Twenty20 cricket competition, hosted in Bangladesh during March-April, 2014. Sri Lanka won the tournament.
Wimbledon Tennis 2014 *(wimbledon)*	2,633	128th Wimbledon Tennis championship held between June 23, and July 6, 2014. Novak Djokovic from Serbia won the championship.

The scan results returned by the VirusTotal API contain domain information from multiple services like TrendMicro, BitDefender, WebSense ThreatSeeker, etc. for a given domain. We marked a URL as malicious if one or more of these services categorized the URL domain as *spam*, *malicious*, or *phishing*. The Web of Trust (WOT) API returns a reputation score for a given domain. Reputations are measured for domains in several *components*, for example, trustworthiness. For each {`domain, component`} pair, the system computes two values: a *reputation* estimate and the *confidence* in the reputation. Together, these indicate the amount of trust in the domain in the given component. A *reputation* estimate of below 60 indicates *unsatisfactory*. The WOT browser add-on requires a confidence value of $\geq$ 10 before it presents a warning about a website. We tested the domain of each URL in our dataset for two components, viz. *Trustworthiness* and *Child Safety*. For our experiment, a URL was marked as malicious if both the aforementioned conditions were satisfied (Algorithm 1). In addition to reputations, the WOT rating system also computes categories for websites based on votes from users and third parties. We marked a URL as malicious if it fell under the *Negative* (including malware, scams etc.) or *Questionable* (including hate, incidental nudity etc.) category group. [6] Further, a URL was marked malicious if it was marked by SURBL, Google Safebrowsing, SpamHaus or PhishTank.

Algorithm 1 Detecting malicious posts from WOT reputation scores

```
for all posts do
    for all URL domains do
        components = GetComponentsFromWOT_API
        for all components do
            if reputation < 60 and confidence ≥ 10 then
                post = malicious
            end if
        end for
    end for
end for
```

The reason for including WOT reputation scores in our labeled dataset of malicious posts was two-fold. Firstly, to study Facebook's current techniques to counter malicious content. Facebook partnered with WOT to protect its users from malicious URLs [10]. Secondly, during news-making events, malicious entities tend to engage in spreading fake, untrustworthy and adult content to degrade user experience [17]. This kind of information, despite being malicious, is not captured by blacklists like Google Safebrowsing and SURBL, since they do not fall under the more obvious kinds of threats like malware and phishing. WOT scores helped us to identify and tag such content. In all, we found 4,622 unique malicious URLs across 11,217 unique Facebook posts (Table II).

[6] Exact category labels and description corresponding to *Negative* and *Questionable* categories can be found at https://www.mywot.com/wiki/API

IV. Analysis

We now present our findings about Facebook's current techniques of malicious content detection and the differences between malicious and legitimate content on Facebook. We use these differences to identify a set of 42 features and apply machine learning techniques to automatically identify malicious content.

A. Efficiency of Facebook's current techniques

Facebook's immune system uses multiple URL blacklists to detect malicious URLs in real time and prevent them from entering the social graph [28]. Understandably, the inefficiency of blacklists to detect URLs at zero-hour limits the effectiveness of this technique [26]. We queried the Graph API in November 2014 to check if Facebook removed any of the 11,217 malicious posts identified by blacklists after being posted. We found that only 3,921 out of the 11,217 (34.95%) malicious posts had been deleted. It was surprising to note that almost two thirds of all malicious posts (65.05%) which got past Facebook's real-time detection filters remained undetected even after 4 (or more) months (July - November, 2014) from the date of post. Collectively, these posts had gathered *likes* from 52,169 unique users and *comments* from 8,784 unique users at the time we recollected the data. Using the URL endpoint of the Graph API [7], we also found that the 4,622 unique URLs present in the 11,217 malicious posts had been shared on Facebook over 37 million times. Figure 1 shows one such malicious post from our dataset which went undetected by Facebook. The short URL in the post points to a scam website which asks users to *like* posts on Facebook to earn money.

