

# Profiling User Activities with Minimal Traffic Traces

Tiep Mai, Deepak Ajwani<sup>(✉)</sup>, and Alessandra Sala

Bell Laboratories, Dublin, Republic of Ireland  
{tieg.mai,deepak.ajwani,alessandra.sala}@alcatel-lucent.com

**Abstract.** There is a need to strike a balance between the pursuit of personalized services based on a fine-grained behavioral analysis and the user privacy concerns. In this paper, we consider the use of web traces with truncated URLs, where each URL is trimmed to only contain the web domain, to remove sensitive user information. In order to offset the accuracy loss in user activity profiling due to URL truncation, we propose a statistical methodology that leverages specialized features extracted from a burst of consecutive URLs representing a micro user action. These bursts, in turn, are detected by a novel algorithm which is based on our observed characteristics of the inter-arrival time of HTTP records. On a real dataset of mobile web traces, consisting of more than 130 million records and 10,000 users, we show that our methodology achieves around 90% accuracy in segregating URLs representing user activities from non-representative URLs.

## 1 Introduction

Behavioral analysis of mobile users based on their web activities has the potential to transform their online experience. It enables service providers to personalize their deliverable, specialize their content, customize recommendations and target advertisements based on user context. For the network operators, it opens up the possibility of provisioning their resources and dynamically managing their network infrastructure (particularly, with the realization of network function virtualization) to effectively serve the varying user and content demand in order to deliver advanced quality-of-service experience.

However, behavioral analysis also raises serious concerns about user privacy. Users are uncomfortable if personalization is taken too far. In the wider philosophical debate between personalized services based on user behavior analysis and preserving the user privacy, there is a need to find a middle ground that will allow for potential benefits of personalized services and still safeguard the fine-grained sensitive user information.<sup>1</sup>

Ideally, the data set for such analyses should be stripped of all sensitive user information, while still allowing for inference of medium-grained user activity.

---

<sup>1</sup> Specific search queries, personal entertainment preferences, purchased products, location etc. are generally considered highly sensitive user information.

This is becoming even more important with the tightening privacy legislations in various countries [1, 4, 7, 18], increasing regulation (e.g., [8]) and heavy penalties for data breaches which has made network operators as well as service providers (e.g., [26]) more careful about the data sets they collect, store and share. The operators would like to store the minimal amount of data to still be able to perform complex analytics, raising the important question of determining the *thin boundary* between the required data for necessary analytics and the data that can enable mining of highly sensitive fine-grained user traits. In this context, we consider the usage of truncated URLs, wherein each URL is trimmed to only contain the web domain. For instance, the HTTP URL *finance.yahoo.com/q?s=BAC* is truncated to *finance.yahoo.com* (to hide the fact that the user had queried for Bank of America Corp. stock price), the URL *https://www.google.com/#q=postnatal+depression* is truncated to *www.google.com* (to avoid leaking the sensitive health query of the user) and the URL *www.amazon.com/Dell-Inspiron-i15R-15-6-inch-Laptop/dp/B009US2BKA* is truncated to *www.amazon.com* (to avoid leaking the searched or purchased product). Already, many network operators only share the truncated URL data-sets with third-party analysts, owing to privacy considerations. For the non-HTTP traces (e.g., HTTPS encapsulated in IP packets), even the network operators, themselves, have limited information available. While a reverseDNS service can be used to extract the URL from the IP address, it does not recover the content type or the query parameters. Thus, it is important to explore whether high accuracy can still be obtained in profiling user activities if an analyst is restricted to only using truncated-URL web trace. In this paper, we investigate this issue.

Specifically, we focus on the task of identifying URLs that are representative of user activities, which is often an important step in profiling user activities. We note that the remaining task of mapping the representative URLs to activity categories (and creating user profiles) can be done using either manual labeling of interesting categories or in an automated way by using external databases or web analytics services (e.g., Alexa [2]).

