$MALT^P$: Parallel Prediction of Malicious Tweets

Eric Lancaster, Tanmoy Chakraborty, and V. S. Subrahmanian

***Abstract*—It has been reported that embedded URLs and multimodal content (images, video, and sound recordings) in tweets are increasingly used to seduce users into a "wrong click," leading to malware infection. In this paper, we predict whether a tweet is malicious or not by examining five classes of features: textual content including sentiment, paths emanating from a URL mentioned in the tweet, attributes associated with URLs, and multimodal content in the tweet. A fifth class of features first constructs a novel "tweet graph" and then defines features by analyzing "metapaths" contained in the tweet graph. Next, we propose a MALicious Tweets in Parallel ($MALT^P$) collective classification algorithm that merges together tweet graphs, metapaths, and collective classification proposed previously in the literature. We conduct detailed experiments using two data sets—Warningbird (WB) and KBA. We show that our metapath-based approach outperforms past efforts at identifying malicious tweets and further show that metapath-based features in conjunction with Alexa ranks and features from KBA yield very high predictive accuracy—over 0.98 on KBA and over 0.94 on KBA, outperforming past work. More significantly, metapath features alone generate a predictive accuracy of 0.977 and 0.923, respectively, on the KBA and WB data sets, significantly outperforming the other methods in isolation. We conduct a further analysis to identify the most important features; surprisingly, our results show that the presence of multimodal content is not a major factor and that metapath-based features dominate in separating malicious from benign tweets.**

***Index Terms*—Machine learning, phishing, predictive modeling, security, social media.**

I. Introduction

TWITTER is being increasingly used as a vector for a variety of cyberattacks. As far back as 2009, the Mikeyy worm[1] tricked users into clicking on a click, and once clicked, sent the infected link to the user's followers as well. In April 2013, security intelligence reported that IBM researchers had discovered malware that used Twitter to spread malicious links.[2] At BlackHat 2016, security researchers at ZeroFox showed that click-through rates by unsuspecting Twitter users exceeded 30%—an astonishing number [1]. A May 2017 article in the New York Times described a Twitter phishing attack disguised as a vacation offer that compromised computers in the U.S. Department of Defense (DoD).[3] More recently, in January 2018, a new virus used the hashtag #FBPE to trick users into following a malicious link to discover who looked at their Twitter profile.[4] Simply put, phishing attacks using Twitter as a distribution channel are growing in number. The goal of this paper is to develop Twitter specific methods to automatically identify malicious tweets.

Though there is much past work on determining whether a link is malicious or not, there is relatively little work on whether a tweet containing a link is malicious or not. The fact that tweet content and form may be critical is apparent from the vacation offer attack that compromised DoD machines, the Mikeyy worm attack, and the #FBPE attack mentioned earlier. All of these use "lures" in the textual content of a tweet to successfully compromise a device.

There is considerable past work on identifying spam URLs. The Warningbird (WB) system [2] pioneered the use of redirection chains. Redirection chains are a sequence of links $\ell_1, \ldots, \ell_n$ in which the first link ℓ_1 is in the tweet and the ith link ℓ_i contains a hyperlink to the $(i+1)$th link. Another important effort [3] on identifying spam URLs continues with the idea of redirection chains which now constitute a graph—and they use methods to detect suspicious nodes in this graph. Neither of these studies use linguistic features nor do they use any other features that may be derivable from tweet content.

We present a more exhaustive study that uses a number of important new features. Our first innovation is that we transform a set of tweets into a tweet graph which is an Resource Description Framework (RDF) graph consisting of different types of nodes (e.g., $\mathsf{user, URL, tweet, andhashtag}$) and labeled edges. Although follower–followee graphs have long been known in Twitter, our tweet graph is fundamentally different and captures not only semantic relationships between entities contained within a tweet but also some aspects of redirection chains. Second, we use the concept of metapaths in such graphs pioneered for other reasons in [4]. Metapaths examine paths with certain sequences of edge labels and our metapath related features examine statistics associated with such paths. Third, we develop features related to the intensity of sentiment [5], [6] expressed in tweets. Fourth, we define a set of features linked to the presence or absence of multimodal content in a tweet—for instance, there are reports [7] that images, video, and audio clips have contained

Manuscript received February 12, 2018; revised July 26, 2018; accepted August 28, 2018. Date of publication September 26, 2018; date of current version December 3, 2018. This work was supported in part by ARO under Grant W911NF-13-1-0421 and Grant W911NF-15-1-0576, in part by ONR under Grant N00014-13-1-0703 and Grant N00014-16-1-2896, and in part by Maryland Procurement Office under Contract H98230-14-C-0137. The work of T. Chakraborty was supported by the Infosys center for AI, Ramanujan Faculty Fellowship, and Indo-U.K. Collaborative Project under Grant DST/INT/UKP-158/2017. *(Corresponding author: V. S. Subrahmanian.)*

E. Lancaster is with the Computer Science Department, University of Maryland, College Park, MD 20742 USA (e-mail: elancast7@gmail.com).

T. Chakraborty is with the Department of Computer Science and Engineering, Indraprastha Institute of Information Technology Delhi, New Delhi 110020, India (e-mail: tanmoy@iiitd.ac.in).

V. S. Subrahmanian is with the Computer Science Department, Dartmouth College, Hanover, NH 03755 USA (e-mail: vs@dartmouth.edu).

Digital Object Identifier 10.1109/TCSS.2018.2869171

[1] https://www.pcworld.com/article/163054/twitter_mikeyy_worm_stalkdaily.html

[2] https://securityintelligence.com/twitter-malware-spreading-more-than-just-ideas/

[3] https://www.nytimes.com/2017/05/28/technology/hackers-hide-cyberattacks-in-social-media-posts.html

[4] https://www.2-spyware.com/remove-twitter-virus.html

embedded malware. As an example, the 2015 Hammertoss malware was embedded in images in tweets.[5] Fifth, we also wondered whether the popularity of an embedded URL is related to the probability of infection in that tweet. For instance, a popular link might induce more people to click on it—we used the well-known Alexa traffic rankings (based on a mix of pageviews and number of unique visitors) as a proxy for popularity. Finally, we propose the MALicious Tweet (*MALT^P* for short) algorithm to predict whether a tweet is malicious or not. *MALT^P* can be viewed as an application of the Heterogenous Collective Classification (HCC) algorithm in [4] to the malicious tweet classification problem with some major twists. First, we define the novel notion of a tweet graph which is different from standard Twitter follower–followee graphs, hashtag/mention cooccurrence graphs, and mention graphs and use this as part of *MALT^P*. Second, *MALT^P* is a parallel version of HCC, making it more scalable. Third, *MALT^P* does not use all possible metapaths in the tweet graph, but only certain important ones, the identification of which is nontrivial.

We tested *MALT^P*'s results on two data sets—the WarningBird [2] and the KBA [3] data sets. Our experiments involved using not only a variety of classifiers but also the *MALT^P* algorithm. We additionally compare *MALT^P* with these two papers' approaches as baselines. When considered alone, *MALT^P* outperformed both these approaches on both data sets.

However, combining our feature ensemble with the features defined in KBA provided the best results, yielding area under the curves (AUCs) over 0.94 on the WB data and over 0.98 for the KBA data. More significantly, we note that *MALT^P* with metapath features alone achieves almost identical AUCs: 0.977 and 0.923 on the KBA and WB data sets, respectively, and significantly beats out existing feature sets in a head-to-head comparison.

Furthermore, when identifying the features that are most important, five of the most important features were metapath based. Moreover, sentiment scores measured by Georgia Tech's Vader system [6] and Alexa rank, used for the first time in this paper for malicious URL detection, were consistently in the top 10 most important features.

Section II provides a detailed description of the *MALT^P* architecture, defines tweet graphs, and describes the features we use in our malicious tweet prediction system, while Section III describes our *MALT^P* algorithm. Section IV describes our experimental methodology and presents both predictive accuracy results (AUCs) as well as results of runtime evaluations. Section V will describe the features most closely linked to predicting whether a tweet is malicious or not. Related work is described in Section VI—and we conclude with thoughts for the future work in Section VII.

II. *MALT^P* Architecture, Tweet Graphs, and Features

In this section, we present several important building blocks of the *MALT^P* framework. First, we present the overall *MALT^P* architecture. Next, we present the important notion of a tweet graph which differs from the usual follower–followee relationships used to represent Twitter data in graph format. Finally, we present several classes of features used in *MALT^P*. Of these, the metapath-based features are new and build on the idea of redirection graphs. In addition, we introduce features based on Alexa rank, sentiment, and multimodal tweet content that have never been used before in the context of predicting if a tweet is malicious. Due to space reasons, we do not describe the features from [3, p. 6].