Fig. 1. One of the 7,296 malicious posts from our dataset which were not deleted by Facebook. We revisited this post after 11 months of being posted.

Above analysis suggests that a large portion of malicious content which goes undetected by Facebook's filters not only stays undetected, but thrives on users' *likes*, *comments* and *shares*. With 4.75 billion posts generated on Facebook every day [9], re-scanning all posts to check for malicious content can be computationally expensive. This demands for an alternative real-time detection technique which does not rely on blacklists to identify malicious content.

WOT warning pages: Facebook partnered with Web of Trust in 2011 to protect its users from malicious URLs [10]. According to this partnership, Facebook claims to show a warning page to the user whenever she clicks on a link which has been reported for spam, malware, phishing or any other kind of abuse on WOT (Figure 2). To verify the existence and effectiveness of this warning page, we manually visited a random sample of 1000 posts containing a URL marked as malicious by WOT, and clicked on the URL. Surprisingly, the warning page did not appear even once. We also noticed that over 88% of all malicious URLs in our dataset (4,077 out of 4,622) were marked as malicious by WOT.

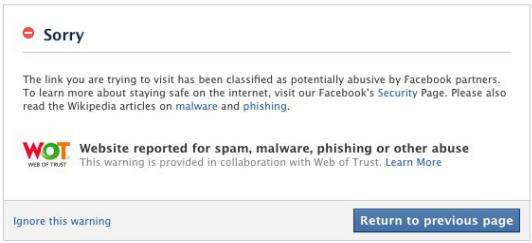


Fig. 2. Warning page shown by Facebook whenever a user clicks on a link reported as abusive on WOT.

B. Key characteristics of malicious content

We analyzed the malicious content in our dataset in three aspects – a) textual content and URLs, b) entities who post malicious content, and c) metadata associated with malicious content. We now look at all these three aspects individually.

1) Textual content and URLs: From our dataset of 11,217 unique malicious posts, we first looked at the most commonly appearing posts. Similar to past work [13], we found various *campaigns* promoting a particular entity or event. However, campaigns in our dataset were very different than those discussed in the past. Table III shows the top 10 campaigns in our dataset of malicious posts. We found that most of the campaigns in our dataset were event specific, and talked about celebrities and famous personalities who were part of the event. Although this seems fairly obvious because of our event based dataset, such campaigns reflect the attackers' preferences of using the context of an event to target OSN users. Attackers now prefer to exploit users' curiosity about news-making events in addition to hijacking trends and posting unrelated content (like promoting free iPhone, illegal drugs, cheap pills, free ringtones, etc.) using topic specific keywords to spread malicious content.

TABLE III. Top 10 most common posts in our dataset of malicious posts.

Post Summary	Count
Sexy Football Worldcup - Bodypainting	155
10 Things Nelson Mandela Said That Everyone Should Know	154
Was Bishop Desmond Tutu Frozen Out of Nelson Mandela's Funeral?	105
Nude images of Kate Middleton	73
The Gospel Profoundly Affected Nelson Mandela's Life After Prison	72
Promotion of Obamacare (Affordable Care Act) through Nelson Mandela's death	67
Radical post about Nelson Mandela	54
Was Nelson Mandela a Christian?	41
R.I.P. Nelson Mandela: How he changed the world	36
Easy free cash	29

Investigating further, we found that the most common type of malicious posts (52.0%) in our dataset were the ones with URLs pointing to adult content and incidental nudity, and marked unsafe for children by WOT. The second most common type of malicious posts comprised of negative and questionable category URLs. These categories comprised

