

Content-Based Approach for Identifying Textual Ads-Portal Domains

MISHARI ALMISHARI

University of California, Irvine

and

XIN LIU and XIAOWEI YANG

Duke University

A textual ads-portal domain refers to a web domain that shows advertisement in the form of ads listing and no real content. The ads content in an ads-portal domain is served by a third-party advertisement syndication service. We develop a machine-learning-based classifier to identify ads-portal domains. The resulting classifier has 97% accuracy in identifying ads-portal domains. We use this classifier to measure the prevalence of ads-portal domains in the Internet and to find the ratio of ads-domains that are typo-squatting. We find that 30.5/26.6% of **.com/*.net* domains are ads-portal domains and 30.16/22.3% of **.com/*.net* ads-portal domains are typos. In addition, we use the classifier along with the DNS trace files to estimate how often Internet users visit ads-portal domains and typo ads-portal domains. It turns out that 4.5/5.4% of the **.com/*.net* domains found in the trace files are ads-portal domains and 24.7/22.4% of **.com/*.net* ads-portal domains in the traces are typos. These numbers show that ads-portal domains and typo-squatting domains are prevalent in the Internet and successful in attracting many users to them. Our identification methodology represents a step towards better mining and categorizing the web domains. It also can be helpful to search engines ranking algorithms, helpful in identifying web spams that redirects to ads-portal domains, and used to discourage access to typo-squatting domains.

1. INTRODUCTION

A textual ads-portal domain refers to a web domain that only shows advertisements in the form of ads listing and does not have real content. Generally, third-party syndication servers provide the ads-content to ads-portal domains. Recently, the existence of such domains in the Internet is more apparent. It is not uncommon for Internet users to come across such domains either accidentally or on purpose. Figure 1 shows some examples of ads-portal domains.

Ads-portal domains are useful in showing related(to their domain names) ads content to users performing what so called Direct Search. Direct Search or Type-in Traffic refers to the practice of searching for a specific topic in the Internet by bypassing regular search engines and directly typing a topic-related domain name in the address bar hoping it would resolve to an ads-portal page related to the sought topic [Wikipedia 2008]. For example, a user interested in nail-related topic may bypass the search engines and directly type *www.nail.com* in the browser address bar, and then the user will be landed on the page

Authors' addresses: Mishari Almishari(contact author), UCI Computer Science Department, Irvine, CA 92697; email: malmisha@uci.edu; Xin Liu and Xiaowei Yang, Duke University Computer Science Department, Durham, NC 27708; email:{xinl,xwy}@cs.duke.com.

shown in Figure 1(a). This domain is an ads-portal domain showing ads links that are nail-related. In this context, ads-portal domains could be helpful to users as they save them the hassle of search engines and immediately show them what they really need.

Despite their usefulness, those ads-portal domains are missused in at least two ways. The first way of missusing ads-portal domains is typo-squatting. Typo-squatting refers to the practice of registering domain names that are typo-graphical errors of other well-known domains [Wang et al. 2006]. The ease of setting up ads-portal domains encourages many Internet users to register typo domains and set these typo domains to resolve to ads-portal pages. Typo-squatters abuse the ads syndication business to monetize the incoming traffic to their domains, which is meant to be to other target domains. The other way of missusing ads-portal domains is in the web redirection spam [Wang et al. 2007]. Web spam pages are web pages that mislead search engines, using questionable search engine optimization (SEO) techniques, to promote their URLs in the lists of search results. Web redirection spam is one type of web spam where the web spam page redirects the browser to another spammer-controlled page, which is mostly an ads-portal page. Wang et al. [2007] show that many web-spam URLs redirect traffic to ads-portal pages controlled by web-spammers. This way, the web spammer monetize the incoming traffic to his/her spam URL.

The first goal of our research is to develop a methodology that accurately identifies ads-portal domains from other web domains. Identifying ads-portal domain is a challenging problem as many ads-portal domains adopt different patterns in showing their ads. What makes it even more difficult is that some non-ads portal domains, such as web directories and web guides, have their look-and-feel similar to ads-portal domains. Figures 2 shows snapshots of such confusing non-ads-portal domains. To address this challenge, we first explore a number of content-based properties of ads-portal domains. Then, we verify the effectiveness of these properties, in terms of distinguishing ads-portal domains, by analyzing their distributions. After that, we employ machine learning techniques and our effective properties to produce a binary classifier that has 91% accuracy in identifying ads-portal domains. Finally, we enhance the performance of the classifier by adding other keyword-based features to the feature vector. The accuracy of the resulting classifier increases to 97%.

It is known that ads-portal domains exist in the Internet and that typo-squatters are using them to monetize the traffic to their typo-domains. How prevalent ads-portal domains are in the Internet? Do they represent a trivial or a major ratio of Internet web domains? What percentage of those domains are typo domains? How often ads-portal domains are accessed by Internet users? Do Internet users access ads-portal domains because of the typos they make? The second goal of our research project is to perform measurements that are intended to answer these questions. In our measurements, we use our identification methodology to identify ads-portal domains and we use third-party services to identify typos. Specifically, we use the well-known typo correction services that are provided by Google [Google 2006] and Yahoo [Yahoo 2008]. If either Google's corrector or Yahoo's corrector corrects a given domain, we consider the domain as a typo.

To measure the prevalence of (typo) ads-portal domains in the Internet, we collected *.com and *.net TLD zone files from *www.verisign.com*, sampled these files, downloaded the sampled domains, and then we fed these domains to our classifier. Surprisingly, we found that 30.5/26.6% of *.com/*.net were ads-portal domains. Also, we found that 30.16/22.3% of *.com/*.net were typos. To measure how often users access these (typo) ads-portal do-

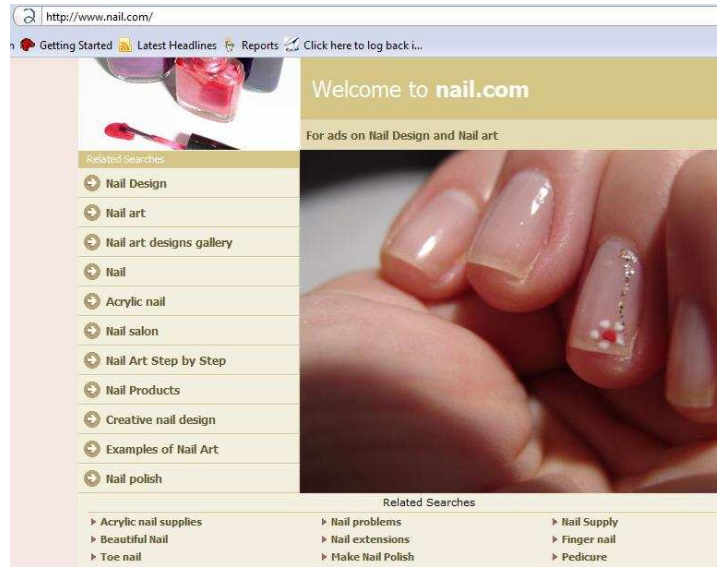
mains, we collected two-month DNS trace files collected at UCI name resolvers. Then, we random sampled *.com and *.net domains, downloaded them, and fed them to our classifier. We found that 4.5/5.4% of the accessed *.com/*.net were ads-portal domains and 24.7/22.4% of the accessed *.com/*.net ads-portal domains were typos. These measurements show the considerable prevalence of ads-portal domains in the Internet and the potential of attracting a large portion of Internet traffic. Also, The measurements show that typo-squatting domains represent a huge ratio of ads-portal domains and many ads-portal domains are accessed because they are typos. Note that these measurement results are completely different from the results in [Wang et al. 2006; McAfee 2007; Banerjee et al. 2008] which mainly show that a large number of typo-squatting domains exist, but do not show how many of the ads-portal domains are typo-squatting.

The importance of this work is multifold. This accurate identification methodology represents a step towards better mining and categorizing the web domains and documents as 30.5/26.6% of the *.com/*.net web domains can be accurately identified by our classifier. Also, our classifier can be helpful to search engines ranking algorithms because knowing a domain is ads-portal may help the search engines to lower-rank this domain in the search result results or at least tag it, as many users using the search engines may not be interested in seeing ads-portal domains. In addition, our classifier can also be helpful to search engines in avoiding indexing web spam pages that redirect to ads-portal domains. Moreover, our methodology can be used, along with some typo function, by Internet browsers to avoid access to typo-squatting domains and making such a practice less profitable. Note that relying on a typo-function only to identify typo-squatting domains may lead to huge ratio of false positives. In fact, we found that 68% of the domains in our DNS traces that were one-error-away from one of the top 100 US domains, taken from *alexa.com*, were legitimate non-ads-portal domains. That means that using our identification methodology is helpful in better identifying typo-squatting domains and not confusing them with legitimate domains that happen to be typos to some well-known domains.

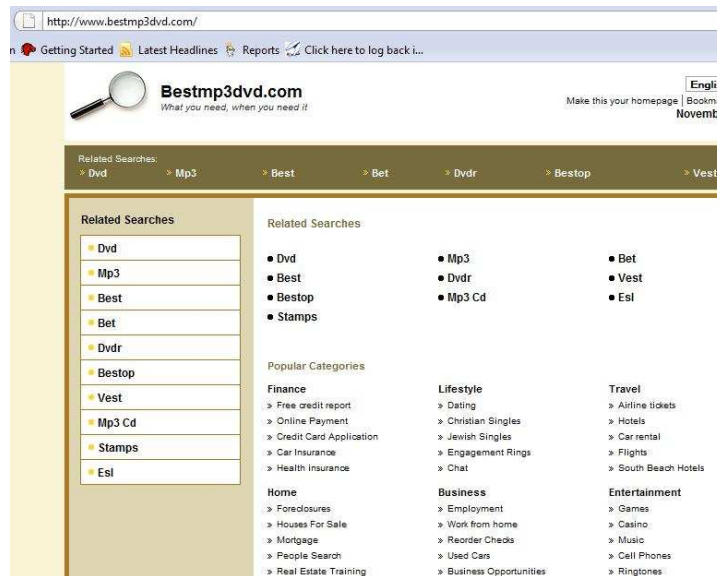
The rest of this paper is organized as follows: In Section 2, we provide some background information about parking services. In Section 3, we describe the data sets we use for training the classifier and performing the measurements. In Section 4, we describe a set of content-based properties of ads-portal domains. In Section 5, we show how we employ machine learning techniques to identify ads-portal domains. In Section 6, we present several experiments that show the prevalence of ads-portal domains in the Internet and how often they are accessed by Internet users. In Section 7, we discuss our future directions. In Section 8, we discuss the related work, and finally, in Section 9, we conclude.