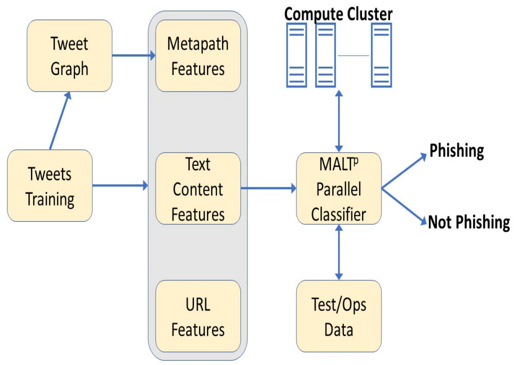

Fig. 1. *MALT^P* Architecture.

A. *MALT^P* Architecture

Fig. 1 presents the architecture of the *MALT^P* system. *MALT^P* operates via several phases. It starts by taking a training set of tweets as input, along with associated metadata about the tweet (e.g., author).

1) *Tweet Graph Construction.* It first automatically transforms the tweet data into a tweet graph which will be defined in Section II-B.
2) *Feature Extraction.* Next, *MALT^P* extracts features in two different ways. Using the tweet graph, it directly extracts a class of features called "metapath" features which will be described in Section II-C. Separately, it extracts features directly from the tweets themselves—these include sentiment related features, features related to multimodal content in the tweet, and features related to the URL or URLs referenced in the tweet.
3) *MALT^P Parallel Classifier.* Using a compute cluster consisting of K compute nodes, *MALT^P* next takes a body of test (unlabeled) tweet data (or operational data) and collectively classifies them using a base classifier. The result classifies each tweet either as a phishing tweet or a nonphishing tweet which is the final result.

B. Tweet Graphs

In this section, we introduce the important concept of a tweet graph. Suppose $\mathcal{T}$ is a set of tweets where each tweet $t \in \mathcal{T}$ has the following fields: $t.Content$ is the textual content of the tweet, $t.User$ is the author of the tweet, and $t.URL$ is the set of URLs in a tweet. The basic intuition behind a tweet graph is that we want to identify the connections between the content of tweets, the URLs that the tweets include, entry

[5] https://phoenixts.com/blog/is-malware-hiding-in-that-twitter-image/

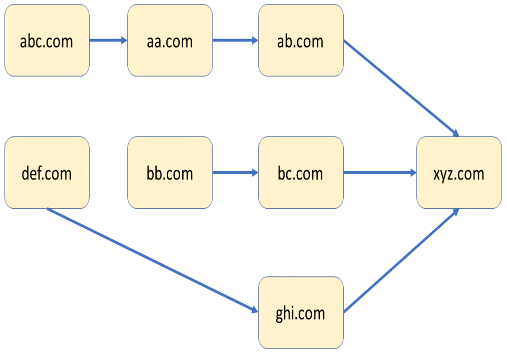

Fig. 2. Sample RCG.

URLs in the redirection graph associated with a tweet data set, and the authors of the tweets involved. All these four entities are important for the following reasons: clearly, the content of a tweet (text, multimodal) are highly relevant to "luring" unsuspecting users to either click on links or images in the tweet. Second, URLs in tweets are clearly important as well as they may lead directly or indirectly to a malicious URL. Third, entry nodes in the redirection graph are significant because they are more likely to be malicious, and finally, users must be considered—malicious users may post tweets that are malicious, and benign users may be gullible enough to retweet such content.

Example 1: Throughout this paper, we will use the following simple tweet data set as a sample set $\mathcal{T}$.

ID	Author	URL	Multimodal
tw_1	Joe	abc.com	no
tw_2	Mary	def.com	yes
tw_3	Ed	ghi.com	yes

Tweet graphs merge together aspects related to the author of a tweet, the content of the tweet, and the redirection chains (defined first in [3]) associated with the set $\mathcal{T}$ of tweets. We first define redirection chains.

A sequence $\ell_1, \ldots, \ell_n$ is a redirection chain with respect to $\mathcal{T}$ if.

1) There exists a tweet $t \in \mathcal{T}$ such that $\ell_i = t.URL$.
2) For all $1 \leq i < n$, the URL ℓ_i redirects traffic to the URL ℓ_{i+1}.

A redirection graph [3] is the graph that consists of all redirection chains associated with $\mathcal{T}$.

Fig. 2 shows an example redirection graph associated with the small tweet data set shown in example 1. The sequence abc.com, aa.com, ab.com, xyz.com is a redirection chain which says that an initial tweet (tw_1 in our example) contained the URL abc.com. However, traffic to abc.com gets redirected to aa.com which in turn redirects traffic to ab.com which in turn redirects traffic to xyz.com at which point no redirection occurs. Fig. 2 shows the three redirection chains that all end at xyz.com.

Nodes in the redirection graph associated with a given set $\mathcal{T}$ of tweets that have high in-degree are nodes that many redirection chains end in—thus, these nodes serve the ultimate goal of delivering a victim to the malicious hacker's desired landing page. The nodes in redirection chain graphs (RCGs) with the highest degrees are called entry nodes by [3]. From Fig. 2, we observe that the URLxyz.com is an entry node as it has the highest degree. If there are multiple such high-degree nodes, they are all denoted as entry nodes as they denote multiple entry points.

We are now ready to define the important concept of a tweet graph.

A tweet graph for $\mathcal{T}$ is an undirected graph represented by the triple $TH(\mathcal{T}) = (V, E, \tau)$ where V is a set of nodes consisting of all tweets, authors, URLs, and certain additional entry nodes. $E \subseteq V \times V$ is a set of edges, and $\tau : V \to \{\mathsf{user}, \mathsf{tweet}, \mathsf{url}, \mathsf{entry}\}$ assigns a type to each node. Basically, nodes in a tweet graph have a single type drawn from the set $\{\mathsf{user}, \mathsf{tweet}, \mathsf{url}, \mathsf{entry}\}$. We require all tweet graphs to satisfy the following axioms.

A1: $u \in \mathcal{T} \wedge (u, v) \in E \wedge \tau(u) = \mathsf{url} \to \tau(v) \in \{\mathsf{user}, \mathsf{tweet}, \mathsf{entry}\}$. This axiom says that url nodes cannot be connected to other url nodes. Intuitively, this axiom requires that we study the connection between urls and other types of nodes in the graph.

A2: $u \in \mathcal{T} \wedge (u, v) \in E \wedge \tau(u) = \mathsf{user} \to \tau(v) \in \{\mathsf{url}, \mathsf{tweet}\}$. This axiom says that user nodes cannot be connected to user or entry nodes. Intuitively, this axiom tries to enforce that the metapaths we study link users to either tweet content or to urls.

A3: $u \in \mathcal{T} \wedge (u, v) \in E \wedge \tau(u) = \mathsf{tweet} \to \tau(v) \in \{\mathsf{url}, \mathsf{user}\}$. This axiom says that tweet nodes cannot be connected to other tweet or entry nodes. Intuitively, this axiom says that we would like to study metapaths in which tweet nodes are linked to url nodes and user nodes.

A4: $u \in \mathcal{T} \wedge (u, v) \in E \wedge \tau(u) = \mathsf{tweet} \to \tau(v) = \mathsf{url}$. This axiom says that entry nodes can only be connected to url nodes. This is because we want to understand the relationship between url nodes explicitly mentioned in at least one tweet and the entry nodes that they are linked to by a redirection chain.

The specific tweet graph we build for $\mathcal{T}$ has as tweet nodes, the set of all tweets in $\mathcal{T}$, as user nodes, the set of all authors of tweets in $\mathcal{T}$, and as url nodes, the set of all URLs appearing in the tweet—τ is defined in an obvious way. There is an edge linking a tweet node with a url node if the tweet contains the specified URL, and likewise between a tweet node and a user node if the user wrote the tweet. The edges between user nodes and url nodes are defined similarly. entry nodes are only linked to url nodes.

Example 2: Fig. 3 shows the tweet graph associated with the sample tweet data set in example 1 and the redirection graph in Fig. 2. Note that many of the URLs that appear in the redirection graph are not included in the tweet graph—only those URLs in the redirection graph that either appear in tweets or are entry nodes appear as URL nodes in the tweet graph. Thus, tweet graphs are fundamentally different from redirection graphs.

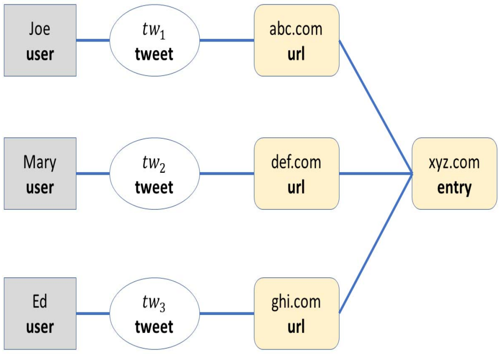

Fig. 3. Example tweet graph. Types of nodes are explicitly marked.