[7] https://developers.facebook.com/docs/graph-api/reference/v2.2/url

of malware, phishing, scam, misleading claims or unethical, spam, hate, discrimination, potentially unwanted programs, etc., and accounted for 45.2% of all posts. Posts containing untrustworthy sources of information (38.22%) were the third most common type of malicious posts. Interestingly, only 325 malicious posts (2.9%) advertised a phishing URL. This is a drastic drop as compared to the observations made by Gao et al. in 2010, where authors found that over 70% of all malicious posts in their dataset advertised phishing [13]. We also found that 18.4% of the malicious posts in our dataset (2,064 posts out of 11,217) advertised one or more shortened URLs. Past literature has shown wide usage of shortened URLs to spread malicious content on microblogging platforms [6], [2]. Use of short URLs has significantly increased mostly due to restriction of message length on OSNs like Twitter. However, restriction on message length does not apply on Facebook. This implies that the primary reason behind usage of shortened URLs on Facebook is obfuscation of actual malicious URLs.

In addition to post categories, we also looked at the most common URL domains in our dataset. We observed that Facebook and YouTube constituted almost 60% of all legitimate URLs shared during the 17 events. The remaining legitimate URLs largely belonged to news websites (cnn.com, bbc.co.uk, etc.). On the contrary, malicious URLs were more evenly distributed across a mixture of news, blogs, sports, entertainment, etc. websites. Our dataset revealed that a large fraction of malicious content comprised of untrustworthy sources of information, which may have inappropriate implications in the real world, especially during events like elections, riots, etc. Most previous studies on detecting malicious content on online social networks have concentrated on identifying more obvious threats like malware and phishing [3], [15], [33]. There exists some work on studying trustworthiness of information on other social networks like Twitter [5], [16]. However, to the best of our knowledge, no past work addresses the problem of identifying untrustworthy content on Facebook.

2) Entities posting malicious content: Content on Facebook is generated by two types of entities – *users* and *pages*. Pages are public profiles specifically created for businesses, brands, celebrities, causes, and other organizations. Unlike users, pages gain "fans," people who choose to *like* a page. In our dataset, we identified pages by the presence of *category* field in the response returned by Graph API search [11] during the initial data collection process. The *category* field is specific to pages; we used this field to differentiate between pages and user profiles. We found that pages were more active in posting malicious URLs as compared to legitimate URLs. Pages were observed to constitute 21% (1,676 out of 7,962) of all malicious entities, while only 10% of all legitimate URL posting entities were pages. A similar percentage of pages (12%) was found to constitute all legitimate entities in our dataset. We also found 43 verified pages and 1 verified user among entities who posted malicious content. The most common type of verified pages were radio station pages (12), website pages (5) and public figure pages (4). Combined together, the 43 verified pages had over 71 million *likes*.

It is important to note that most of the past attempts at studying malicious content on Facebook did not capture content posted by pages, and concentrated only on users [1], [13], [29]. Malicious content originating from pages in our dataset brings out a new dimension, which hasn't been studied in the past. Facebook limits the number of *friends* a user can have, but there is no limit on the number of people who can *like* (subscribe to) a page. Content posted by a page can thus, have much larger audience than that of a user, making malicious content posted by pages potentially more widespread and dangerous than that posted by individual users. We found that in our dataset, pages posting malicious content had 123,255 *likes* on average (min. = 0, max. = 50,034,993), whereas for legitimate pages, the average number of *likes* per page was only 45,812 (min. = 0, max. = 502,938,006).

3) Metadata: There are various types of metadata associated with a post, for example, application used to post, time of post, type of post (picture / video / link), location etc. Metadata is a rich source of information that can be used to differentiate between malicious and legitimate users. Figure 3 shows the distribution of the top 25 applications used to post content in our dataset. [8] We observed that over 51% of all legitimate content was posted through mobile apps. This percentage dropped to below 15% for malicious content. Third party and custom applications (captured in "Other" in Figure 3) were used to generate 11.5% of all malicious content in our dataset as compared to only 1.4% of all legitimate content being generated by such applications. This behavior reflects that malicious entities make use of web and third party applications (possibly for automation) to spread malicious content, and can be an indicator of malicious activity. Legitimate entities, on the other hand, resort to standard mobile platforms to post.