The key challenge in filtering out the representative URLs from noisy truncated traffic trace is that a truncated trace lacks many crucial features for such a filtering. These include the file name suffix (e.g., .jpg, .mp3, .mpg etc.) that is usually a good indicator of the content type as well as number, type and values of parameters in the URL strings. Nonetheless, we show that even with the truncated URLs, we can achieve a highly accurate automated classification of web-domains into those that represent the user activity and those that don't. The key insight that we bring in this paper is that a user's traffic trace is composed of many data bursts. A burst usually corresponds to a *micro user action* like a web click, chat reply, etc. and is typically associated with a unique activity. We show that novel features related to burst measurements, such as positioning of a URL in a burst, the number of URLs in the burst containing the web-domain etc., can improve the accuracy of filtering the noise (unintentional traffic such as spam, analytics, advertisements as well as other non-representative traffic such

as images, multimedia, scripts) out of the traffic trace, by around 20%, offsetting the loss due to URL truncation.

To achieve this result, we need to decompose a traffic trace of a user into its constituent data bursts. The problem here is that there is a significant variation in the traffic pattern across different users, at different timestamps and different activities. Even the distribution shape of the inter-arrival time of HTTP records differs significantly from one user to another. We resolve this problem by proposing a novel burst decomposition algorithm that adapts itself to any distribution shape, rather than relying on specific distributions.

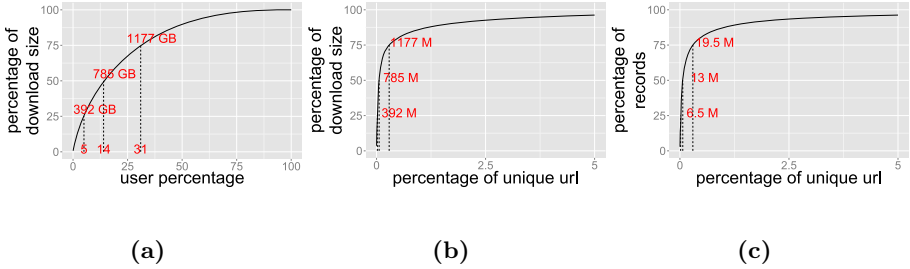
We provide an extensive experimental evaluation over more than 130 million HTTP records generated from 10,000 users over a period of 30 days. The experimental analysis demonstrates that our methodology provides high accuracy (around 90%), in segregating representative URLs from non-representative URLs.

Our approach, thus, enables the network operators to personalize services without risking the leakage of more sensitive user data (as the sensitive information need not be stored or shared). Specifically, it enables many medium-grained personalization applications, including, but not restricted to, product recommendation and targeted advertisement. For instance, knowing when their users read, shop, browse and play games, enables telecom operators to create better pricing schemes that are personalized and targeted for different users and demographics. Such profiling of user activity also opens up many avenues for network optimization to service providers. For instance, system resources can be better allocated to match the data access rate and desired delay time for gaming activities at specific time in the day and better caching strategies can be designed.

**Outline.** In Section 2, we show that there is a considerable variation in the user activity, that necessitates the data-dependent feature extraction and complex statistical models to deal with this problem. Section 3 presents an overview of our methodology. In Section 4, we argue that there is a considerable variation in the distribution shapes of the inter-arrival time of HTTP records and thus, the burst decomposition techniques that rely on specific distribution shapes do not work well across the entire user spectrum. In Section 5, we show how we can remedy the situation by using a threshold on the inter-arrival time of HTTP records, that adapts to the distribution profile of each user. Section 6 presents the results of domain classification using the features extracted from burst measurements. Section 7 presents an overview of some related work.

## 2 Variation in User Activity

The main goal of our investigation is to develop an automated procedure to filter out representative URLs from the noisy trace of truncated HTTP records. In this section, we describe our dataset. We show that in this dataset, there is a significant variation between different users in terms of non-representative traffic, user activities, number of HTTP records etc. In the next section, we propose a



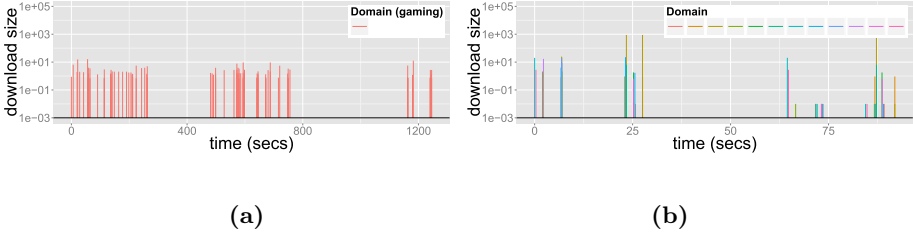
**Fig. 1.** Summary statistics for the traffic trace of a randomly chosen batch of 2000 users

novel methodology that employs robust algorithms for extracting user-dependent features to overcome this high variation in user activity.