Example 3: Suppose we augment the situation shown in Fig. 3. Suppose an additional user, Lisa, has written a tweet tw_4 that explicitly contains both the URLs $def.com$ and $ghi.com$. In this case, the graph of Fig. 3 would be augmented with a new user node "Lisa" and a new tweet node tw_4. There would be an edge connecting the user node "Lisa" and the tweet node tw_4. In addition, the tweet node tw_4 would have an edge to both the url nodes $def.com$ and $ghi.com$.

We note that even though the Twitter data have been used to generate graphs (e.g., follower–followee graphs and mention graphs) before, tweet graphs, as defined earlier, are fundamentally different. They contain diverse types of nodes (not just people) and the edges involve input from other graphs (e.g., redirection graphs).

C. Metapaths

Metapaths were initially introduced in [4] as a way of reasoning about different types of paths in a heterogeneous information network.

A metapath of length n in the tweet graph $TG(\mathcal{T})$ is a sequence of node types $t_1 = \mathsf{url} \to t_2 \to t_n \to t_{n+1}$ where each t_i is a type. Note that the first type must always be url. For instance, url → user → url is a metapath of length 2. The reason for this is that metapaths capture the relationship between the URLs that are explicitly mentioned in a tweet, and other information related to both the tweet and the URL such as the author of the tweet, the content of the tweet, and the entry nodes that the URL is linked to. Because URLs in the tweets are the biggest focus of this paper, the notion of metapath is centered around these URLs.

Example 4: If we return to the sample tweet graph in Fig. 3, $mp_1 = \mathsf{url} \to \mathsf{entry}$ is an example metapath of length 1, while $mp_2 = \mathsf{url} \to \mathsf{entry} \to \mathsf{url} \to \mathsf{tweet}$ is a metapath of length 2. Intuitively, a metapath is a shorthand for a class of paths whose nodes have the appropriate types. We call such paths "instances" of a metapath according to the definition given in the following.

A sequence of nodes $u_1, \ldots, u_n$ in the tweet graph $TG(\mathcal{T})$ is an instance of the metapath $t_1, \ldots, t_n$ iff for all $1 \le i \le n$, $\tau(u_i) = t_i$.

Example 5: Returning to the two metapaths in example 4, we see that abc.com → xyz.com is an instance of mp_1, while abc.com → xyz.com → def.com → $\mathsf{tw_2}$ is an instance of metapath mp_2.

D. Features

We use the tweet graph and the notion of metapaths to define a set of features that (to our knowledge) are new and have not been previously used in malicious tweet prediction. In addition, we utilize features based on sentiment in the text of a tweet, Alexa rankings of traffic to websites, and past work in malicious tweet identification [2], [3]. Due to space restrictions, we focus on the metapath related features.

1) Metapath Features: Suppose $m > 0$ is an arbitrary but fixed integer denoting a bound on the length of metapaths considered. In this case, we consider the following metapath related features. Given a tweet graph TG, we use $Paths(TG, m)$ to denote the set of all metapaths of length m or less in TG. We use $TPaths(tw, TG)$ to denote the set of all paths in $Paths(TG, m)$ that contain the tweet tw.

1) Malicious: This is simply the number of URL nodes marked "malicious" in $TPaths(Tw, TG)$.
2) Benign: This is simply the number of URL nodes marked "benign" in $TPaths(Tw, TG)$.
3) url, tweet, entry, user: This is the total number of url (respectively, tweet, entryuser) nodes in $TPaths(Tw, TG)$.
4) url − user − url: This is the number of metapaths in $TPaths(Tw, TG)$ containing url − user − url as a subpath.
5) url − tweet − url, url − entry − url, url − user − tweet − url, url − tweet − user − url: These variables are similar to the one earlier.

We note that the metapath-based features have been carefully selected and do not include all possible metapaths.

For instance, metapaths are required to start at a URL node because the reputation of a URL (e.g., as measured by Alexa rankings—see in the following) may be significant in determining if a tweet referencing that URL is malicious. Thus, we do not consider metapaths starting with nodes other than url. In addition, the metapath-based paths consider in the last two feature types (bullets) mentioned earlier also end at url node. The intuition behind these metapaths is that an initial url referenced explicitly in a tweet leads to another url at the end of one of these types of paths in the tweet graph. Clearly, whether the original tweet is malicious or not depends on the url that it leads to in the tweet graph—if the latter is malicious, the tweet may be malicious as well. Moreover, whether the tweet is malicious or not may also depend upon the user involved (e.g., if he is malicious or gullible enough to tweet or retweet malicious tweets) as well as the content of the tweet. This is why the types of metapaths listed earlier are used to generate features in our framework.

In this paper, we used $m \in \{3, 4\}$. We did not want to examine $m \ge 5$ because the all-paths computation problem for $m \ge 5$ would be prohibitively expensive in graphs with millions of nodes. Likewise, $m \le 2$ was deemed too small and a quick set of experiments suggested that it would not be an avenue worth pursuing.

2) Sentiment: We evaluated the sentiment score of the text portion of each tweet using the Vader [6] sentiment analysis engine that returns a score between -1 (maximally negative) and $+1$ (maximally positive) for each tweet. Note that unlike some sentiment engines [5] that assign a score to a piece of text and a topic, VADER only assigns a score to a piece of text as a whole and does not provide topic-specific sentiment scores.

3) Multimodal: We developed binary features that checked if a given tweet contained an embedded image, video, or audio file in it.

4) Alexa Rank: Alexa[6] is a well-known commercial service that tracks information about traffic to a web site. The Alexa Rank of a tweet was defined as the worst Alexa score of any URL in the redirection chain starting at a URL originating in that tweet. Thus, the Alexa rank of the tweet tw_1 in Fig. 2 would be the minimum of the Alexa ranks of the URLs abc.com, aa.com. ab.com, xyz.com.

5) Features From Past Work: In addition to the above-mentioned features, we also leveraged features from past work [8]–[10] that were available in the WB and KBA data sets.

III. *MALT^P* Algorithm

In this section, we propose the *MALT^P* algorithm to identify malicious tweets. The algorithm is built upon the ideas of collective classification pioneered by Getoor *et al.* [11] and the idea of metapaths. *MALT^P* is a parallel version of the HCC collective classification algorithm pioneered by Yu *et al.* [4]. In addition to parallelizing HCC, we also introduce the novel concept of tweet graphs and novel kinds of metapath-based features associated with tweet graphs, as well as other feature types (e.g., sentiment, multimodal, and Alexa rank) never used before to the best of our knowledge in the context of identifying malicious tweets.

The algorithm's detailed working can be explained as follows. Given a training set Tr and a test set $Test$ of tweets, the algorithm first extracts the tweet graph $G = TW(Tr)$ associated with Tr, and extracts (from Tr) the features defined in Section II—these feature vectors are extracted in parallel in line 3 of the algorithm. After this step, each tweet tw_j can be viewed as a pair $(fv(tw_j), y_j)$ where $fv(tw_j)$ is the vector of features and y_j is its label if tw_j is in the training set Tr.

The *MALT^P* algorithm then does hyperparameter optimization on the training set for the given base classifier C, finding the best parameter settings for C—we use $C^\star$ to denote the resulting classifier with selected parameters. Note that because we do 10-fold cross validations within the training set to perform hyperparameter optimization, this step is also performed in parallel by performing the 10-folds independently on different CPUs. This concludes the training phase. Note that this phase can be executed in advance and independently of the test/operational phase as follows.