Although Facebook has more web users than mobile users [8], our observations may be biased towards mobile users due to our event specific dataset. Past literature has shown high social network activity through mobile devices during such events [17].

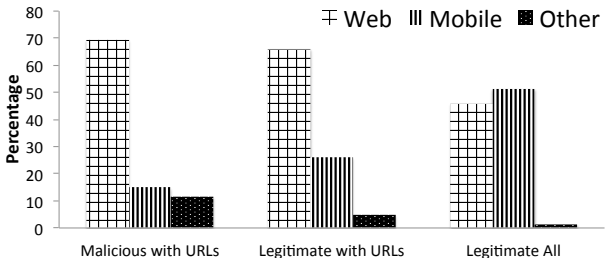


Fig. 3. Sources of malicious content, legitimate content with URLs, and all legitimate content. Mobile platforms were preferred over web for posting legitimate content.

We also observed significant difference in the content types that constituted malicious and legitimate content. Over 50% of legitimate posts containing a URL were photos or videos whereas this percentage dropped to below 6% for malicious content. A large proportion of these photos and videos were uploaded on Facebook itself. This was one of the main reasons for facebook.com being the most common legitimate domain in our dataset. We used these, and some other features to train multiple machine learning algorithms for automatic detection of malicious content. The results of our experiments are presented in the next section.

[8]The top 25 applications were used to generate over 95% of content in all three categories we analysed.

V. Detecting malicious content automatically

Past efforts for automatic detection of spam and malicious content on Facebook largely focus on detecting campaigns [12], [13], and rely heavily on *message similarity* features to detect malicious content [25]. Researchers using this approach have reported consistent accuracies of over 80% using small feature sets comprising of 6-7 features. However, this approach is ineffective for detecting newly emerged malicious content since the aforementioned models require to have seen similar spam messages in the past. To overcome this inability, we propose an extensive set of 42 features (see Table IV) to detect malicious content, excluding features like message similarity, likes, comments, shares etc., which are absent when new malicious posts surface. We group these 42 features into four categories based on the their source; Entity (E), Text content (T), Metadata (M) and Link (L).

TABLE IV. Features used for machine learning experiments. We extracted features from four sources, viz. entity, content, metadata, and link.

Source	Features
Entity (9)	is a page / user, gender, page category, has username, username length, name length, no. of words in name, locale, likes on page
Text content (18)	Presence of !, ?, !!, ??, emoticons (smile, frown), no. of words, avg. word length, no. of sentences, avg. sentence length, no. of English dictionary words, no. of hashtags, hashtags per word, no. of characters, no. of URLs, no. of URLs per word, no. of uppercase characters, no. of words / no. of unique words
Metadata (8)	App, has FB.com URL, has *message*, has *story*, has *link*, has *picture*, type, *link* length
Link (7)	has HTTP / HTTPS, hyphen count, paramters count, parameter length, no. of subdomains, path length

We trained four classifiers using 11,217 unique malicious posts as the positive class and 11,217 unique legitimate posts, randomly drawn from the 1,210,920 unique legitimate posts containing one or more URLs (see Table II) as the negative class. All experiments were performed using Weka [18]. A 10-fold cross validation on this training set yielded a maximum accuracy of 86.9% using the Random Forest classifier. Table V describes the results in detail. We also performed the classification experiments using the four category features (E, T, M, and L) separately, and observed that link (L) features performed the best, yielding an accuracy of 82.3%. A combination of all four category features, however, outperformed the individual category scores, signifying that none of the category features individually could identify malicious posts as accurately as their combination. We also calculated accuracy using 1 through all 42 features, adding features one by one in decreasing order of their information gain value, and found that the accuracy peaked at the top 10 features (Figure 4(a)). All four classifiers achieved higher accuracy when trained on the top 10 features, as compared to accuracy when trained on all 42 features (Table V). Table VI shows the information gain value and source of the top 10 features.