2. DOMAIN PARKING OVERVIEW

Parked domain is an unused domain that maps to an ads-portal page, which fetches its ads from a parking service (an advertisement syndication service). Parked domains could either resolve or forward to an ads-portal page served by a parking service. Most of the time, parked domains are hosted at parking services' servers, but parked domains could be hosted at other servers. The ads-content served by a domain parking service forms the main content of the served ads-portal page. Thus, the served ads content, from some parking service, collectively looks like a complete web page. Figure 1 shows screen shots of parked domains from two different parking services. More screen shots of parked domains can be found at http://research.microsoft.com/URLTracer/Ads_on_Parked_

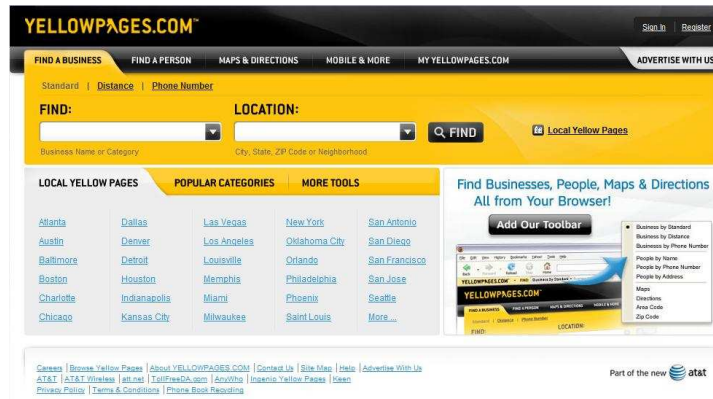


(a)



(b)

Fig. 1. Ads Portal Domains



(a)



(b)

Fig. 2. Confusing Non-Ads Portal Domains

Domains.htm. In contrast, ads-content served by other Internet advertisement syndication services, such as Google AdSense [Google 2008] forms only a section of a web page that has other real content.

Parked domains map to ads-portal pages through several methods. A parked domain could have a third-party URL that fetches the ads content from a parking service; i.e., a parked domain could be frame-based and has a third-party URL that fetches the ads content from a parking service. Also, a parked domain could forward the user request to a parking service which shows the ads content. The forwarding could be at the DNS level, HTTP level, or HTML level. Forwarding at the DNS level [Mockapetris 1987] means that a parked domain name resolves to the IP address of a parking service. Forwarding at the HTTP level means that a parked domain redirects HTTP traffic to a parking service's server through 3XX HTTP reply message [Fielding et al. 1999]. Forwarding at the HTML level means that a parked domain returns HTML content that forwards to a parking service's server through META tag [Raggett et al. 1998].

The ads-portal page pointed to by the parked domain can be in one of two forms: one-click page and two-click page. In one-click page, the sponsored ads are shown immediately

to the user. In two-click page, the user lands into a page that ads subcategories (index page). When the user clicks on one of the shown subcategories, he/she will be shown a page with sponsored links falling under the clicked category.

Recently, domain parking business has pushed the edge and some services started serving, in addition to the ads content, some useful information for the purpose of better monetizing the traffic. However, in this paper when we refer to the typical parking services that provide textual ads content and no other real content. In this paper, we use “parking service” and “advertisement syndication service” interchangeably. Also, when we refer to ads-portal domains, we mean text-based ads-portal domains.

3. DATA SETS COLLECTION

As stated in 1, there are two goals of this research: developing an ads-portal page classifier and exploring several characteristics of ads-portal domains through measurements (ads-portal ratio, typo ratio, ads-portal access ratio, etc). To address these goals, we create five different data sets: COM-Zone, NET-Zone, COM-Trace, NET-Trace, and Positive-Negative-Samples. The following Sections explain the data sets in details.

3.1 Zone Sets

COM-Zone and NET-Zone data sets are extracted from a couple of TLD - Top Level Domain - Zone files: *.com zone file and *.net zone file¹. *.com/*.net zone file consists of all the domains that are registered under *.com/*.net TLD - Top Level Domain². *.com/*.net zone file has ~ 76/12 million domains. We randomly selected 200,000 domains from each of the zone file. For each domain in the sampled sets, we try to download its web content³. COM-Zone/NET-Zone represents the domains that are successfully downloaded from the 200,000 random sample of the *.com/*.net zone file. Table I shows the number of domains in COM-Zone and NET-Zone sets. We create these sampled zone sets for two reasons. First, we use them to create Positive-Negative-Samples data set, which is used in devising the identification methodology. Second, we use them in Section 6 to measure certain characteristics of ads-portal domains.

3.2 DNS Trace Sets

COM-Trace and NET-Trace data sets are extracted from DNS trace files. We collect two-month DNS trace files corresponding to the months of June and July of 2008. The DNS trace files are collected at the name servers of UCI -University of California, Irvine- campus and are sent periodically to our machines. The trace files consist of resource records that correspond to user resolution requests for domains other than *.uci.edu. The main reason of using the trace files is to collect a representative sample of web domains that users, within the campus, browse. Since HTTP requests are typically preceded by a DNS request of A-type or CNAME-type [Mockapetris 1987], we filter out, from the trace files, all the resource records that are not of A-type or CNAME-type. This results in a set of *sim* 15

¹The zone files have been provided by VeriSign(www.verisign.com) during the month of July, 2008.

²VeriSign(www.verisign.com) has the policy of excluding from the zone files all the registered domains that are not associated with some name servers

³When we try to download a sampled domain, we first try to download it with the *www* prefix. If that fails, we try to download it as it is, without the *www* prefix

Table I. Data Sets

Data Set	#Domains
COM-Zone	168,298
NET-Zone	160,554
COM-Trace	89,063
NET-Trace	86,985
Positive-Negative-Samples	2400

millions domains⁴.

From this filtered set, we create two sample sets. The first set is 100,000 random sample of all the **.com* domains found in the filtered traces that are of two levels, such as *yahoo.com*. The second set is sampled in the same way as the first one; the only difference is that we sample **.net* instead of **.com*. Similar to COM-Zone and NET-Zone, we try to download the web content of those sampled domains. COM-Trace/NET-Trace represents the domains that are successfully downloaded from the 100,000 random sample of the (two level) **.com/*.net* trace domains. Table I shows the number of domains in COM-Trace and NET-Trace sets.

The reason for creating COM-Trace and NET-Trace sets is that we want to create sets from the traces that are comparable to the ones created from the TLD zone files. COM-Trace/NET-Trace is a representative set of the accessed, from UCI, two-level **.com/*.net* domains and COM-Zone/NET-Zone is a Representative set of the two-level **.com/*.net* web domains existing in the Internet. Note that, when sampling the traces, we consider the 3-level domains that have the prefix *www* as 2-level domains since most of the time the two-level domains map to the *www* sub-domain; i.e., *anydomain.com* maps to *www.anydomain.com*. Hence, an access to *www.anydomain.com* is mostly considered an access to *anydomain.com*.

3.3 Positive and Negative Samples

Positive-Negative-Samples set consists of 2,400 web domains. Six hundreds of those are positive(ads-portal) samples and the remaining are negative(non-ads-portal). This data set is used to in Sections 4 and 5 to evaluate the effectiveness of several content-based properties and the performance of the identification methodology.

For the negative samples, we collect 1,800 different domains from all Yahoo Directory Top categories [Yahoo 2007] to cover a wide range of different types of web documents. The negative samples are distributed almost equally over the different categories. For the positive samples, the process of collecting them goes through the following steps:

(1) We obtain a list of fourteen well-known parking services and for each, we collect few samples of ads-portal(parked) domains. Most of those samples are retrieved from the parking services' web sites and the remaining are collected from other domains such as *parkquick.com*. This set of ads-portal domains is by no mean comprehensive since it has only few samples for each parking service.

(2) From these samples, we extract the signatures of the fourteen parking services. The signature of a parking service is a regular expression commonly found in the ads-portal(parked domain) pages served by that parking service. Mostly, the signature of a

⁴Note that A/CNAME resource record could correspond to a non-web domain resolution request or a web domain resolution requests that is not initiated by a user/browser. We acknowledge this limitation.

parking service includes the domain name/URL of the parking service. The signatures are extracted manually.

(3) From this small set of ads-portal domains, we create a larger more comprehensive set using the fourteen different signatures. We use our COM-Zone and NET-Zone data sets (see Section 3.1) to find all the domains in these two sets that have any of the fourteen signatures. The results of this step is fourteen different large sets for fourteen different parking services.

(4) For each of the fourteen different parking sets, we randomly select 100 domains and manually inspected them to eliminate the false positives.

(5) Using these manually filtered random sets, we extract 600 ads-portal domains distributed equally, if possible, over the fourteen parking services.

The reason we collect the parked domains samples from many different parking services is that we want our positive set to be comprehensive enough to cover most ads-portal templates and patterns. Note that we cannot rely on parking service signatures as a way of detecting ads-portal domains for the following reasons:

- (1) Relying on signatures means that we cannot identify ads-portal domains fetching ads from syndication services with unknown signatures. In Section 6.3, we show that $\sim 30\%$ of the ads-portal domains detected using our identification methodology are of on unknown signatures.
- (2) Detecting ads-portal domains through signatures involves many false positives. From our manual inspection, we found that many false positives are involved. In fact, one of the signatures lead to more than 20% false positives.

We download the web content for all the domains in the Data Set. When we download the web content of a domain, we download all the embedded files needed to display the HTML index page and treat them as one page. For example, If the index page is frame-based, we download all the documents to which a frame refers. If the domain is forwarding to another domain, we download the web content corresponding to the final destination.