Function 1 *MALT^P*

```
Input: $Tr$-training set, $Test$-test set, $m$-max metapath
    length, $C$—classifier, $max$—int, $K$—number of processors
Output: Classification (benign/malicious) of all tweets in
    $Test$
 1: Construct tweet graph $G = TG(Tr)$;
 2: Split $Tr$ into $K$parts $Tr_1, \ldots, Tr_K$ of approximately equal
    size
 3: in parallel do: $Data(Tr_i) = \{fv(tw, y) \mid tw \in Tr_i\}$
 4: % fv(tw) is the feature vector of tweet $tw$, $y$ is $tw$'s label
 5: $Data(Tr) = \bigcup_{i=1}^{K} Data(Tr_i)$
 6: $C^\star$ = result of computing best parameter settings for
    classifier $C$ on $Data(Tr)$ using parallel 10-fold cross
    validation;
 7: % end of training phase, test phase done below
 8: Expand tweet graph: $G' = TG(Tr \cup = p Test)$
 9: Split $Test$ into $K$parts $Test_1, \ldots, Tes_K$ of approximately
    equal size
10: in parallel do: $Data(Test_i) = \{fv(tw, y) \mid tw \in Test_i\}$
11: $Data(Test) = \bigcup_{i=1}^{K} Data(Test_i)$
12: $i = 1$
13: while $i \leq max$ do
14:   for each URL node $u \in Nodes(G') - Nodes(G)$ do
15:     $u.benign = |\{v | C^\star(v) = benign \wedge u \in \mathsf{mpath}(u, m)\}|$

16:     $u.mal = |\{v \mid C^\star(v) = mal \wedge u \in \mathsf{mpath}(u, m)\}|$
17:     % number of benign/malicious nodes in metapaths of
        length $m$ or less starting at $u$
18:     $u.mal$ = number of benign nodes in metapaths of
        length $m$ or less starting at $u$
19:   end for
20:   Update feature vectors for each tweet $tw \in Tr$
21:   $C^\star$ = find best parameters for classifier $C$ in parallel
22:   update $Data(Test)$ with these estimated labels;
23:   Assign a label to each tweet node in $TG(G') - TG(G)$
      using $C^\star$;
24:   Update feature vector-ground truth pairs for each tweet
      $tw \in Test$ with the new label
25:   $i++$
26: end while
27: return  Labels for all tweets in $Test$
```

At this point, the test or operational data are considered (of course, *MALT^P* either cannot see the class labels or the class labels do not exist). The tweet graph G is now expanded to a new tweet graph G' by the addition of the tweets in the test data Test. We then extract all the features for the tweets in the test data in parallel in line 10 of the algorithm (Function 1). For any tweet $tw_h \in$ Test in the test data, we thus have the feature vector $fv(tw_h)$ but no class label, i.e., we do not know if it is malicious or not. Building on the collective classification algorithm proposed in [4], *MALT^P* then applies the classifier $C^\star$ learned previously (in Line 21) to classify each of the tweets $tw_h \in$ Test. This is an estimated class—not necessarily the final one for the test data. This causes each of the tweets in the Test set to be tentatively labeled either "benign" or "malicious." Because the tweet graph was updated from G to G', this potentially changes the values

[6] https://support.alexa.com/hc/en-us/articles/200449744-How-are-Alexa-s-traffic-rankings-determined-

TABLE I

STATISTICS ON THE TWO DATA SETS USED

	# Tweets	# Users	# URLs	Malicious ()
KBA	1.14M	373K	1.04M	186.6K
Warningbird	62.73M	373.2K	754K	186.6K

of the metapath-based features for the tweets in the original training data Tr. We, therefore, update the feature vectors of the original training data, but not the ground truth label which is assumed to be correct as it is training data.

At this stage, we have the following situation.

1) The feature vectors for the tweets in the training set have changed (potentially), but their ground truth status is what we started out with.
2) The feature vectors for the tweets in the test set have been correctly computed, but the predicted class is only an estimate.

We, therefore, learn a new set of hyperparameters for classifier C using the totality of tweets in both Tr and Test (using the labels estimated for the tweets in the Test set as a pseudoground truth). This causes the classifier C^* to be updated with a new set of hyperparameters that take the estimated classifications of the tweets in the test set into account—again using 10-fold cross validation. Note that the ground truth labels of the data in the Test set are never considered—only the estimated labels generated by the $MALT^P$ algorithm are considered.

This process is repeated iteratively till a maximum number of iterations (max) is completed. At this point, the $MALT^P$ algorithm returns the final classification generated for the tweets in the Test set.

$MALT^P$ is an adaptation of the HCC algorithm in [4] which introduces the following novel contributions. First, it uses the novel concept of a tweet graph that differs from the usual follower–followee graphs, hashtag/mention cooccurrence graphs, and mention graphs. Second, it can be viewed as adding a parallelization of the cross validation, feature extraction, and hyperparameter optimization steps. Third, it uses a carefully selected set of metapath-based features rather than blindly using all possible metapaths to generate features—which would lead to scalability problems.

IV. EXPERIMENTAL RESULTS

In this section, we report the results of a detailed set of experiments that we ran on the WB and KBA data sets. These experiments compare our methods with those in [2] and [3]

The characteristics of these data sets are given in Table I. All experiments were run on an AMD Opteron 6274 running Red Hat Enterprise Linux with 256 GB of main memory, 64 CPUs, and a 2.2 GHz. Of this computing capacity, only eight processors and 40 GB were used at most to run our code, which was entirely implemented in Python.

Following the lead of [3], for training purposes, we randomly selected a balanced subset of both data sets (as was done in [3]) of approximately the same size.

TABLE II

(a) AUCs of $MALT^P$ Using Different Classifiers and Just One Class of Features—KBA Data Set (Top) and WB Data Set (Bottom) With $m = 3$

Feature	SVM	RF	NB	SGD	KNN
Metapath	0.976	0.977	0.841	0.957	0.962
KBA	0.713	0.885	0.648	0.585	0.862
Multimodal	0.506	0.506	0.506	0.506	0.497
Alexa	0.5	0.783	0.482	0.5	0.76
Sentiment	0.531	0.62	0.540	0.541	0.573

Feature	SVM	RF	NB	SGD	KNN
Metapath	0.918	0.923	0.835	0.908	0.919
KBA	0.622	0.668	0.583	0.576	0.582
Multimodal	0.508	0.508	0.508	0.508	0.502
Alexa	0.5	0.778	0.492	0.5	0.726
Sentiment	0.495	0.539	0.497	0.495	0.514

We divided up all the features into the following classes.

1) Metapath-based features defined in this paper using metapath lengths of 3 and 4.
2) Multimodal features as defined in this paper.
3) Alexa rank features as defined in this paper.
4) **Sentiment**-based features derived using VADER [6].
5) KBA-based features described in [3].

In total, this gave us 31 combinations of classes of features (excluding the empty set of feature classes). We note that features from Warningbird [2] were also used to establish baselines on the WB data set—but because these features include IP address related features that were not available in the KBA data set, we did not use these. We ran experiments (10-fold cross validation) with all 31 combinations using an ensemble of five classifiers: support vector machines (SVMs), naive bayes (NB), random forest (RF), stochastic gradient descent (SGD), and k-nearest neighbors (KNNs). For each of the 62 cases (31 combinations × 2 metapath lengths), we measured the recall on both the benign and malicious tweets, the precision on both the benign and malicious tweets, the false positive rate, the false negative rate, the accuracy, and the area under a receiver operating characteristic curve.

A. Predictive Value of $MALT^P$ With Each Type of Feature

Our first experiment asked the question: How good are any of the five features types listed earlier in predicting malicious tweets? Table II shows the results of applying $MALT^P$ on both data sets with individual feature types—the best results are shown in bold face. The results show that metapath-based features significantly outperform all the other classes of features irrespective of which of the five classifiers is used. On the KBA data set, the algorithm provided in [3] is beaten by almost 9% points by $MALT^P$ using metapath features alone compared to the use of KBA features alone. Moreover, on the WB data set [2], the results are even more stark with Alexa features that have the second best AUC (0.778 under RF) compared to metapath which delivers 0.923 under RF as well. From these results, we can conclude that even by themselves, the metapath family of features perform very well, irrespective of which classifier is used.

TABLE III

AUCs of *MALT*^{P} Using Best Classifiers on a Single Feature Type With Metapath Lengths $m = 3$ and $m = 4$ on the WB and KBA Data Sets. Best Classifiers are Shown in Each Cell in the Table, Along With the AUC Generated. A∗: Situation Where Multiple Classifiers Generated an Identical Best Result

Feature	KBA $m = 3$	KBA $m = 4$	WB $m = 3$	WB $m = 4$
Metapath	RF,0.977	RF, 0.977	RF, 0.923	RF, 0.925
KBA	KBA, 0.885	RF, 0.885	RF, 0.775	RF, 0.668
Multimodal	*,0.506	*,0.506	* 0.508	*,0.508
Alexa	RF, 0.783	RF, 0.783	RF, 0.668	RF, 0.775
Sentiment	RF, 0.62	RF, 0.62	RF, 0.539	RF, 0.539

TABLE IV

Feature Combinations Used by *MALT*^{P} With the Five Best AUCs for the WB Data Set (Top) and the KBA Data Set (Bottom), Together With Precision and Recall for the Class of Malicious Tweets

Feature Set	AUC	Precision[m]	Recall[m]
MPath,KBA,Alexa	0.942	0.829	0.973
MPath,KBA,Alexa,MModal	0.942	0.829	0.972
MPath, Alexa	0.940	0.839	0.979
MPath, Alexa, MModal	0.940	0.839	0.978
MPath, Alexa, KBA, Sent.	0.937	0.845	0.958

Feature Set	AUC	Precision[m]	Recall[m]
MPath,KBA,MModal	0.982	0.907	0.981
MPath, KBA	0.982	0.904	0.982
MPath, KBA, Alexa	0.981	0.905	0.981
MPath, KBA, MModal, Alexa	0.981	0.905	0.981
MPath, Alexa	0.979	0.89	0.985