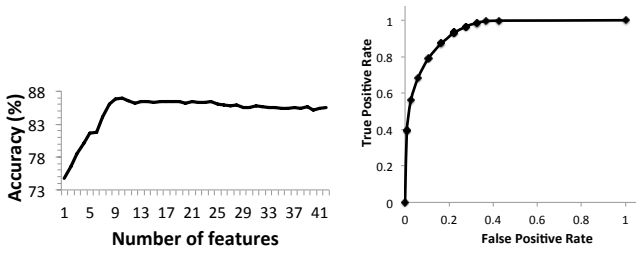
We performed further experiments to observe the change in true positive rate, false positive rate and accuracy values with change in training dataset sizes. The Random Forest classifier was used for these experiments since it gave the highest accuracy amongst the four classifiers we used in the previous experiment. We used all 42 features to train the classifier in this experiment. Keeping the size of the positive class constant (11,217 instances), the size of the training set was varied by varying the size of the negative class instances in the ratio 1:1/2, 1:1, 1:2 and 1:5. This yielded 5,609, 11,217, 22,434 and 56,085 negative class instances consecutively, randomly drawn from 1,210,920 unique legitimate posts containing one or more URLs (see Table II). We were able to achieve a maximum true positive rate of 97.7% for malicious posts using the 1:1/2 split. The false positive rate dropped to a lowest of 3.4% for 1:5 split. The 1:1 split yielded the lowest average false positive rate. Figure 4(b) shows the receiver operating characteristics (ROC) curve for the Random Forest classifier trained on a 1:1 split dataset using all 42 features, where we achieved maximum area under curve (AUC) of 0.94.

TABLE V. Ten-fold cross validation accuracies for four classifiers over six different feature sets.

Feature Set	E	T	M	L	All	Top10
Naive Bayesian	58.9	52.0	75.0	66.3	58.8	74.3
Decision Tree	**63.8**	65.4	80.8	82.0	85.0	85.8
Random Forest	63.6	**65.6**	**80.9**	**82.3**	**85.5**	**86.9**
AdaBoost	59.5	62.8	76.5	71.8	76.8	77.4

TABLE VI. Source and information gain value of the top 10 features.

Feature	Source	Info. Gain
Presence of Facebook.com URL	Metadata	0.240
Post type	Metadata	0.219
Length of parameter(s) in URL	Link	0.216
Application used to post	Metadata	0.209
Length of *link* field	Metadata	0.201
Number of parameters in URL	Link	0.178
Number of sub-domains in URL	Link	0.110
Length of URL path (after domain)	Link	0.093
Number of hyphens in URL	Link	0.084
Presence of *story* field	Metadata	0.071



(a) (b)

Fig. 4. (a) Accuracy values of the Random Forest classifier for 1 through 42 features. Accuracy peaked to 86.9% at top 10 features. (b) Receiver operating characteristic (ROC) curve for the Random Forest classifier trained on all 42 features. The Area Under Curve (AUC) was 0.94.

A. General content model versus event-specific content model

We trained a separate machine learning model using the same 42 features on a dataset of 5,446 malicious posts and 5,446 legitimate posts picked from a dataset of random, non event-specific Facebook posts containing URLs. [9] This dataset was collected by querying the Graph API using *http* as a search keyword and consisted of 267,517 posts. Creation time of posts in this dataset spanned across 2 years (April, 2013 - March,

[9] We used the same methodology to find malicious posts as we did for the 17 events in our training data.

2015). A 10-fold cross validation on this model yielded an accuracy of 91.65%. However, when this new model was tested on our event-specific dataset (11,217 malicious and 11,217 legitimate posts), the accuracy plummeted to 68.63%. This drop in performance reflects that a model trained to detect *malicious posts in general* is not capable of detecting *event-specific malicious posts* efficiently. Malicious posts generated during news-making events need to be addressed separately, which was the reason behind our focus on event-specific posts in this paper.