4. CONTENT-BASED PROPERTIES

As we look through different ads-portal domains in Positive-Negative-Samples data set, we observe that they promote their ads-content in similar fashion. Hence, we identify a set of content-based properties/features that distinguish ads-portal domains from others. In the remainder of this Section, we explore several content-based properties and show their effectiveness, in terms of highly increasing/decreasing the ads-portal likelihood, through detailed statistical analyses. Note that in this Section, when we refer to “data set”, we mean Positive-Negative-Samples data set, which is defined in Section 3.3.

4.1 Anchor Text Ratio

The content of an ads-portal domain is mainly ads-related without any real content. Mostly, the ads are shown in the form of lists of textual hyperlinks. Since the main content of an ads-portal page is ads-related anchor text, we would expect most of the characters shown in the page belong to the anchor text. In light of this observation, we define a feature called Anchor Text Ratio that measures the intensity of the anchor text in a web page. Specifically, Anchor Text Ratio of a domain D is defined as follows:

$$\text{AnchorTextRatio}(D) = \frac{\text{Number of Characters in the Anchor Text of } D}{\text{Total Number of NonMarkup Characters in } D}$$

Note that only alpha-numeric characters are counted. Due to the nature of most ads-portal domains, we intuitively think that Anchor Text Ratio of ads-portal domains would be high and distinguishable from many of the non-ads ones. To investigate the effect of Anchor Text Ratio, we plotted several distributions that collectively show the effectiveness of Anchor Text Ratio. The distributions are shown in Figure 3.

Figure 3 - like all other figures in this section - combines two different distributions: domain fraction and posterior probability distributions. In addition, the figure shows the prior probability. All the distributions are computed over Positive-Negative-Samples data set. The x-axis depicts a set of value ranges of Anchor Text Ratio (in Figure 3, the first range refers to the web domains having Anchor Text Ratio values between 0 and 0.07). The bar graph and the left y-axis depicts the percentage of web domains in our data set (Positive-Negative-Samples) that falls into a specific range. For example, the left-most bar in the graph indicates that 11.2% of the web domains in the data set have Anchor Text Ratio values that fall in the range [0, 0.07). The line points and the right y-axis depicts the posterior probability

$$P_p = P\{D \text{ is an AdsPortal Domain} \mid \text{AnchorTextRatio}(D) \in I\}$$

distribution over the value ranges (I 's) of Anchor Text Ratio. The horizontal line and the right y-axis depicts the prior probability of being an ads-portal domain

$$P_a = P\{D \text{ is an AdsPortal Domain}\}$$

, which is equal to the fraction of the ads-portal domains in our data set (Positive-Negative-Samples); i.e., $600/2400 = 0.25$. The prior probability horizontal line is plotted so that the difference between the prior and posterior probabilities would become clearer.

It can be seen in Figure 3 that the posterior probabilities P_p highly vary from the prior probability in almost the full range of the Anchor Text Ratio values. There are only few ranges (for instance, the range [0.35, 0.42)) at which the posterior probability and the prior one are close to each other.

Figure 3 shows how the posterior probability changes when Anchor Text Ratio changes. We can observe from figure that, except for few cases, the ads-portal likelihood increases as Anchor Text Ratio increases. From the figure, we can conclude that a (low) high Anchor Text Ratio value is a good indicator for being (a) an (non-)ads-portal domain.

In Figure 3, we can observe that the posterior probabilities P_p highly vary from the prior probability in almost the full range of the Anchor Text Ratio values. There are only few ranges (for instance, the range [0.35, 0.42)) at which the posterior probability and the prior one are close to each other. To quantify the posterior effect of the feature (how it affects the posterior distribution), we introduce a new metric called Average Posterior Change Ratio *APCR*. The definition of *APCR* is as follows:

$$APCR = \sum_{\forall \text{ Interval } I} P_p \text{ Change Ratio in } I \times \text{Fraction of Domains in } I \quad (1a)$$

where :

$$P_p \text{ Change Ratio in } I = \frac{|P_p \text{ in } I - P_a|}{\text{Max } P_p \text{ Displacement in } I} \quad (1b)$$

and :

$$\text{Max } P_p \text{ Displacement in } I = \begin{cases} 1 - \frac{P_a}{P_p} & \text{if } P_p \text{ in } I > P_a \\ \frac{P_a}{P_p} & \text{otherwise} \end{cases} \quad (1c)$$

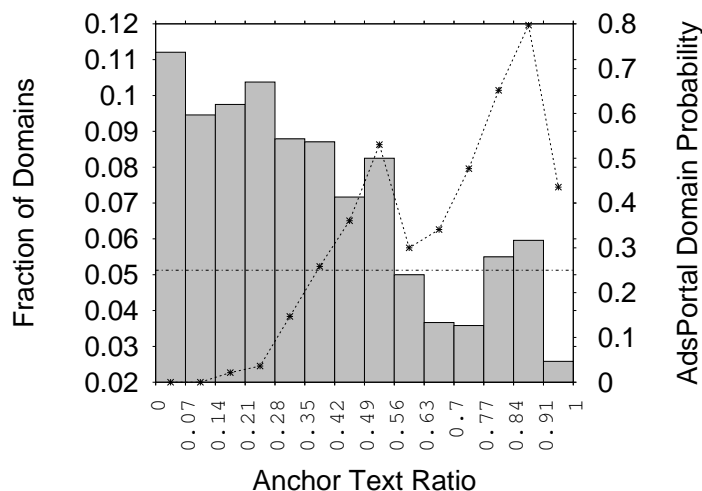
Basically, the Posterior Change Ratio in I measures the relative displacement of the posterior probability P_p from the prior one P_a . For example, the value of 0 indicates 0 relative displacement; i.e., P_p and P_a are equal, and the value of 1 indicates maximum relative displacement; i.e., P_p is displaced from P_a to either ends, 1 or 0. Note that we choose to measure the ratio of displacement instead of the absolute displacement ($|P_p - P_a|$) because the influence of the feature value on the posterior probability is clearer if explained by the ratio instead of the absolute value. For example, the posterior probability of 0 is as distinguishing as the posterior probability of 1; i.e., we can accurately identify ads/non-ads domains with 100% accuracy. If we use the ratio, the Posterior Change Ratio would be 1 in both cases. But if we use the actual absolute displacement ($|P_p - P_a|$), the absolute displacement would 0.25 in the case of 0 and 0.75 in the case of 1. Therefore, the absolute displacement gives these two cases totally different values even though the posterior probability of 0 is as distinguishing and effective as of 1.

The Average Posterior Change Ratio $APCR$ is the average of Posterior Change Ratio's of all the feature value ranges I 's weighted by the ratio of domains in I 's. We choose a weighted average instead of an equally-weighted average (arithmetic mean) because the effect of the Posterior Change Ratio, for an interval I , on the average value should be proportional to the ratio of domains, in I . Note that the weighted average would be more resistant to noises than the arithmetic mean because a noisy interval would generally have a low ratio of domains falling into it. Again, $APCR$ may take a value between 0 and 1. For example, if we have the P_p value the same as the P_a value in all I 's, $APCR$ would be equal to 0. On the other hand, if we have the P_p value equals to either 1 or 0 in all I 's, the $APCR$ would be equal to 1.

As Table II shows, the $APCR$ value for the Anchor Text Ratio is 56% which indicates that the Posterior Change Ratio, on average, is 56%. Note that the optimal value, in terms of distinguishability, is 100%. This value is reasonably high and emphasizes on the effectiveness of Anchor Text Ratio on the posterior probability distribution. The $APCR$ value shows the strong correlation between the Anchor Text Ratio and event of being an ads-portal domain, and that Anchor Text Ratio highly affects ads-portal likelihood.

4.2 Common Link Ratio

We observe that anchor text of different links embedded in ads-portal domains tend to share words. That is, we find some degree of coherence and commonality, in terms of words sharing, among anchor text of ads-links shown in ads domain pages. This degree of coherence varies from strongly to loosely coherent. But generally, there is some degree of coherence in many of the ads-portal domains we come across.

Fig. 3. **Parking Service Typo Ratio**

Feature	APCR
Anchor Text Ratio	56%
Common Link Ratio (N=1)	38%
Common Link Ratio (N=2)	42%
Common Link Ratio (N=3)	47%
Common Link Ratio (N=4)	46%
Number of Hyper-Link Images	51%
Number of Links	47%
Number of Non-Markup AlphNum	50%
Frame-Based	24%
Lengthy Link	39%

Table II. **APCR Values for different Features**

An ads-portal domain that is strongly coherent has most of its anchor text of the ads links sharing the same keyword(s) related to some topic. The anchor text in such a domain may promote for a specific merchandise, e.g. cameras. After further investigation in the parking/syndication service business, we found that a number of parking/syndication service generated contextual ads that were relevant to the domain name of the ads-portal domain or relevant to a set of keywords fixed by the domain name owner of the ads-portal domain. We believe this is the reason why there are many strongly coherent ads-portal domains. Strongly coherent ads-portal domains are good targets for type-in traffic [Wikipedia 2008]. In type-in traffic, the user types a domain name in the address bar and expects to land on a page that shows ads related to the domain name. Since showing related ads may increase the income of the parking/syndication services, probably from type-in traffic, we expect that many ads domains are strongly coherent. Figure 1(a) shows an example of an ads-portal/parked domain page that is strongly coherent. Note that almost all the links in the figure have the word “nail”.

On the other hand, there are loosely coherent ads-portal domains. In a loosely coherent

ads domain, the page presents a set of ads links that represent multiple topics. For example, it could show ads links related to finance and ads links related to technology at the same time. In those domains, there exists some common words in different links but at lesser degree than the strongly coherent ones. Figure 1(b) shows a loosely coherent ads-portal domain. It can be observed from the figure that some links, such as “Airline tickets” and “Mortgage”, do not share words with any other link in the page. But there are other links, such as “Free credit report” and “Credit Card Application”, that share some word(s).