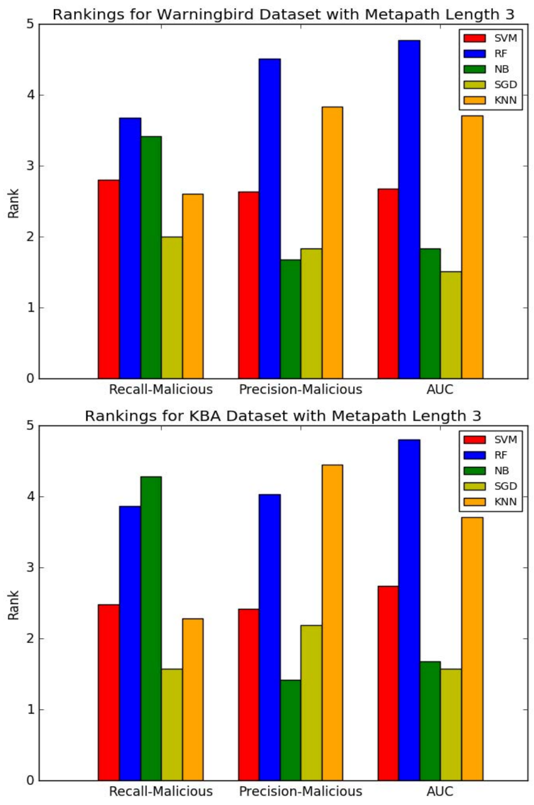

Fig. 4. Scores of *MALT*^{P} with each classifier's performance across all 62 feature combinations and m Values—five is the best possible score, one is the lowest. WB data set (top). KBA data set (bottom).

B. Comparing AUCs of MALT^P *With Best Classifier for Each Type of Feature*

Our next question was to look at both data sets (KBA and WB) and identify the family of features that generated the best result when we varied m between 3 and 4. The results are given in the following Table III. We can immediately conclude that metapath-based features provide the best predictions—moreover, there is virtually no difference between the quality of predictions between the situations when $m = 3$ and when $m = 4$. As a result, we primarily focus on the $m = 3$ case in the future because computing metapaths in Steps 15 and 16 of the *MALT*^{P} algorithm is expensive and computing shorter metapaths is, therefore, much more efficient.

C. Predictive Value of Combination of Features in MALT^P

Our next experiment compared the predictive value when all 31 combinations of feature types are considered (strictly speaking, there are 32 combinations of the 5 feature type, but one of them is the empty set which obviously has no predictive value). Because it is difficult to succinctly present a table with 310 entries (31 feature combinations × 5 feature types × 2 possible settings for m), Table IV shows the feature combinations with the five highest AUCs when $m = 3$ for each of the two data sets.

D. Classifier Which Generates the Best Performance for MALT^P

Of the five classifiers tested, we evaluated the rank of each classifier across all 31 feature combinations, for both $m = 3$ and $m = 4$ for a total of 62 combinations. Because we tested five classifiers in all, we assigned a score of five to the highest ranking classifier in each setting, a score of four to the second best performer, and so forth. For each classifier, we averaged the score across these 62 combinations, with five being the highest possible and one being the lowest possible. Fig. 4 presents the results on both the WB and the KBA data sets on three measures—AUC, precision on the malicious class, and recall on the malicious class.

In the case of the WB data set, we see that *MALT*^{P} with RF has the highest performance on all three measures and is clearly the winner. In the case of the KBA databset, however, we see that RF has the highest AUC but the second highest precision and recall on the malicious class. However, *MALT*^{P} with NB that has a better recall and *MALT*^{P} with KNN that has a better precision on the KBA data set are inferior to RF in the other four cases across the two data sets. As a consequence, we believe that *MALT*^{P} with RF yielded the best performance overall with the highest AUCs and either the highest or second highest on all other measures across both data sets.

Because RF was the best classifier in almost all respects, the rest of this analysis focuses on the results generated by RF.

E. Run Time

Our final experiment assessed the run time of the *MALT*^{P} algorithm on a cluster consisting of 1, 4, and 8 CPUs,

TABLE V

RUN TIME (IN SECONDS) OF *MALT*^{P} WITH RF (THE BEST PERFORMING CLASSIFIER) ON A PARALLEL MACHINE INCLUDING BOTH TRAINING AND TESTING TIMES

Classifier	CPUs	KBA	Warningbird
RF	1	15813	1977
RF	2	8072	532
RF	4	5460	357
RF	8	3838	28

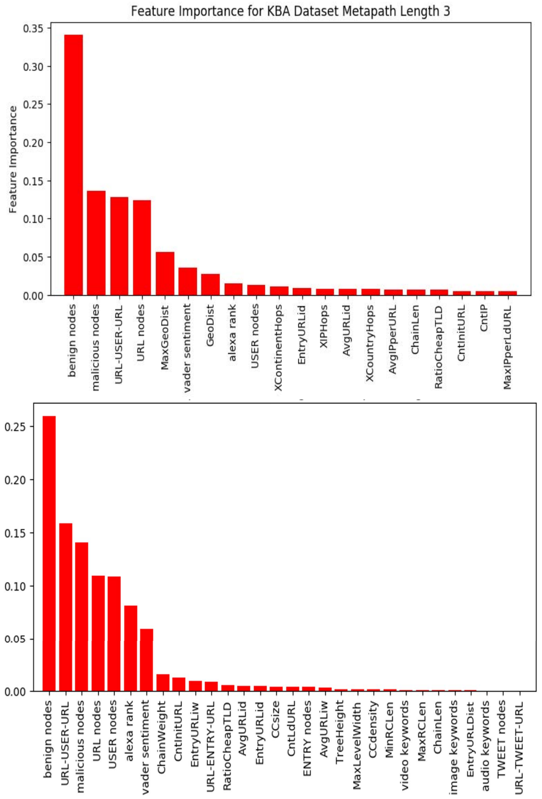

Fig. 5. Importance of features identified used by *MALT*^{P} using the best classifier (RF). Important features associated with the KBA data set (top), while ones associated with the WB data set (bottom).

respectively. Table V shows the run times for *MALT*^{P} using RF that was shown earlier to be the best classifier.

V. KEY FEATURE ANALYSIS

This section analyzes the key features that were linked to predicting whether a tweet was a phishing tweet or not.

Fig. 5 shows the most important features in both the KBA and the WB data sets identified by the *MALT*^{P} algorithm using the RF classifier with $m = 3$.

We see that 5 out of the 10 most important features on both the KBA data set and the WB data set are metapath related. Moreover, we see that seven of the most important features in each of the data sets were used for the first time in predicting malicious tweets (the two nonmetapath related ones that are new are sentiment score and Alexa rank). These results suggest that the features introduced in this paper are indeed critical for good predictions, especially as they span two very different data sets.

TABLE VI

IMPORTANCE OF THE EIGHT MOST IMPORTANT FEATURES AVERAGED ACROSS BOTH DATA SETS

Feature	Score
# Benign Nodes in Metapaths of Length 3	0.4
# Malicious Nodes in Metapaths of Length 3	0.224
# $\mathsf{url} \rightarrow \mathsf{user} \rightarrow \mathsf{url}$ Metapaths of Length 3	0.128
# url nodes in Metapaths of Length 3	0.09
# user nodes in Metapaths of Length 3	0.05
Sentiment score of tweet	0.05
Alexa Rank	0.05
RatioCheapTLD	0.01
# $\mathsf{url} - \mathsf{entry} - \mathsf{url}$ Metapaths of Length 3	0.009

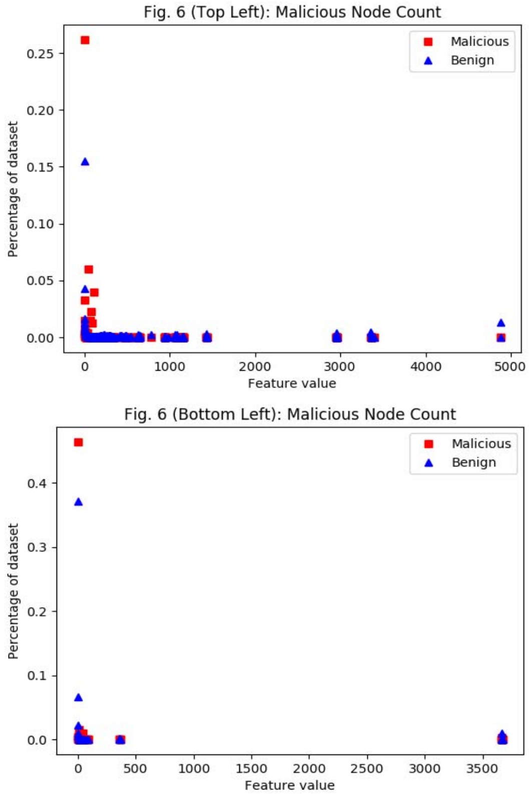

Fig. 6. Distribution of the number of malicious nodes in the KBA (top) and WB (bottom) data sets.