B. Performance over time

To check the effectiveness of our technique over time, we trained 3 models (M1, M2, and M3) using all 42 features by splitting our dataset into 3 equal-sized subsets across time (M1 trained on D1 = April - July, 2013; M2 trained on D2 = August - December, 2013; M3 trained on D3 = January - July, 2014). We also collected test data about the Ebola outbreak in Africa during August - October, 2014, consisting of 3,248 malicious and 3,248 randomly picked legitimate posts (D4). Each model was evaluated a) using 10-fold cross validation, and b) by testing on all data subsets from future time intervals. For example, M1 was tested on D2, D3, and D4; M2 was tested on D3, and D4, etc. Figure 5 represents the performance of all models over time. True positive rates obtained from 10-fold cross validation on all models were consistently high, and varied between 88.6% and 94.3%. However, we noticed a gradual overall decrease in the true positive rates of all models over time (except M1). Model M1 (trained on April - July, 2013 data) showed a considerable rise in performance when tested on D4 (Ebola dataset). Investigating this behavior lies outside the scope of this report; we intend to analyze this behavior in more detail in the future.

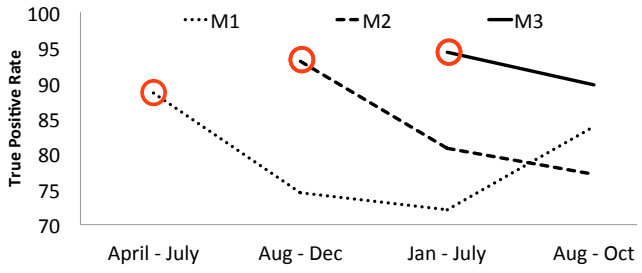


Fig. 5. True positive rates of all models over time. Encircled points indicate true positive rates on 10-fold cross validation. M1 reached the lowest true positive rate of 72.1% over Jan - July, 2014 test data.

Drop in true positive rates possibly indicate attackers' changing strategies over time. However, to maintain high true positive rates, the model can easily be retrained by collecting data using the methodology described in Section III-B.

C. Performance comparison with previous models

Gao et al. [12] used a clustering approach to identify spam campaigns and used features from these clusters to train a supervised machine learning model. In contrast, our technique treats each post separately and also takes individual malicious posts (which are not part of a campaign) into consideration. Since Gao et al.'s system was designed to detect spam messages which were part of a campaign, authors could not estimate the amount of malicious posts that this approach missed (false negatives). To this end, we applied the same clustering technique and threshold values used by Gao et al. [13] on our dataset to get an estimate of the false negatives of their approach. Since our entire dataset was already labeled (as opposed to Gao et al.'s dataset), we did not apply clustering on our entire dataset to find malicious posts. Instead, we applied clustering only on malicious posts in our dataset, and compared how many of those clusters met the *distributed* and *bursty* threshold values previously used (>5 users per cluster, and <90 minutes median time between consecutive posts respectively). Applying clustering on our 11,217 malicious posts yielded a total of 4,306 clusters. Out of these, only 183 clusters (containing 4,294 posts) met the *distributed* and *bursty* thresholds, yielding a high false negative rate of 61.7%. Our model also achieved a considerably higher true positive rate of 97.7% as compared to 80.9% achieved by Gao et al.'s system. These results indicate that existing clustering techniques are less accurate and detect less than half the amount of malicious posts as compared to our machine learning model.

Note that the dataset used by Gao et al. was collected by crawling the Facebook network (which is no longer permitted by Facebook) and captured features like users' social degree and interaction history. Due to the unavailability of Gao et al.'s dataset, and absence of social degree and interaction history features in our dataset, we were not able to do an ideal comparison of our detection techniques. Also, while Gao et al.'s system was designed to be deployed at the OSN service provider side, our approach uses completely public features and is deployed at the client side. We believe that the performance of our model will increase if it is deployed at the OSN service provider side and supplied with more user and post metadata which is not available publicly.