To capture the coherence at various degrees (strong, loose) in ads-portal domains, we define a new feature called Common Link Ratio. Common Link Ratio CLR_N of a domain D is defined as follows:

$$CLR_N(D) = \frac{\# \text{ links sharing words with other } N \text{ links in } D}{\# \text{ links in } D}$$

In the above equation, “sharing words” means sharing one or more. Words in links are stemmed using the Porter stemmer [Porter 1980] and the stop words are removed. We compute Common Link Ratio for different values of N : 1, 2, 3, and 4. Note that we compute Common Link Ratio for different N values to capture different levels of coherence web domains have among their links. Thus, N represents a correlation factor.

Figures 4, 5, 6 and 7 show the distributions (fraction of domains, posterior and prior) of Common Link Ratio with the N values of 1, 2, 3, and 4. We can observe the effect of the correlation as we compare the different intervals of the same figure. The four figures show that as Common Link Ratio value increases, the prevalence of ads-portal domains increases. This increasing trend makes intuitive sense - many parking/syndication services feed ads-portal domains with ads that are relevant to their domain names or some sets of keywords. Consequently, the anchor text in the ads links is correlated and the posterior probabilities have increasing trends in the figures.

Also, we can observe the effect of the correlation as we compare the same intervals in the four figures. If we compare the distributions (bar graph and line points) at the right end intervals of Figures 4, 5, 6 and 7, we can observe that as the correlation factor (N) increases, from 1 to 4, the bar graphs (fraction of domain) decrease and line points values (posterior probability) increase. For example, if we compare the bar graphs and line points at the interval [0.84-91) of the four figures, we can see that as the correlation factor (N) increases, the bar graph decreases and the line point value increases. This implies that for a web domain, the stronger the correlation is, the higher the likelihood of being an ads-portal is.

The four figures show that the posterior distributions largely deviate from the prior ones. To accurately quantify this deviation, we compute $APCR$ value of Common Link Ratio. Table II shows the $APCR$ values of Common Link Ratio, for different N values, which range from 38% to 47%. These $APCR$ values imply that the ads-portal likelihood significantly changes when considering the Common Link Ratio.

4.3 Number of Image Links

Since we are trying to identify textual ads-portal domains, which shows most of their ads using anchor text, we would expect that many of textual ads-portal domains have few image links, if any. That might be the case because parking/syndication services may prefer to reduce their bandwidth costs by reducing the size of the served content. To further

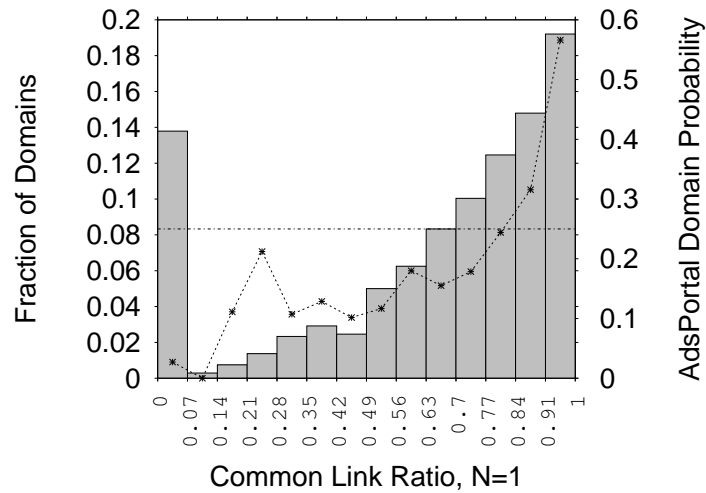


Fig. 4. Parking Service Typo Ratio

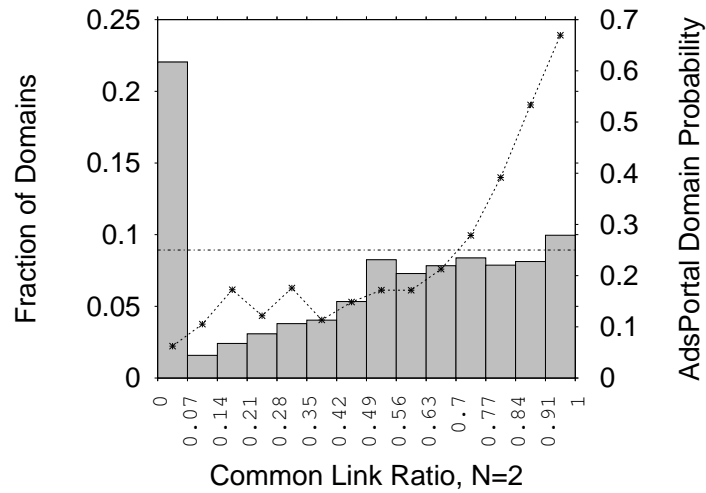


Fig. 5. Parking Service Typo Ratio

investigate the effect of the number of image links, we plot the distributions (bar graph and line points) of the number of image links as shown in Figure 8.

Figure 8 shows, with few exceptions, that the posterior probability has a descending trend, i.e., as the number of image links increases, the posterior P_p probability decreases. This decreasing trend complies with the nature of those textual ads-portal domains as they show their ads in the form of textual ads listing with few images. From Figure 8, we can conclude that few number of image links might be a good indicator that a given domain is an ads-portal. Note that in Figure 8, there is one exception in the range [11,12), where the posterior P_p probability unexpectedly spikes. The reason for that is related to the ads-portal

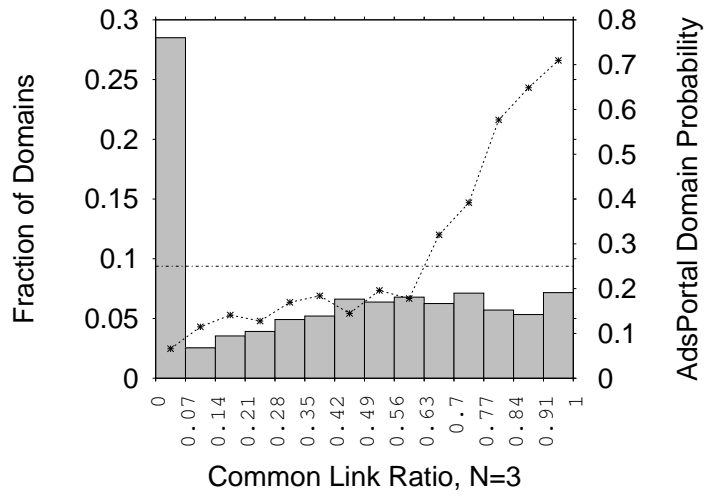


Fig. 6. Parking Service Typo Ratio

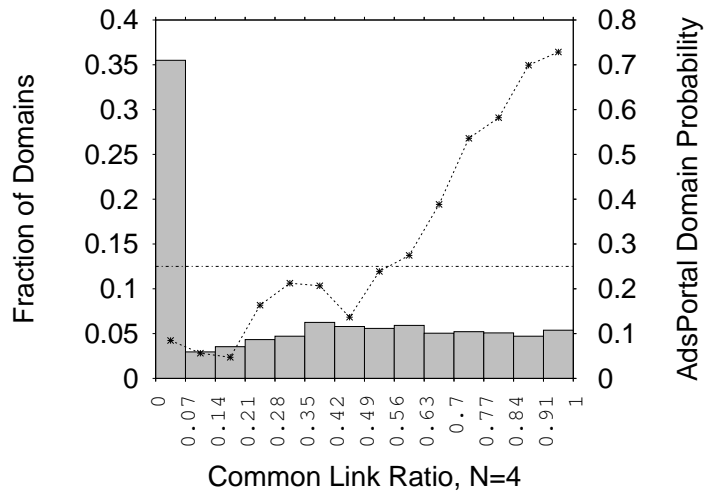


Fig. 7. Parking Service Typo Ratio

pattern produced by one parking service, where it uses many image links as its ads-links.

It can also be observed from the figure that the posterior probability is highly different from the prior one at many of the intervals and to accurately quantify the change in the posterior probability, we compute the *APCR* value of Number of Image Links. Table II shows that *APCR* value of Number of Image Links feature is 51%; i.e., Posterior Change Ratio, on average, is 51%. This shows that the ads-portal likelihood is largely effected when we consider the number of image links.

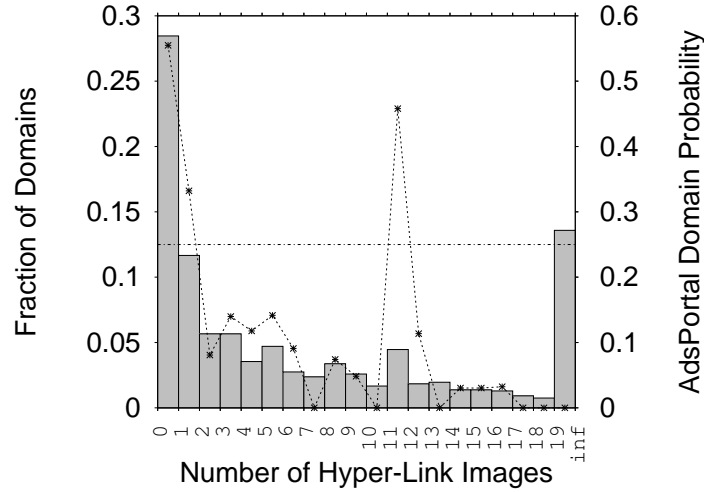


Fig. 8. **Parking Service Typo Ratio**

4.4 Number of Links

An ads-portal domain mostly show ads links and no real content. Thus, the number of links in a domain could be useful in identifying (non-)ads-portal domains. To further explore the usefulness of Number of Links property, in terms of identifying ads-portal domains, we plot the distributions as shown in Figure 9. We can observe from the figure that the ratio of ads-portal domains increases when the the number of links is moderate (between 21 and 70). That is, very few/large number of links is an indicator that a given domain is not an ads-portal.