Table VI shows the 10 most important features in descending order based on their importance (averaged across the two data sets).

Again, we see from this table that the set of most important features is dominated by metapath related features and features used for the first time in this paper for the purpose of predicting malicious tweets.

We now examine each of these eight variables in some depth.

A. Percentage of Malicious/Benign Nodes in Metapaths of Length 3

Fig. 6 is a histogram that shows the distribution of malicious node counts in metapaths of length 3. We see that both graphs show that malicious node counts in metapaths mostly obey a

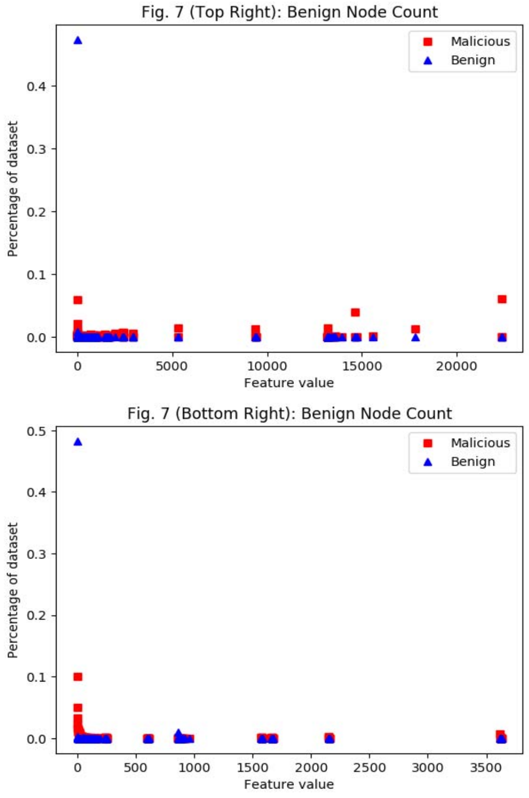

Fig. 7. Distribution of the number of benign nodes in the KBA (top) and WB (bottom) data sets.

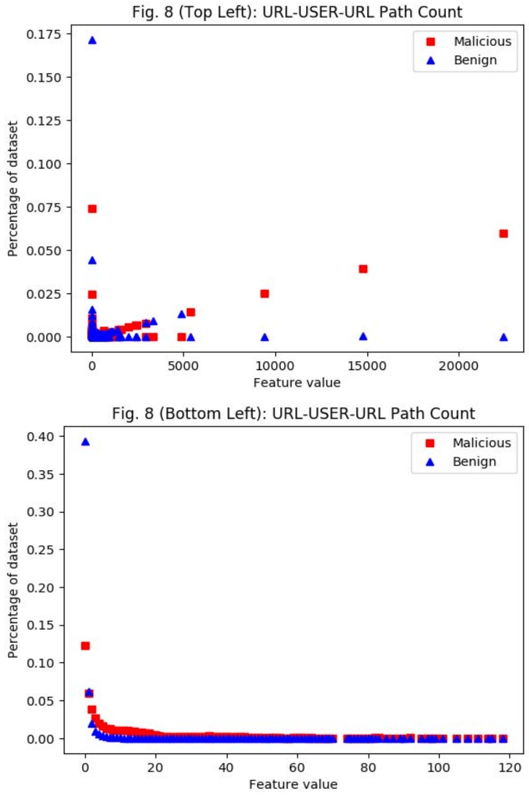

Fig. 8. Distribution of the number of metapaths in the KBA (top) and WB (bottom) data sets.

power law distribution with a long tail. As we get further along the tail, the probability of a tweet being malicious increases but is not sufficient to decisively classify a tweet as malicious.

B. Percentage of Benign Nodes in Metapaths of Length 3

Fig. 7 is a similar histogram. As the count increases, the probability of the associated tweets being malicious increases as well.

C. Number of url → user → url *Metapaths*

Fig. 8 shows the histogram as the number of such metapaths increases. In the case of both data sets, we see that when the number of such paths associated with a tweet exceeds around 40, the probability of the tweet being malicious is high, though we emphasize that this alone cannot be used as a predictor.

D. Number of url *Nodes in Metapaths of Length* 3

Fig. 9 shows the histogram of the number of URLs in metapaths of length 3 associated with a tweet. Again, this is a long-tailed distribution and we note that when the URL node count is high (over 40 for KBA and about 17 for WB), the probability that the associated tweet is malicious is high.

E. Number of user *Nodes in Metapaths of Length* 3

Fig. 10 shows the histogram of the number of user nodes in metapaths of length 3 associated with a tweet. Note that because of the axioms governing metapaths, we can only have 0 or 1 user node within a single metapath (such a metapath must have the form url → user → tweet → url). The figure shows that user node counts in such metapaths are more likely to be 1 for malicious tweets, but this cannot be used alone for prediction as the false positive rate would be too high.

F. Sentiment Score of a Tweet

Fig. 11 shows the histogram of the number of user nodes in metapaths of length 3 associated with a tweet. Interestingly, we see that tweets in data sets mostly have neutral sentiment and have similar distributions across both the benign and malicious classes of tweets.

G. Normalized Alexa Rank of URLs in a Tweet

Because Alexa Rank can vary dramatically in magnitude, we define a normalized Alexa rank $\text{NAR}(u) = (\text{AR}(u) - \text{MIN}/\text{MAX-MIN})$ where MIN, MAX are the lowest and highest Alexa ranks in our data. This nomalization places all Alexa ranks in the $[0, 1]$ interval. Fig. 12 shows the resulting distribution. Alexa ranks by themselves do not distinguish well between malicious and benign classes as they have similar long-tailed distributions.

VI. Related Work

Although the detection of malicious URLs has been studied for a long time, detection of malicious tweets has been

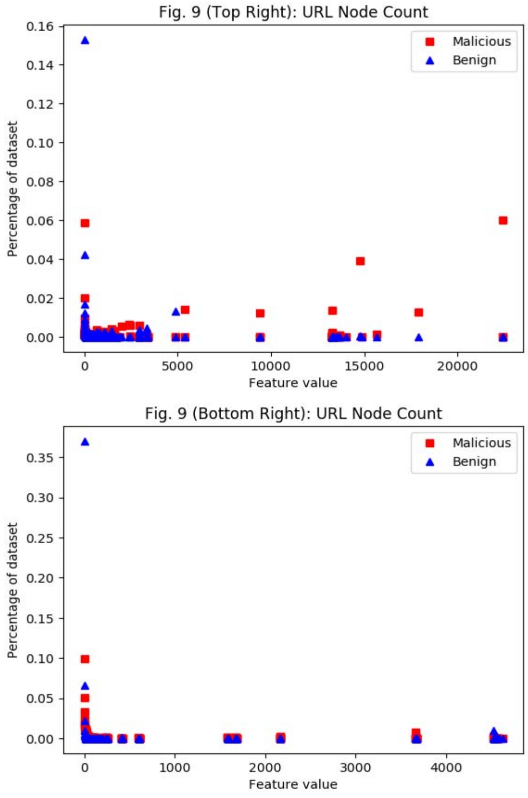

Fig. 9. Distribution of the number of url nodes in metapaths of Length 3 in the KBA (top) and WB (bottom) data sets.

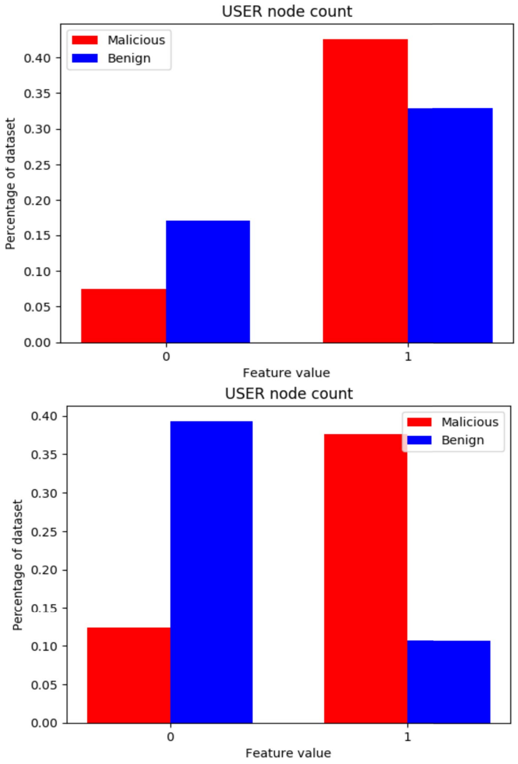

Fig. 10. Distribution of the number of user nodes in metapaths of length 3 in the KBA (top) and WB (bottom) data sets.