We were unable to compare our model accuracy results with other previous work due to two major reasons; a) absence of features like *likes, comments, message similarity* etc. at zero-hour (used by Rahman et al.), and b) public unavailability of features like number of friends, messages sent, friend choice, active friends, page likes etc. (used by Stringhini et al. [29] and Ahmed et al. [1]).

VI. REST API and Browser Plug-in

To provide a real world solution for the problem of detecting malicious content on Facebook, we built a REST API (Application Programming Interface) using the Random Forest classifier trained our labeled dataset. The API is publicly accessible and can be queried by sending a HTTP GET request containing a Facebook post ID (example: http://multiosn.iiitd.edu.in/fbapi/endpoint/?fid=10152796445701749). Due to Facebook's API limitations, our API currently works only for public posts which are accessible through Facebook's Graph API. Our API fetches the post and user / page profile information using Facebook's Graph API and generates a feature vector, which is subjected to a pre-trained classifier. The label returned by the classifier is output by the API in JSON format along with the original Facebook post ID. We also built a plug-in for Google Chrome and Firefox browsers. Once installed and enabled, this plug-in loads whenever a user opens her Facebook page, and extracts the post IDs of all public posts in the user's newsfeed. The post IDs are then sent to the REST API. If the API returns the label *malicious* for a

post, the plug-in marks the post with an "alert" symbol. We intend to make this plug-in publicly available after evaluating its efficiency and usability.

VII. Limitations

We do not claim that our dataset is representative of the entire Facebook population. Facebook does not provide any information about what fraction of public posts is returned by Graph API search. However, to the best of our knowledge, our dataset of 4.4 million public posts and 3.3 million users is the biggest dataset in literature, collected using Facebook APIs.

We understand that the WOT ratings that we used to create our labeled dataset of malicious posts are obtained through crowd sourcing, and may suffer biases. However, WOT states that in order to keep ratings more reliable, the system tracks each user's rating behavior before deciding how much it trusts the user. In addition, the meritocratic nature of WOT makes it far more difficult for spammers to abuse.

VIII. Conclusion

OSNs witness large volumes of content during real world events, providing malicious entities a lucrative environment to spread scams, and other types of malicious content. We studied content generated during 17 such events on Facebook, and found substantial presence of malicious content which evaded Facebook's existing immune system and made it to the social graph. We observed characteristic differences between malicious and legitimate posts and used them to train machine learning models for automatic detection of malicious posts. Our extensive feature set was completely derived from public information available at post creation time, and was able to detect more number of malicious posts as compared to existing clustering based spam campaign detection techniques. Finally, we deployed a real world solution in the form of a REST API and a browser plug-in to identify malicious Facebook posts in real time. In future, we would like to test (and improve) the performance and usability of our browser plug-in. We would also like to investigate Facebook pages spreading malicious content in further detail. Further, we intend to study malicious posts which do not contain URLs.

References

[1] F. Ahmed and M. Abulaish. An mcl-based approach for spam profile detection in online social networks. In *IEEE TrustCom*, pages 602–608. IEEE, 2012.

[2] D. Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. P. Markatos, and T. Karagiannis. we. b: The web of short urls. In *Proceedings of WWW*, pages 715–724. ACM, 2011.

[3] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *CEAS*, volume 6, page 12, 2010.

[4] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gonçalves. Detecting spammers and content promoters in online video social networks. In *Proceedings of ACM SIGIR*, pages 620–627. ACM, 2009.

[5] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *WWW*, pages 675–684. ACM, 2011.

[6] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru. Phi.sh/$ocial: the phishing landscape through short urls. In *CEAS*, pages 92–101. ACM, 2011.