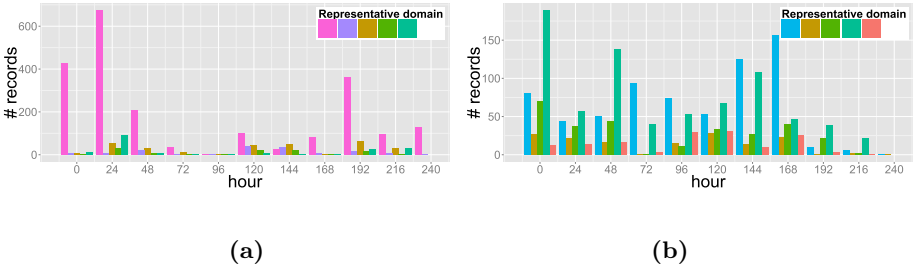
**Dataset.** Our dataset consists of more than 130 million web-logs generated from randomly selected 10,000 users over a period of 30 days from an anonymous network operator. In our traces, each record contains information fields such as user hashed ID, truncated-URL, download size, upload size and timestamp. Note that our dataset is not restricted to any particular domain or limited to a small set of volunteer users. Being a network-side dataset, it is fairly large and diverse in terms of the domains and the users covered. The flip side of this is that it is also very noisy – it contains not just the URLs that a user types in his browser, but also all the redirects, secondary URLs (pictures, embedded videos etc.) and unintentional data (scripts, analytics, advertisement, spam etc.).

**Variation in Total Traffic.** We first observe that there is a significant variation in the HTTP traffic generated by different users. For instance, the number of HTTP records ranges from low tens for some users to tens of thousands for other users, over the 30 day period of study. In fact, a majority of HTTP download traffic (75%) is generated by just 31% of user (Figure 1a). We observe even more skewed distribution for the traffic in terms of the generating activity domain. Less than 0.5% of domains generate 75% of traffic in terms of download size (figure 1b) and HTTP record counts (Figure 1c). Note that even though a large majority (99.5%) of URLs together constitute only a small portion (25%) of the traffic, these less popular URLs are more likely to characterize the unique features of different users and therefore, they play a critical role in differentiating user specific behavior. Thus, it is vitally important to correctly classify these URLs into those that represent the user activities and those that don't.

**Variation in Type of Traffic.** Even among the users with similar total traffic, the kind of web activities and the fraction of non-representative URLs in the traffic trace varies considerably between the users. For instance, Figure 2 shows the web trace snapshot of two users, illustrating two different activity patterns. Different colored segments in this figure represent traffic from different domains, which can be either representative or non-representative. The trace of the first



**Fig. 2.** Snapshot of timestamped HTTP records with download size for two users, showing a significant variation in amount of non-representative traffic



**Fig. 3.** Daily record count of top 5 user activities for two users; different colored bars represent different user activities such as reading, searching, gaming

user (Figure 2a) has only one domain, i.e. gaming, and in fact, repeated records from a single URL for more than 1300 seconds. For this user, there is no non-representative traffic to filter out. However, the web browsing activity of another user shown in Figure 2b alternates between a large number of domains (scripts, multimedia, HTML CSS, advertisements, analytics etc.) in less than 100 seconds, even though he/she is browsing a single web-page during this time. This variation in activity patterns is reflected in download size, inter-arrival time as well as number of HTTP records. In addition, the timestamp patterns of HTTP records also varies significantly from one user to another (see Figure 2).