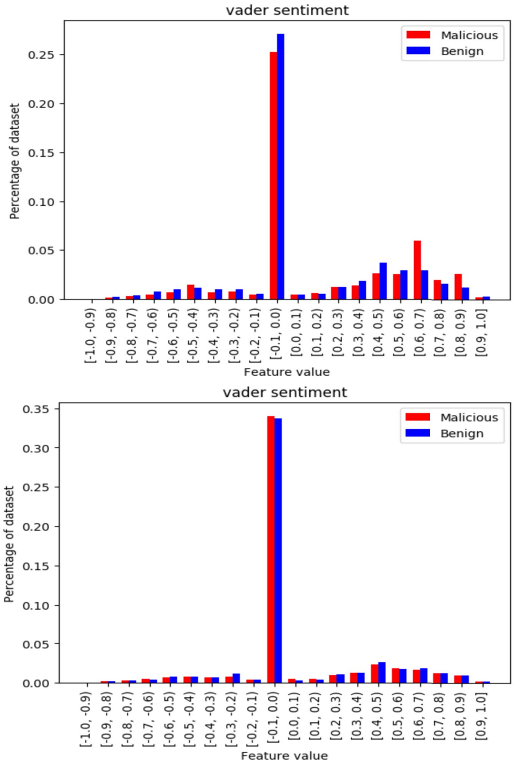

Fig. 11. Sentiment score for tweet content in the KBA (top) and WB (bottom) data sets.

studied less. The Monarch project at Berkeley [12] used logistic regression to identify malicious URLs in tweets. They crawled URLs which were submitted to web services and performed various operations to determine if those URLs redirect to malicious websites. tweetattacks.com [13] conducted a Twitter spam program which provides a web interface to delude recipients into believing that spam tweets are not spam. From a given set of tweets, Castillo *et al.* [14] evaluated the level of social media credibility of interesting topics and classified these topics into two different classes—credible or noncredible. Antoniades *et al.* [15] crawled Twitter and presented a characterization of short URLs from URL shortening services such as Bitly and TinyURL. They showed that e-mails and online social media are the biggest consumers of URL shortening services. Their analysis also reveals that most short URLs—1) last for a short duration and 2) follow a lognormal curve in their click distribution. Grier *et al.* [16] characterized spam tweets and reported that Twitter spam dominates e-mail spam in terms of short spreading time and amount of damage. They further reported that blacklist services are perform poorly in detecting new threats and, therefore, their use in detecting spam is not ideal. Beck [17] identified destructive messages and showed that special keywords are used to create spam and phishing messages in order to attract targeted users. Gao *et al.* [18] used message-based features for malicious message detection by considering the syntactic similarity of messages. Gupta and Kumaraguru [19] took 14 high impact news events of 2011 and analyzed their corresponding tweets to measure the credibility of the information. Chen *et al.* [20] analyzed over 600 million tweets with embedded URLs and found that around 1% of URLs are malicious.

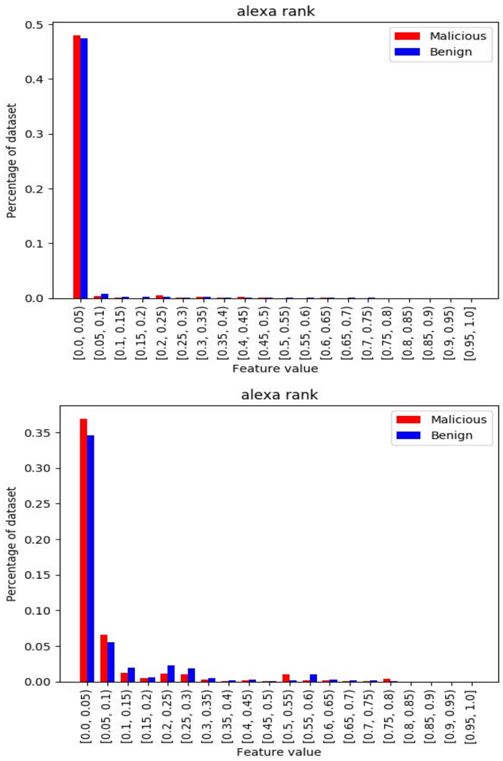

Fig. 12. Distribution of normalized Alexa rank for URLs in the KBA (top) and WB (bottom) data sets.

Nilizadeh *et al.* [21] proposed POISED, a system that leverages the differences in propagation between benign and malicious messages on social networks to identify malicious and other unwanted content. Yardi *et al.* [22] introduced a new hashtag #robotpickupline for malicious tweet detection based on three heuristics: suspicious URL search, pattern matching of usernames, and keyword detection. Kwak *et al.* [23] studied the quantitative analysis of the information diffusion in the entire Twittersphere. They concluded that if a tweet contains more than three hashtags, then they are likely to be malicious. Martinez-Romo and Araujo [24] used language modeling to measure topic divergence between tweets (using Kullback–Leibler-divergence) related to trending topics and the tweet under inspection. Their malicious tweet detection system also involved various URL blacklists. Wang *et al.* [25] studied the click traffic pattern of those URLs which are shortened by Bitly. They classified short URLs into malicious and nonmalicious URLs using click traffic analysis. Gharge and Chavan [26] collected tweets from many trending topics and classified the tweets as malicious versus nonmalicious based on content. Nepali and Wang [27] proposed a supervised model to detect malicious short URLs from tweet content. Soman and Murugappan [28] used Fuzzy K-means clustering to detect malicious tweets in trending topics. Furthermore, they used an extreme learning machine (ELM) classifier to classify the tweets. Santos *et al.* [29] proposed filtering spam tweets based on e-mail content analysis. Dongari and Reddy [30] used the correlation of URL redirection to detect suspicious URLs on a Twitter data set. Sedhai and Sun [31] developed S^{3}D, a semisupervised spam detection framework for Twitter spam detection consisting of four types of detectors: blacklist domain detector, a near-duplicate detector to label tweets of confidently prelabeled tweets, reliable ham detector to label tweets that are posted by trusted users, and multiclassifier-based detector. Aggarwal *et al.* [32] analyzed features related to phishing tweets to identify malicious links. Wu *et al.* [33] considered various microblogging sites and proposed a unified approach which captures various social relationship among users and similarity among messages. Lee and Kim [2] proposed WB, a (near) real-time suspicious URL detection system in the Twitter stream by investigating URL redirect chains extracted from tweets. Kwon *et al.* [3] proposed a spam URL detection approach by leveraging the properties of URL redirections widely deployed by spammers. Burnap *et al.* [34] developed a real time supervised classifier which takes into account machine activity generated while clicking a certain URL present in a tweet. Zhang *et al.* [35] and Gupta *et al.* [36] proposed very recent collective classification models for identifying malicious spam campaigns and campaigners in Twitter. Dutta *et al.* [37] showed how malicious tweets are promoted by blackmarket services.

VII. Conclusion

There is increasing evidence that tweets are being used by malicious hackers for a variety of purposes ranging from spreading spam to spreading malware. Examples include the #FBPE attack on U.S. DoD computers and the Hammertoss malware, among many others.

In this paper, we study the problem of identifying malicious tweets using three novel concepts. First, we use the novel concept of a tweet graph. Although there is a natural follower–followee graph associated with tweets, our tweet graph is an RDF graph that captures semantic properties of nodes. Second, we define a novel set of metapath-related features for the first time in the context of malicious tweet prediction. Third, we develop a parallel collective classification algorithm called $MALT^P$ to predict which tweets are malicious by building on a sequential algorithm in [4]. $MALT^P$ is an adaptation of the HCC algorithm in [4] which introduces the following novel contributions. First, it uses the novel concept of a tweet graph that differs from the usual follower–followee graphs, hashtag/mention cooccurrence graphs, and mention graphs. Second, it can be viewed as adding a parallelization of the cross validation, feature extraction, and hyperparameter optimization steps. Third, it uses a carefully selected set of metapath-based features rather than blindly using all possible metapaths to generate features—which would lead to scalability problems.

The results show that using metapath-based features alone, we are able to get very high AUCs, as well as high precision, and recall for the class of malicious tweets. On the KBA and WB data sets, respectively, the AUCs obtained by $MALT^P$ using metapath-based features alone are 0.977 and 0.923. When we combine metapath-based features with other features, we use for the first time on this problem (sentiment and Alexa ranks) and past work in KBA [3], we generate slightly higher AUCs (0.98 and 0.94, respectively). Moreover, *MALT^P* beats two recent efforts in this area: [2], [3] in head to head tests. In addition, we identify the key features that are important in our predictive model—5 of the top 10 are metapath-based and 7 of the top 10 were used for the first time in *MALT^P*.