We can observe from the figure that the posterior probability P_P is highly variant from prior one in most of the range values. To quantify the effect of Number of Links feature on the posterior probability, we compute the $APCR$ value of Number of Links. As shown in Table II, the $APCR$ value is 47%; ie, Posterior Change Ratio, on average, is 47%. That shows how the posterior probability is significantly effected by Number of Links and ,consequently, Number of Links is strongly correlated with the event of being ads-portal.

4.5 Number of Non-Markup Alphanumeric Characters

We observe that ads-portal domains tend to have small content in terms of the amount of alpha-numeric text shown in their pages. One possible justification is that a parking/syndication service does not want to lose the visitor attention by showing him/her so much textual content in ads-portal page. To further investigate this property, we plot the distributions as shown in Figure 10. Mostly, the posterior probability decreases as the Number of Non-Markup Alphanumeric Characters increases. Specifically, the range of values [300, 1500) shows a major increase in P_p . Thus, limited number of non-markup Alphanumeric Characters might be a good indicator that a domain is an ads-portal.

Also, Figure 10 shows that posterior probability P_P drastically varies from the prior one at many of the intervals. We compute $APCR$ value to accurately quantify the effect of the feature on the posterior distribution. Table II shows that the $APCR$ value of Number of

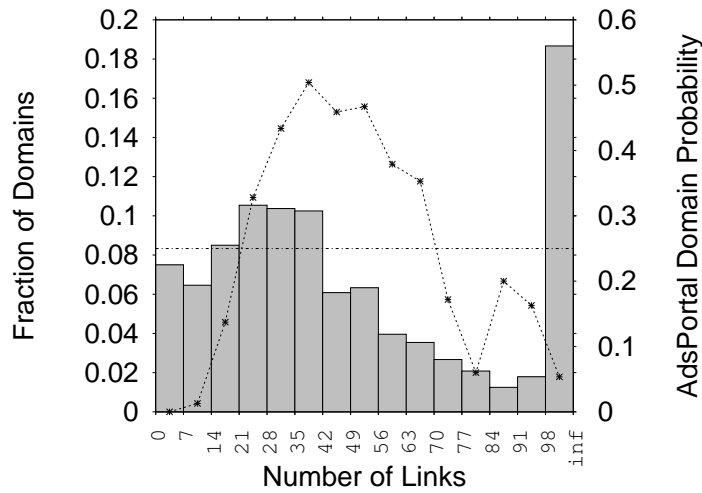


Fig. 9. **Parking Service Typo Ratio**

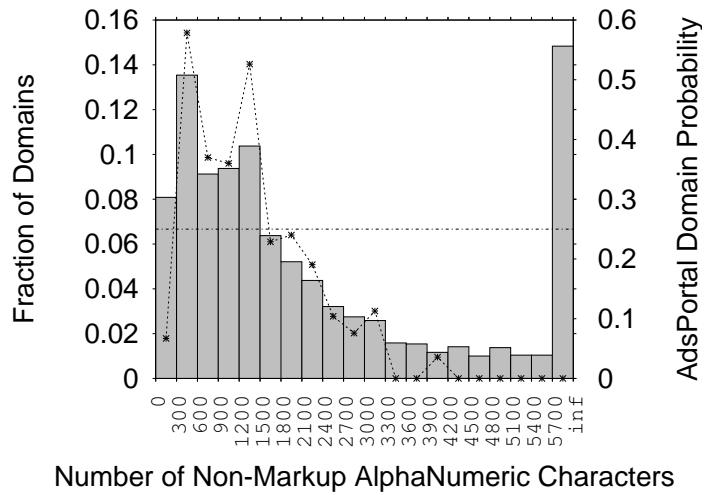


Fig. 10. **Parking Service Typo Ratio**

Non-Markup Alphanumeric Characters is 50%, which indicates that Posterior Change Ratio, on average, is 50%. The *APCR* value suggests that this property significantly changes ads-portal likelihood.

4.6 Frame-Based Domains

We observe that a number of ads-portal domains are frame-based; i.e., they use the frame HTML structure to fetch ads content from parking/syndication services. That naturally comes from how some of the parking/syndication services provide their ads-content: an ads-portal domain obtains the dynamic ads content from the parking/syndication service

using an HTML frame that refers to the parking/syndication service.

We compute the ratio of domains that are frame-based and the resulting posterior probability from our Positive-Negative-Samples data set. A ratio of 10% of the web domains in our data set are frame-based and 70% of those are ads-portal domains. That means that ads-portal likelihood increases from 25% (prior) to 70% (posterior). On the other hand, 90% of the web domains are not frame-based and only 20% of those are ads-portal domains. This shows how Frame-Based feature would significantly change the ads-portal likelihood if the domains are frame-based.

The *APCR* value of Frame-Based feature, as shown in in Table II, is 24%; i.e., the Posterior Change Ratio, on average, is 24%. Even though the *APCR* value of Frame-Based is less than the previous features, it is still influential on the posterior distribution. Note that from Frame-Based feature, the Posterior Change Ratio is high when the feature is true, but Posterior Change Ratio is low when the feature is false. The low value of Posterior Change Ratio, when the feature is false, reduces the *APRC* value to 24%.

4.7 Lengthy Link Domains

We observe that a number of ads-portal domains tend to have the size of their anchor text small, in terms of the number of alpha-numeric characters, per link. One possible justification is that showing many links with lots of textual content may distract the visitors and encourage them to leave the domains without clicking on any of the sponsored-links. The prevalence of short textual links in ads-portal domains is clearly more observable in ads-portal domains that show their ads at two-levels: one level serves as an index page and the other level shows the actual sponsored-links.

To further investigate the effect of the size of the links, we define new boolean feature called Lengthy Link Domains that is set to *True* if the domain has lengthy links and *False* otherwise. Specifically, the feature is set to 1 whenever the page has 3 or more links each of which with anchor text of 30 or more alpha-numeric characters.

A ratio of 65% of the web domains in our Positive-Negative-Samples data set have this feature set to 0 and 36% of those are ads-portal domains. That is, the likelihood of being an ads-portal increases from 25% (prior) to 36%(posterior). On the other hand, 35% of the web domains have their feature values set to 1 and only 96% of those are non-ads-portal domains. That is, the likelihood of not being an ads-portal increases from 75% (prior) to 96%(posterior). The *APCR* value of Lengthy Link feature, as shown in in Table II, is 39%; i.e., the Posterior Change Ratio, on average, is 39%. Clearly, these measures indicate a strong correlation between Lengthy Link domains and the posterior probability. We acknowledge that the thresholds used to set Lengthy Link domains feature are subjectively selected and there might be room for improvement.

5. USING CLASSIFIERS TO IDENTIFY ADS-PORTAL DOMAINS

In the previous section, we analyze several properties/features of ads-portal domains and show how specific value ranges of those features could highly increase/decrease the ads-portal likelihood. Moreover, we show the effectiveness of those features in terms of their influence on the ads-portal likelihood using a defined metric called *APCR* (Equality 1a, Section 4.1). However, using each feature individually may not be helpful or may lead to a large range of false positives. For example, Anchor Text Ratio property (see Section 4.1), which is our most effective feature in terms of changing the ads-portal likelihood (highest *APCR* value), may lead to a huge range of false positives if we solely rely on it to identify

ads-portal domains. For example, let us assume we use a simple condition that identifies a given domain as ads-portal if it has Anchor Text Ratio ≤ 0.42 ; otherwise, it identifies the given domain as non-ads-portal. That is, if Anchor Text Ratio value of a given domain falls into the value ranges that increases the ads-portal likelihood (see Figure 3), we consider the domain as ads-portal; otherwise, we consider it as non-ads-portal. In this case, the ratio of false positive is 49.5%, which is very high. Even if we limit the condition to the value range that has the highest ads-portal likelihood, range $[0.84, 0.91)$, the false positive ratio will be 20% and we will miss lots of ads-portal domains that have Anchor Text Ratio falling into other value ranges.

Since relying solely on one feature may lead to many false positives, it would be more effective to combine multiple features and leverage their discriminative capabilities. One way to combine the features is to consider them as one feature vector and feed them to a machine learning classifier. Thus, we combine the features described in Section 4 into one feature vector of 10 features (for Common Link Ratio, we consider N values of 1, 2, 3 and 4) and compare several classification algorithms. Particularly, we try the following classifiers: Random Forest [Breiman 2001] (with 20 trees), Decision Tree (C.45) [Quinlan 1993], Support Vector Machines (SVM) [Drucker et al. 1999; Joachims 2001], K-Nearest Neighbor [Mitchell 1997], RIPPER [Cohen 1995], Random Tree [Almishari 1995], Bayesian Networks [Mitchell 1997], Neural Networks [Mitchell 1997], Decision Table [Kohavi 1995]. In addition, we try some optimization techniques to classifiers; namely, we try boosting (with 10 iterations) of Decision Tree [Freund and Schapire 1995; Quinlan 1996], bagging (with 20 trees) [Breiman 1996; Quinlan 1996] of Decision Tree, and boosting (with 10 iterations) of Decision Stump [Iba and Langley 1992; KAWAKITA et al. 2005].

We train the classifiers on Positive-Negative-Samples data set, which is explained in Section 3.3. We use the WEKA java classes [Witten and Frank 2005] to run and test different machine learning classification algorithms. One reliable way of evaluating the accuracy of different classifiers is to use the ten-fold cross validation method [Mitchell 1997]. In the ten-fold cross validation, we randomly divide the data set into 10 sets of equal size, perform 10 different training/testing steps in which each step consists of training a classifier on 9 sets, and then testing it on the remaining set. We take the average of the results as the accuracy of the classifier.

We use the positive/negative precision and recall values to summarize the performance of a classifier. The positive (negative) recall value shows the fraction of the ads-portal domains (non-ads-portal domains) that are correctly identified by the classifier among all ads-portal (non-ads-portal) domains fed to the classifier. The positive (negative) precision shows the fraction of ads-portal (non-ads-portal) domains among the set of domains classified as ads-portal (non-ads-portal) domains by the classifier.