[7] Z. Chu, I. Widjaja, and H. Wang. Detecting social spam campaigns on twitter. In *Applied Cryptography and Network Security*, pages 455–472. Springer, 2012.

[8] Facebook. http://newsroom.fb.com/company-info/. *Facebook Company Info.*, 2014.

[9] Facebook, Ericsson, and Qualcomm. A focus on efficiency. *Whitepaper, Internet.org*, 2013.

[10] Facebook Developers. Keeping you safe from scams and spam. *https://www.facebook.com/notes/facebook-security/keeping-you-safe-from-scams-and-spam/10150174826745766*, 2011.

[11] Facebook Developers. Facebook graph api search. *https://developers.facebook.com/docs/graph-api/using-graph-api/v1.0#search*, 2013.

[12] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. N. Choudhary. Towards online spam filtering in social networks. In *NDSS*, 2012.

[13] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Internet Measurement Conference*, pages 35–47. ACM, 2010.

[14] Google. Safe browsing api. *https://developers.google.com/safe-browsing/*, 2014.

[15] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @ spam: the underground on 140 characters or less. In *CCS*, pages 27–37. ACM, 2010.

[16] A. Gupta and P. Kumaraguru. Credibility ranking of tweets during high impact events. In *PSOSM*. ACM, 2012.

[17] A. Gupta, H. Lamba, and P. Kumaraguru. $1.00 per rt #bostonmarathon #prayforboston: Analyzing fake content on twitter. In *eCRS*, page 12. IEEE, 2013.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[19] Hispasec Sistemas S.L. VirusTotal Public API. *https://www.virustotal.com/en/documentation/public-api/*, 2013.

[20] J. Holcomb, J. Gottfried, and A. Mitchell. News use across social media platforms. *Technical report, Pew Research Center.*, 2013.

[21] Marca.com. Luis suarez used as bait for facebook scam. *http://www.marca.com/2014/07/18/en/football/barcelona/1405709402.html*, 2014.

[22] OpenDNS. Phishtank api. *http://www.phishtank.com/api_info.php*, 2014.

[23] E. Owens and C. Turitzin. News feed fyi: Cleaning up news feed spam. *http://newsroom.fb.com/news/2014/04/news-feed-fyi-cleaning-up-news-feed-spam/*, 2014.

[24] E. Owens and U. Weinsberg. News feed fyi: Showing fewer hoaxes. *https://newsroom.fb.com/news/2015/01/news-feed-fyi-showing-fewer-hoaxes/*, 2015.

[25] M. S. Rahman, T.-K. Huang, H. V. Madhyastha, and M. Faloutsos. Efficient and scalable socware detection in online social networks. In *USENIX Security Symposium*, pages 663–678, 2012.

[26] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang. An empirical analysis of phishing blacklists. In *Sixth Conference on Email and Anti-Spam (CEAS)*, 2009.

[27] SpamHaus. Domain block list. *http://www.spamhaus.org/dbl/*, 2014.

[28] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *Workshop on Social Network Systems*, page 8. ACM, 2011.

[29] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *ACSAC*, pages 1–9. ACM, 2010.

[30] SURBL, URI. Reputation data. *http://www.surbl.org/surbl-analysis*, 2011.

[31] M. Szell, S. Grauwin, and C. Ratti. Contraction of online response to major events. *PLoS ONE 9(2): e89052, MIT*, 2014.

[32] TheGuardian. Facebook spammers make $200m just posting links, researchers say. *http://www.theguardian.com/technology/2013/aug/28/facebook-spam-202-million-italian-research*, 2013.

[33] A. H. Wang. Don't follow me: Spam detection in twitter. In *SECRYPT*, pages 1–10. IEEE, 2010.

[34] WOT. Web of trust api. *https://www.mywot.com/en/api*, 2014.

[35] M. Zech. Flight 17 spam scams on facebook, twitter. *http://www.nltimes.nl/2014/07/22/flight-17-spam-scams-facebook-twitter/*, 2014.