Acknowledgment

The authors would like to thank the referees for many excellent comments.

References

[1] J. Seymour and P. Tully, "Weaponizing data science for social engineering: Automated E2E spear phishing on Twitter," in *Proc. Black Hat USA*, 2016, p. 37.

[2] S. Lee and J. Kim, "WarningBird: A near real-time detection system for suspicious URLs in Twitter stream," *IEEE Trans. Depend. Sec. Comput.*, vol. 10, no. 3, pp. 183–195, May 2013.

[3] H. Kwon, M. B. Baig, and L. Akoglu, "A domain-agnostic approach to spam-URL detection via redirects," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Australia: Springer, 2017, pp. 220–232.

[4] X. Kong, P. S. Yu, Y. Ding, and D. J. Wild, "Meta path-based collective classification in heterogeneous information networks," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1567–1571.

[5] V. S. Subrahmanian and D. Reforgiato, "AVA: Adjective-verb-adverb combinations for sentiment analysis," *IEEE Intell. Syst.*, vol. 23, no. 4, pp. 43–50, Jul. 2008.

[6] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. 8th Int. Conf. Weblogs Social Media (ICWSM)*, 2014. Accessed: Apr. 20, 2016. [Online]. Available: http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

[7] H. V. Nath and B. M. Mehtre, "Analysis of a multistage attack embedded in a video file," *Inf. Syst. Frontiers*, vol. 17, no. 5, pp. 1029–1037, 2015.

[8] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 649–656.

[9] G. L'Huillier, R. Weber, and N. Figueroa, "Online phishing classification using adversarial data mining and signaling games," in *Proc. ACM SIGKDD Workshop CyberSecur. Intell. Inform.*, 2009, pp. 33–42.

[10] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in *Proc. NDSS*, 2010, pp. 1–14.

[11] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, p. 93, 2008.

[12] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2011, pp. 447–462.

[13] *Tweet Attacks Pro*. Accessed: Oct. 2017. [Online]. Available: http://www.tweetattackspro.com/

[14] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 675–684.

[15] D. Antoniades *et al.*, "We.b: The Web of short URLs," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 715–724.

[16] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: The underground on 140 characters or less," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 27–37.

[17] K. Beck, "Analyzing tweets to identify malicious messages," in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT)*, May 2011, pp. 1–5.

[18] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. N. Choudhary, "Towards online spam filtering in social networks," in *Proc. NDSS*, 2012, pp. 1–16.

[19] A. Gupta and P. Kumaraguru, "Credibility ranking of tweets during high impact events," in *Proc. 1st Workshop Privacy Secur. Online Social Media*, 2012, Art. no. 2.

[20] C. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou, "6 million spam tweets: A large ground truth for timely Twitter spam detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 7065–7070.

[21] S. Nilizadeh *et al.*, "POISED: Spotting Twitter spam off the beaten paths," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1159–1174.

[22] S. Yardi, D. Romero, and G. Schoenebeck, "Detecting spam in a Twitter network," *First Monday*, vol. 15, no. 1, 2009. [Online]. Available: http://journals.uic.edu/ojs/index.php/fm/article/view/2793/2431_

[23] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 591–600.

[24] J. Martinez-Romo and L. Araujo, "Detecting malicious tweets in trending topics using a statistical analysis of language," *Expert Syst. Appl.*, vol. 40, no. 8, pp. 2992–3000, 2013.

[25] D. Wang, S. B. Navathe, L. Liu, D. Irani, A. Tamersoy, and C. Pu, "Click traffic analysis of short URL spam on Twitter," in *Proc. 9th Int. Conf. Collaborative Comput., Netw., Appl. Worksharing (Collaboratecom)*, Oct. 2013, pp. 250–259.

[26] S. Gharge and M. Chavan, "An integrated approach for malicious tweets detection using NLP," in *Proc. Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Oct. 2017, pp. 435–438.

[27] R. K. Nepali and Y. Wang, "You look suspicious!!: Leveraging visible attributes to classify malicious short URLs on Twitter," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2016, pp. 2648–2655.

[28] S. J. Soman and S. Murugappan, "Detecting malicious tweets in trending topics using clustering and classification," in *Proc. Int. Conf. Recent Trends Inf. Technol. (ICRTIT)*, Apr. 2014, pp. 1–6.

[29] I. Santos, I. Miñambres-Marcos, C. Laorden, P. Galán-García, A. Santamaría-Ibirika, and P. G. Bringas, "Twitter content-based spam filtering," in *Proc. Int. Joint Conf. SOCO-CISIS-ICEUTE*. San Sebastián, Spain: Springer, 2014, pp. 449–458.

[30] A. Dongari and M. S. Reddy, "Suspicious URL detection system using SGD algorithm for Twitter stream," *Int. J. Comput. Sci. Inf. Eng., Technol.*, vol. 2, no. 4, pp. 1–6.

[31] S. Sedhai and A. Sun, "Effect of spam on hashtag recommendation for tweets," in *Proc. 25th Int. Conf. Companion World Wide Web*, 2016, pp. 97–98.

[32] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, "PhishAri: Automatic realtime phishing detection on Twitter," in *Proc. eCrime Researchers Summit (eCrime)*, Oct. 2012, pp. 1–12.

[33] F. Wu, J. Shu, Y. Huang, and Z. Yuan, "Co-detecting social spammers and spam messages in microblogging via exploiting social contexts," *Neurocomputing*, vol. 201, pp. 51–65, Aug. 2016.

[34] P. Burnap, A. Javed, O. F. Rana, and M. S. Awan, "Real-time classification of malicious URLs on Twitter using machine activity data," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2015, pp. 970–977.

[35] X. Zhang, S. Zhu, and W. Liang, "Detecting spam and promoting campaigns in the Twitter social network," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 1194–1199.

[36] S. Gupta, A. Khattar, A. Gogia, P. Kumaraguru, and T. Chakraborty, "Collective classification of spam campaigners on Twitter: A hierarchical meta-path based approach," in *Proc. Web Conf. (WWW)*, 2018, pp. 529–538.

[37] H. S. Dutta, A. Chetan, B. Joshi, and T. Chakraborty. (2018). "Retweet us, we will retweet you: Spotting collusive retweeters involved in blackmarket services." [Online]. Available: https://arxiv.org/abs/1806.08979

Eric Lancaster is a Senior in computer science and mathematics from the University of Maryland, College Park, MD, USA.

His current research interests include machine learning and artificial intelligence research.

Tanmoy Chakraborty received the Ph.D. degree from IIT Kharagpur, Kharagpur, India, in 2015, as a Google India Ph.D. Fellow. His Ph.D thesis was recognized as best thesis by the IBM Research India, Xerox Research India, and Indian National Academy of Engineering.

He was a Post-Doctoral Researcher with the University of Maryland, College Park, MD, USA. He is currently an Assistant Professor and a Ramanujan Fellow with the Department of Computer Science and Engineering, Indraprastha Institute of Information Technology Delhi, New Delhi, India. His current research interests include data mining, social network analysis, and data-driven cybersecurity.

Dr. Chakraborty was a recipient of the DAAD Faculty Fellowship and the Early Career Research Award.

V. S. Subrahmanian served as the Director for the University of Maryland's Institute for Advanced Computer Studies. From 1989 to 2017, he was a Professor of computer science with the University of Maryland, College Park, MD, USA, and the Head of the Center for Digital International Government, College Park. He is currently the Dartmouth College Distinguished Professor in cybersecurity, technology, and society and the Director of the Institute for Security, Technology, and Society, Hanover, NH, USA. He is an Expert on big data analytics including methods to analyze text/geospatial/relational/social network data, learn behavioral models from the data, forecast actions, and influence behaviors. His models have been used to forecast terror attacks and terror network evolution, to reduce poaching, to identify bad actors on social media, to forecast systemic banking crises, to maximize airline profits, to address cyberattacks, and more. He was named to ISIHighlyCited.com which lists the top-most cited computer scientists of all time. His work has been featured in numerous outlets such as *the Baltimore Sun*, *the Economist*, *Science*, *Nature*, *the Washington Post*, and *American Public Media*. He has authored or co-authored 5 books, edited 6, and authored or co-authored over 300 articles.

Dr. Subrahmanian is a Fellow of the American Association for the Advancement of Science and the Association for the Advancement of Artificial Intelligence. He previously served on DARPA's Executive Advisory Council on Advanced Logistics and an Ad Hoc Member of the U.S. Air Force Science Advisory Board in 2001. He has received numerous awards. He serves on the editorial boards of numerous journals including *Science*, the Board of Directors of the Development Gateway Foundation (set up by the World Bank), Washington, DC, USA, SentiMetrix, Inc., Bethesda, MD, USA, and the Research Advisory Board of Tata Consultancy Services, Mumbai, India.