Table III shows the performance results of the all of the above classifiers. We can see from the table that the Random Forest classifier shows the superior performance in all of the four metrics: Positive Recall, Positive Precision, Negative Recall, and Negative Precision. Note that the bagging classifier performance is really close to the Random Forest one. The reason is that both classifiers are almost the same except that in Random Forest, there is some randomness involved in the selection the features. The exact description of bagging, Random Forest and the other classifiers is out of the scope of this paper. For more information about the mentioned classifiers and how text classification works, the

Table III. Performance Results of Several Classifiers - Feature vector consists of only the 10 content-based features

Classifier	Positive Recall	Positive Precision	Negative Recall	Negative Precision
Random Forest	0.94	0.91	0.97	0.98
SVM (C=1)	0.78	0.84	0.95	0.93
Nearest Neighbor	0.87	0.81	0.93	0.95
K-Nearest Neighbor (K=10)	0.87	0.80	0.93	0.96
Unpruned C4.5 Decision Tree	0.91	0.88	0.96	0.97
Pruned C4.5 Decision Tree	0.91	0.89	0.96	0.97
Boosting C4.5 Decision Tree	0.74	0.84	0.95	0.92
Bagging C4.5 Decision Tree	0.93	0.90	0.96	0.97
RIPPER	0.92	0.88	0.96	0.97
Random Tree	0.86	0.84	0.95	0.95
Bayes Net	0.83	0.85	0.95	0.94
Neural Net	0.85	0.82	0.94	0.95
Decision Table	0.88	0.90	0.97	0.96
Boosting - Decision Stump	0.74	0.84	0.95	0.92

reader is advised to refer to the above references and [Mitchell 1997; N. Cristianini and J. Shawe-Taylor 2000].

Even though the evaluation measures of the Random Forest classifier are good, we need to enhance the performance, especially for the ads-portal Precision, which makes the classifier wrong in 9% of the times it declares a domain as an ads-portal. As a step to enhance the performance, we add another set of features that boosts the discriminative strength of all the classifiers. These added features are words and phrases (“for sale”, “sponsored listings”, etc.) that commonly/uncommonly exist in ads-portal domains. We add these words/phrases to the feature vector as boolean features corresponding to the occurrence of these words/phrases. In selecting these words/phrases, we first identify an initial set of words/phrases that commonly or uncommonly exist in ads-portal domains based on our personal observation. Then, we eliminate from this set the ones that do not affect the ads-portal likelihood. We end up with 56 different words and phrases. The average *PACR* of these words/phrases is 15% and that shows their influence on ads-portal likelihood.

Table IV shows the evaluation results after adding these word-based features. Clearly, adding them improves the performance of nearly all the classifiers, especially in ads-portal precision and recall values. There are some domains that have similar look-and-feel to ads-portal domains, for example search directories. Before adding the word-based features, such confusing non-ads domains may have similar feature values to the ads-portal domains, and consequently they might be misclassified. The addition of the word-based features helps the classifiers to recognize these confusing domains and accurately classify them. For example, Random Forest classifier, before adding the word-based features, misclassifies *www.yellowpages.com* as an ads-portal. This domain is confusing in the sense that it is non-ads-portal domain that has similar look-and-feel to those of ads-portal ones; the domain *www.yellowpages.com* is an Internet Yellow Pages and search directory. After adding the word-based features, Random Forest classifier is able to correctly classify *www.yellowpages.com* as a non-ads-portal domain. Again, Random Forest shows the superior performance after the addition of the word-based features. Note that many of the other classifiers have comparable performance. Since Random Forest classifier, shown in Table IV, shows the best performance, it is the classifier that we use in our measurement in Section 6. In Section 6.6, we further validate the accuracy of this classifier on different

Table IV. **Performance Results of Several Classifiers - Feature vector consists of the 10 content-based features and the word-based features**

Classifier	Positive Recall	Positive Precision	Negative Recall	Negative Precision
Random Forest	0.98	0.97	0.99	0.99
SVM (C=1)	0.96	0.95	0.98	0.99
Nearest Neighbor	0.95	0.92	0.97	0.98
K-Nearest Neighbor (K=10)	0.96	0.90	0.97	0.99
Unpruned C4.5 Decision Tree	0.97	0.95	0.98	0.99
Pruned C4.5 Decision Tree	0.97	0.95	0.98	0.99
Boosting C4.5 Decision Tree	0.92	0.92	0.97	0.97
Bagging C4.5 Decision Tree	0.97	0.96	0.99	0.99
RIPPER	0.96	0.95	0.98	0.99
Random Tree	0.87	0.85	0.95	0.96
Bayes Net	0.96	0.93	0.98	0.99
Neural Net	0.96	0.95	0.98	0.99
Decision Table	0.92	0.96	0.99	0.97
Boosting - Decision Stump	0.92	0.92	0.97	0.97

Table V. **Performance Results of Several Classifiers - Feature vector consists of only word-based features**

Classifier	Positive Recall	Positive Precision	Negative Recall	Negative Precision
Random Forest	0.88	0.92	0.98	0.96
SVM (C=1)	0.83	0.96	0.99	0.94
Nearest Neighbor	0.86	0.91	0.97	0.95
K-Nearest Neighbor (K=10)	0.73	0.95	0.99	0.92
Unpruned C4.5 Decision Tree	0.86	0.88	0.96	0.95
Pruned C4.5 Decision Tree	0.84	0.93	0.98	0.95
Boosting C4.5 Decision Tree	0.65	0.96	0.99	0.90
Bagging C4.5 Decision Tree	0.82	0.92	0.98	0.94
RIPPER	0.83	0.91	0.97	0.95
Random Tree	0.83	0.86	0.96	0.94
Bayes Net	0.81	0.93	0.98	0.94
Neural Net	0.87	0.92	0.97	0.96
Decision Table	0.76	0.96	0.99	0.92
Boosting - Decision Stump	0.65	0.96	0.99	0.90

data sets. The accuracy results are close to the one shown in Table IV.

For further comparison among classifiers with different feature sets, we train the above classifiers on feature vectors that consist of only word-based features, which are described above. Table V shows the performance results of all the classifiers. We can see from the table that there is no clear winner. But, if we consider the total sum of all the four performance metrics as a way to compare the classifiers, Random Forest will be the best. This Random Forest classifier is worse than Random Forest classifier shown in Table IV in all the four performance metrics. Also, this Random Forest classifier is worse than the one in Table III in terms of the overall sum of the four metrics.

6. MEASUREMENT RESULT

Several studies [Wang et al. 2006; McAfee 2007] show the existence of ads-portal and typo-squatting domains in the Internet. However, we do not know how many of ads-portal domains are in the Internet. Do they represent a trivial or a major ratio of the web domains? What percentage of ads domains are typo domains? Do Internet users access ads-portal

domains? Are typographical errors the main reason why users access ads domains? This Section explains several experiments that address the above questions. Knowing the ratio of ads-portal domains helps in better characterizing the Internet documents and may indicate the degree of success of this type of business. If the business is successful, more effort to better monetize the traffic is worthwhile. In addition, finding the ratios of typo-squatting ads domains is important in measuring the extent of the problem and knowing if some counter measures are needed to discourage the spread of typo-squatting practice.

6.1 Number and Ratio of Ads-portal Domains in the Internet

Table VI shows the number of ads-portal domains and their corresponding ratios, relative to the data set size, in the zone data sets. Apparently, the number of ads-portal domains and their ratios are high. The ads-portal domains represent 30.5/26.6% of the COM-Zone/NET-Zone data sets. This indicates that approximately quarter of the web domains that fall into the (two-level) *.net and *.com domains are ads-portal. These ratios show a large prevalence of ads-portal domains in the Internet. They also indicate that this type of business and monetization of traffic is to some extent successful as ads-portal domains represent a major ratio of the Internet domains. Note that the *.net and *.com represents a major portion of the Internet web domains, if not the largest.

6.2 Number and Ratio of Accessed Ads-portal Domains

Table VII shows the numbers and ratios of ads-portal domains found in the trace sets: COM-Trace and NET-Trace. Although these numbers are not as large as the ones in the zone sets, they represent considerable ratios of the accessed domains. Since ads domains only show ads listings, we initially expected that their access ratios would be extremely small. Unexpectedly, the computed ratios of ads-portal domains represent around 5% of the domains accessed from UCI to the (two-level) *.net and *.com domains. These ratios show that ads-portal domains are successful, with respect to UCI traces, in attracting many visitors.

At this point, it would be helpful to understand what are the factors that lead to this high access ratio. We think that there are three possible reasons leading to the high access ratio of ads-portal domains: Direct Search, Search Engines, and Typo-squatting. First, we believe that some users access ads-portal domains because they are performing Direct Navigation [Wikipedia 2008]; i.e., they are searching for some topics by bypassing search engines and directly typing in the address bars of their browsers some generic names related to their topics, hoping to find related content. An example of such a generic name ads-portal domain is *www.nail.com* as shown in Figure 1(a). Second, we think that search engines are contributing to the high access ratio of ads-portal domains. Wang et al. [2007] show how web-spammers trick the search engines to promote for URLs that redirect to ads-portal domains. In addition, there might be some ethical, or white-hat, SEO (Search Engine Optimization) techniques that ads-portal domains are using. Finally, we believe that some users access ads-portal domains because the domains happen to be typos. For example, *hotmail.com* is an ads-portal domain that is one error away from *hotmail.com*. In Section 6.5, we study the much the typo-factor is contributing to the high access of ads-portal domains. For the other two factors, we leave their analyses as part of our future work.

Table VI. Numbers and ratios of ads-portal domains in the zone data sets

Data Set	#Ads Domains	%Ads Domains
COM-Zone	51,266	30.5%
NET-Zone	42,763	26.6%

Table VII. Numbers and ratios of ads-portal domains in the DNS trace data sets

Data Set	#Ads Domains	%Ads Domains
COM-Trace	3,998	4.5%
NET-Trace	4,745	5.4%

Table VIII. Top ten parking services in terms of their shares of ads-portal domains in the Internet - *.com and *.net

Parking Service	Ratio-Zone
GoDaddy.com	41.3%
DomainSponsor.com	12.7%
RevenueDirect.com	3.6%
Hitfarm.com	3%
Fabulous.com	2.2%
Sedo.com	2.2%
NameDrive.com	1.9%
TrafficZ.com	1.24%
Parked.com	1%
Google.com ⁵	0.5%

6.3 Parking Service Distribution in the Internet

An ads-portal domain fetches its ads content from a syndication/parking service. To understand how different syndication/parking services are contributing to the large number of ads-portal domains in the Internet, we collect the signatures of fourteen different well-known parking services. The signatures are basically the same as the one we use in Section 3.3. Given those signatures, we match each of the ads-portal domains, detected in our COM-Zone and NET-Zone data sets, to the corresponding parking service.

Table VIII shows the top ten parking services in terms of their relative shares in the set of detected ads domains in both COM-Zone and NET-Zone. Apparently, the shares of different parking services are not equal. *GoDaddy.com* has the largest share, followed by *DomainSponsor.com* parking service. In fact, these two parking services are responsible of $\sim 54\%$ of the ads-portal domains. Many of the remaining are of comparable ratios and few has small ratios.

We find that 30.2% of the ads-portal domains are of unknown parking service signatures. This shows the generality of our identification methodology; i.e., we train our classifier on fourteen ads-portal templates, and then we are able to detect many ads-portal domains served from unknown syndication/parking services. These measures suggest that *GoDaddy.com* is probably the most dominant parking service. Note that *GoDaddy.com* is also a domain name registrar and it parks all of the domains that customers register if the registered domains do not have any name servers. This automatic domain parking of the newly registered domains might be the reason why *GoDaddy.com* is the most dominant parking service.

Table IX. Numbers and ratios of typos in ads-portal domains found in COM-Zone-5000 and NET-Zone-5000

Data Set	#Ads Typo Domains	%Ads Typo Domains
COM-Zone-5000	1,761	35.2%
NET-Zone-5000	1,394	27.9%

6.4 Typo-Squatting Domains in the Internet

To know the ratio of typo-domains, we must first know how we can identify typos. If we have a comprehensive list of target domains, we can use distance functions such as edit distance[Gusfield 1998] to identify typo domains. Unfortunately, we do not have a comprehensive list of all possible target domains. Even if we have it, we need a very efficient implementation of distance functions so we can efficiently run them over a large number of target domains. To avoid such complications, we resort to third-party typo identification services. Specifically, we use the well-known typo correction (spelling suggestion) services that are provided by Google [Google 2006] and Yahoo [Yahoo 2008]. Therefore, if a domain is corrected by either Google or Yahoo typo correctors, we consider the domain as a typo.

Since the typo correction services impose limitations on the number of typo correction queries per day, we take 5000 random samples of ads-portal domains found in each of the zone sets, COM-Zone and NET-Zone, and that results in two random sets of ads-portal domains: COM-Zone-5000 and NET-Zone-5000. Table IX shows the number of typos and their corresponding ratios in COM-Zone-5000 and NET-Zone-5000. A ratio of 35.2%/27.9% of ads-portal domains in COM-Zone-5000/NET-Zone-5000 are typo domains (according to our definition of typo). In fact, the measures in Table VI and Table IX imply that 10.7/7.4% of the (two level) **.com/*.net* domains are typo-squatting domains⁶. The results show that typo domains represent a major portion of the ads-portal domains in **.com* and **.net* and, potentially, in the Internet. In other words, typo-squatting as a practice is highly contributing to the large number of ads-portal domains in the Internet and highly abusing the advertisement syndication business.

Note that our finding about typo-squatting is different from the ones in [McAfee 2007; Banerjee et al. 2008; Wang et al. 2006] which show that many typo-squatting domains exist but do not show what ratio of ads-domains they represent. The $\sim 7\%$ difference between the ratios in COM-Zone-5000 and NET-Zone-5000 shows that typo-squatters are more interested in registering under **.com* than **.net*. One possible reason is that typo-squatters find more popular domains in the **.com* zone than the **.net* zone.

For each of the fourteen parking services *ParkService_i*, we find the ads-portal domains in COM-Zone and NET-Zone parking with *ParkService_i* and how many of those domains are typo domains (using our typo definition). The ratios of these typos are shown in Figure 11. As shown in the figure, the typo ratios range from 16% to 55.8%. The mean of the typo ratios is 38.6% and the standard deviation is 11.2%. Apparently, for most of the parking services, typo-squatting domains represent considerable ratios of the ads-portal domains parking with them. In the figure, ParkingSpa.com has the highest typo ratio(55.8%) and Fabulous.com has the lowest typo ratio(16%).

⁶From Table VI and Table IX, $30.5\% \times 35.2\% = 10.7\%$ and $26.6\% \times 27.9\% = 7.4\%$



Fig. 11. **Parking Service Typo Ratios (Zone)** - In this figure, the typo ratio of a parking service $ParkService_i$ is the number of typo ads-portal domains in COM-Zone and NET-Zone parking with $ParkService_i$ over the total number of ads-portal domains in COM-Zone and NET-Zone parking with $ParkService_i$

6.5 Typo-Squatting Domains in the Traces

Similar to Section 6.4, we find the typos in the trace files, COM-Trace and NET-Trace sets. Table X shows the number of typo ads-portal domains and their ratios, relative to the number of ads-portal domains in the sets. The table shows that 46.2/40% of ads portal domains in COM-Trace/NET-Trace are typos (according to our typo definition in Section 6.4). Apparently, these ratios are high and suggest that typo-squatting, with respect to UCI campus, highly contributes to the number of accessed ads-portal domains.

In fact, the measures in Table VII and Table X imply that 2.1/2.2% of the (two level) $*.com/*.net$ accessed domains, from UCI, are typo-squatting⁷. These ratios altogether with the ones in Section 6.4 suggest that typo-squatting, with respect to UCI, is successful in attracting many users.

For each of the fourteen parking services $ParkService_i$, we find the ads-portal domains in COM-Trace and NET-Trace sets parking with $ParkService_i$ and how many of those domains are typo domains (using our typo definition). The ratios of these typo are shown in Figure 12. As shown in the figure, the typo ratios range from 25% to 62.5%. The average of the typo ratios is 42.6% and the standard deviation is 15.7%. Note that there is not any ads-portal domain found in the trace that has ParkingPanel.com's signature. Apparently, parking services, with respect to UCI campus, are making a considerable advantage of users typographical errors, with Hitfarm.com having the highest typo ratio(62.5%) and DomainParking.com having the lowest typo ratio(25%).

⁷From Table VII and Table X, $4.5\% \times 46.2\% = 2.1\%$ and $5.4\% \times 40\% = 2.2\%$

Table X. Numbers and Ratios of typos in ads-portal domains found trace data sets

Data Set	#Ads Typo Domains	%Ads Typo Domains
COM-Trace	1,848	46.2%
NET-Trace	1,902	40%

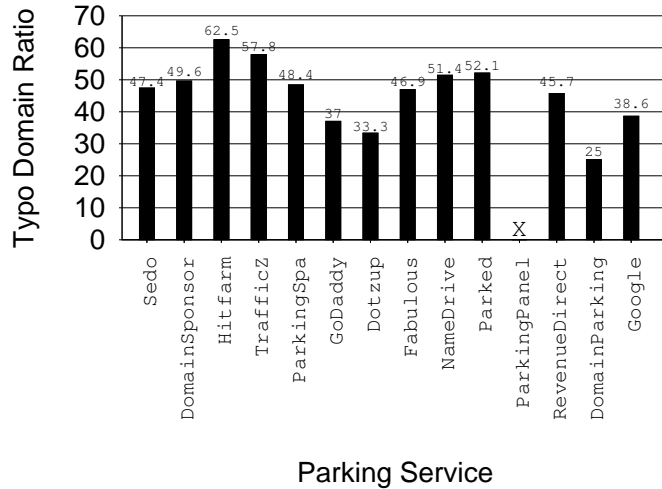


Fig. 12. **Parking Service Typo Ratios (Trace)** - In this figure, the typo ratio of a parking service $ParkService_i$ is the number of typo ads-portal domains in COM-Trace and NET-Trace parking with $ParkService_i$ over the total number of ads-portal domains in COM-Trace and NET-Trace parking with $ParkService_i$

Table XI. Accuracy results of the ads-portal domains detected in the four data sets

Data Set	Accuracy
COM-Zone	98%
NET-Zone	96%
Com-Trace	95%
NET-Trace	96%

6.6 Accuracy Verification

To verify the classifier accuracy of the detected ads-portal domains in the four sets: Com-Zone, Net-Zone, Com-Trace and Net-Trace, we performed manual verification. For each of these four sets, we random sampled (around 50 samples) the set of its detected ads-portal domains and manually inspected the accuracy. The accuracy results of the detected ads-portal domains in the four data sets are shown in Table XI. The accuracy is high and close to the one shown in Section5. This shows that with different data sets, the accuracy of our classifier is almost the same.

7. FUTURE WORK

In Section 0??, we show that the share of a parking service in the set of accessed ads-portal domains may not be proportional to its share in the set of registered ads-portal domains.

This opens the question about what could increase/decrease the incoming traffic to a parking service and a parked domain. As part of future work, we try to answer this question in details. We want to answer this question from two different perspectives: the parking service perspective and the domainer perspective. That is, we want to know what the parking service can legitimately do to increase its traffic, and what the domainers can legitimately do to increase their profit from parked domains. Also, in Section 6, we study the typo factor and show that typos represent a considerable ratio of the accessed ads-portal domains. As part of our future work, we want to study how the direct search (type-in traffic) and search engines are increasing/decreasing the access to ads-portal domains.

Even though many of the typo-squatting domains map to ads-portal domains, we observe other forms of typo-squatting: typo-squatting domains forwarding to other non-ads domains and typo-squatting domains serving malicious content. In fact, There has been an incident that a typo-squatting domain of *www.google.com* was providing malware to its visitors [F-Secure 2005]. We would like to work on a framework that discourages access to the typo-squatting domains once and for all.

8. RELATED WORK

There is an enormous amount of work done in this area of web classification and mining. One of the most relevant work to ours is the one in [Ntoulas et al. 2006] for web spam detection. Ntoulas et al. [2006] develop a binary classifier that identifies web spam pages from their content. A set of content-based heuristics for identifying are web spam pages is identified. Then, these heuristics are combined into a machine learning decision tree [Quinlan 1993] classifier. The classifier reaches 86.2/91.1% recall/accuracy values after boosting [Freund and Schapire 1995] it. Even though this work solves different problem, we developer our classifier in similar fashion. That is, we identify heuristics, verify the effectiveness of these heuristics by drawing their distribution, and finally, combine them into a machine learning classifier.

Esfandiari and Nock [2005] study Advertisement filtering. They propose a methodology to filter out ads-related URLs from a web page through weighted majority algorithm. So, their methodology works at the links level- i.e, identifying if a URL is an ads URL. Whereas, our methodology works at the page level and identifies if the whole page is an ads-portal. Similarly, Kushmerick [1999] proposes a methodology based on inductive learning that automatically removes advertisement images from pages before they are downloaded. However, their methodology is dedicated to removing ads images but in our case the ads are mostly textual links. In addition to that, the methodology treats each image independently where we treated the whole page as one unit. In terms of existing components, there is a Mozilla Firefox extension called AdsBlock [Mozdev 2008] that blocks and removes unwanted content based on filters set by the user. Two types of filters are offered: simple (simple string of text) and regular expression. There is no automatic way of detection,so user intervention is required. Moreover, it treats each link independently but our system treats the whole page as one unit.

In [Wang et al. 2007], one type of web spam - redirection spam - is studied and analyzed. Redirection spam refers web spam URLs that redirect the URLs to spammer-controlled domains, which are mostly in the form of ads-portal. The authors propose a five layer double funnel model that shows how the web redirection spam works. The authors shows important domains at each layer and their related characteristics. Those findings could be

helpful for search engine ranking algorithms to be more robust against spam. This work shows one way of abusing the ads-portal domains using the redirection spam. Our work along with the findings of [Wang et al. 2007] can be used by search engines to help in degrading the web spam ranking in the search results.

Typo-squatting is studied in [Wang et al. 2006; Banerjee et al. 2008; McAfee 2007]. These studies reach the conclusion that many typo-squatting domains are registered and exist in the Web. But these studies do not show how often typo-squatting domains are accessed and how much of the ads-portal domains are typo-squatting. Wang et al. [Wang et al. 2006] show that a large number of typo-squatting domains exist (registered) and a large number of those typo-squatting domains are parked with few parking services that serve ads on those domains. The authors identify parked domains by checking if the third-party URL refers to a parking service, essentially similar to the parking signatures. But for us, we use a machine learning classifier that enables us to detect more parked domains regardless of the parking service. Wang et al. [2006] implement a tool called “Strider URL Tracer” that displays the third-party URLs and helps the trademark owners to check if there are typo-squatting domains of their domains by automatically generating and scanning typo domains. But the problem is that it does not automatically detect which of the generated and scanned typo domains are typo-squatting domains. A manual examination is needed.

Banerjee et al. [2008] study the extent of typo-squatting. For 900 well-known domains, they generate around 3 millions similar URL variations and then investigate to see which are registered phony/typo-squatting domains. They find that typo-squatting domains exist at a large extent. Also, they find that the most of the typo-squatting domains are of one character variation of the original target domains. McAfee [2007] studies the prevalence of typo-squatting. For 2,771 target domains, 1.9 million different single error typos have been generated. In the typo set, 127,381 suspected typo-squatting domains have been identified. Unlike our machine learning way of identifying parked domains, McAfee uses the existence of a parking service signature (URLs, pieces of text) in the content of the domain as a way to identify typo-squatting domains. Also, McAfee has equipped its extension site advisor [McAfee 2008] with the capabilities of identifying typo-squatting domains. The extension would show a yellow color if the site is a typo-squatting site with no risk. If the site is risky than other red color would be shown.

9. CONCLUSION

A textual ads-portal domain refers to a web domain that shows advertisement in the form of ads listing and no real content. The ads content in an ads-portal domain is served by a third-party syndication service. Ads-portal domains are useful in showing related ads content to users performing direct search. However, ads-portal domains are misused in at least two ways: typo-squatting and web spamming.

In this paper, we develop a machine-learning-based classifier to identify ads-portal domains. The features of the classifier are extracted from the web content of the domain. In developing the classifier, we first create negative and positive samples set. Then, we identify set of features that are effective in distinguishing ads-portal domains. Finally, we combine these features along with other keyword-based features into a machine learning classifier. The resulting classifier has 97% accuracy in identifying ads-portal domains. Our identification methodology represents a step towards better mining and categorizing the web domains. Also, it can be helpful to search engines ranking algorithms, helpful in

identifying web spams that redirects to ads-portal domains, and used to discourage access to typo-squatting domains.

We use this classifier along with Internet Zone files for **.com* and **.net* to measure the prevalence of ads-portal domains in the Internet and to find the ratio of ads-domains that are typo-squatting. We find that 30.5/26.6% of **.com/*.net* domains are ads-portal domains and 30.16/22.3% of ads-portal domains in the **.com/*.net* zones are typos. These numbers show the prevalence of both ads-portal and typo-squatting domains in the Internet. In addition, we use the classifier along with DNS trace files to estimate how often Internet users visit ads-portal domains and typo ads-portal domains. It turns out that 4.5/5.4% of the **.com/*.net* domains found in the trace files are ads-portal domains and 24.7/22.4% of **.com/*.net* ads-portal domain in the traces are typos. These numbers show that ads-portal domains and typo-squatting domains are successful, with respect to our traces, in attracting many users to them.

REFERENCES

- ALMISHARI, M. 1995. Random Tree. In *Proceedings of the 9th International Conference on Machine Learning*.
- BANERJEE, A., BARMAN, D., FALOUTSOS, M., AND BHUYAN, L. N. 2008. Cyber-Fraud is One Typo Away. In *Infocom 2008 mini-conference*.
- BREIMAN, L. 1996. Bagging predictors. *Machine Learning* 24, 2, 123–140.
- BREIMAN, L. 2001. Random Forests. *Machine Learning* 45, 1, 5–32.
- COHEN, W. 1995. Fast Effective Rule Induction. In *Proceedings of the 12th International Conference on Machine Learning*.
- DRUCKER, H., VAPNIK, V., AND WU, D. 1999. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks* 10, 5, 1048–1054.
- ESFANDIARI, B. AND NOCK, R. 2005. Adaptive Filtering of Advertisements on Web Pages. In *In Proc. of International World Wide Web Conference (WWW)*.
- F-SECURE. 2005. Google.com installed malware by exploiting browser vulnerabilities. <http://www.f-secure.com/v-descs/google.shtml>.
- FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK, H., MASINTER, L., LEACH, P., AND BERNERS-LEE, T. 1999. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616.
- FREUND, Y. AND SCHAPIRE, R. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*.
- GOOGLE. 2006. Google SOAP Search API. <http://code.google.com/apis/soapsearch/>.
- GOOGLE. 2008. Google adsense. <http://www.google.com/adsense>.
- GUSFIELD, D. 1998. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press.
- IBA, W. AND LANGLEY, P. 1992. Induction of one-level decision trees. In *Proceedings of the 9th International Conference on Machine Learning*.
- JOACHIMS, T. 2001. A statistical learning model of text classification with support vector machines. In *Proceedings of the 24th ACM International Conference on Research and Development in Information Retrieval*. ACM Press.
- KAWAKITA, M., MINAMI, M., EGUCHI, S., AND LENNERT-CODY, C. E. 2005. An introduction to the predictive technique AdaBoost with a comparison to generalized additive models. In *Fisheries research*.
- KOHAVI, R. 1995. The power of decision tables. In *Proceedings of the European Conference on Machine Learning*.
- KUSHMERICK, N. 1999. Learning to remove Internet advertisements. In *The 3rd International Conference on Autonomous Agents*.
- MCAFEE. 2007. McAfee’s Study of Typosquatting. www.mcafee.com/typosquatters.
- MCAFEE. 2008. McAfee SiteAdvisor. <http://www.siteadvisor.com/>.
- MITCHELL, T. 1997. *Machine Learning*. McGraw Hill.
- MOCKAPETRIS, P. 1987. *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION*. RFC 1035.
- MOZDEV. 2008. AdBlock. <http://adblock.mozdev.org/>.

- N. CRISTIANINI AND J. SHAWE-TAYLOR. 2000. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA.
- NTOULAS, A., NAJORK, M., MANASSE, M., AND FETTERLY, D. 2006. Detecting Spam Web Pages through Content Analysis. In *In Proc. of International World Wide Web Conference (WWW)*.
- PORTER, M. 1980. An algorithm for suffix stripping. *Program* 14, 3, 130–137.
- QUINLAN, J. 1993. *c4.5: Programs for Machine Learning*. Morgan Kaufmann.
- QUINLAN, J. R. 1996. Bagging, boosting, and c4.5. In *13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference*.
- RAGGETT, D., HORS, A. L., AND JACOBS, I. 1998. *HTML 4.0 Specification*. <http://www.w3.org/TR/1998/REC-html40-19980424>.
- WANG, Y.-M., BECK, D., WANG, J., VERBOWSKI, C., AND DANIELS, B. 2006. Strider Typo-Patrol: Discovery and Analysis of Systematic Typo-Squatting. In *Proc. Usenix SRUTI Workshop*.
- WANG, Y.-M., MA, M., NIU, Y., AND CHEN, H. 2007. Spam Double-Funnel: Connecting Web Spammers with Advertisers. In *In Proc. of International World Wide Web Conference (WWW)*.
- WIKIPEDIA. 2008. Type-in traffic. http://en.wikipedia.org/wiki/Type-in_traffic.
- WITTEN, I. H. AND FRANK, E. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.
- YAHOO. 2007. Yahoo! directory. <http://dir.yahoo.com/>.
- YAHOO. 2008. Yahoo Search Web Services. <http://developer.yahoo.com/search/web/V1/spellingSuggestion.html>.