**Variation in User Behaviors.** We also observe that there is a significant variation between different users in terms of the activities themselves. To summarize the aggregated variation of the top- $k$  domains of both representative and non-representative traffic, we use the following global entropy-based metric to measure this variation:

$$S_a = \frac{-\sum_i \frac{n_i}{nk} \log\left(\frac{n_i}{nk}\right) - \log(k)}{\log(nk) - \log(k)}, \quad (1)$$

where  $n_i$  is the number of times that URL  $i$  appears in the top  $k$  domains, satisfying  $\sum_i n_i = nk$ . By Equation (1), the variation metric  $S_a$  is maximized at 1 when all users have different non-overlapping top  $k$  domain set and is minimized

at 0 when all users have the same non-ordered top  $k$  domain set. For the web trace data,  $S_a$  is 0.484 with 1455 distinct domains from among the top  $k = 5$  domains for the 2000 users. The discovered  $S_a$  value suggests that there is a significant variation in the top activities among the different users. We show this intuition graphically in Figures 3a and 3b, where we depict the activity variation of two users over time. For this figure, we filtered out the non-representative domains manually, selected top 5 representative domains for each user according to the number of HTTP records. Figure 3 presents the daily record counts for each representative domain and demonstrates both the temporal and activity variations in terms of activity types and the magnitude across two randomly selected users.

**Summary.** These above variational statistics imply that the methods to extract features for separating noise from the representative URLs have to adapt to changing user patterns. In particular, the variation in the total traffic and the timestamp patterns necessitates user-adaptive solutions that we explore in the next sections.

### 3 Our Methodology

In this section, we present an overview of our methodology to automatically classify the web-domains into those that represent the user activities and those that don't. The key feature of this methodology is the usage of novel features derived from the burst decomposition of a user's web-trace that improves the accuracy of the classification, offsetting the loss due to URL truncation.

The main intuition behind our methodology is that a user's browsing activity consists of several data bursts. These data bursts correspond to micro user actions, such as a web click or a chat reply. In each burst, there are some URLs representing the user activity intermixed with other unintentional web-traffic such as advertisements, web-analytics etc and secondary URLs corresponding to multimedia associated with the representative URL. Our statistical methodology decomposes the web-trace back into its constituent data bursts. It then leverages specialized features from data bursts (e.g., the position of a URL in a data burst, the number of unique URLs in a data burst, burst duration, burst download size etc.) to segregate the representative web-domains from the remaining web-domains. In Section 6, we show that the usage of features derived from data burst help in significantly improving the accuracy of the segregation task.

A key challenge in our methodology is the decomposition of the web-trace into data bursts. As highlighted already in Section 2, there is a considerable variation in the traffic patterns of different users. We found that even the distribution of inter-arrival time of HTTP records is very different for different users. This makes it particularly difficult to model these data bursts and to find good thresholds to decompose the web-trace into data bursts. We solve this problem by having different thresholds for different users and ensuring that the threshold computing function is robust with respect to the distribution shape. This is

achieved using a novel technique to generate thresholds for each user that adapts to any distribution of inter-arrival time.

## 4 Inter-arrival Time Distribution Models

In this section, we study the inter-arrival time of HTTP records with a view to finding good thresholds that will decompose a user’s traffic-trace into burst of records that represent micro user actions.

As described in Section 3, the key concept behind burst is that when a user performs a micro action like web click, chat reply etc., it not only generates many HTTP records related to the representative activity, but also a large number of secondary records such as advertisements, web analytics, webscripts etc. These records are all intermixed. When the user completes the current micro-action, e.g. reading the current web page, and starts a new one, e.g. opening the next page, a new burst is generated with its associated records. So, the observed inter-arrival time records are the combined results of within-burst and out-of-burst records. However, we expect that the within-burst HTTP records are closer together and the out-of-burst records are far apart in time. By computing an appropriate *separation threshold* on the inter-arrival time, we aim to decompose the traffic into its constituent bursts.

Since traffic patterns and the inter-arrival time distributions for different users are very different, we can’t expect a global threshold to work well for all users. Instead, we compute a different threshold for each user specific to his/her traffic patterns. If the difference between the time-stamp of a record and its predecessor is greater than the computed *separation threshold* for that user, the record marks the beginning of a new burst. Otherwise, the record belongs to the burst of its predecessor.

To learn the *separation threshold* for each user, our first approach is to learn the probability density function of inter-arrival time for the users. By computing the best-fitting parameters for this density function for each user and defining the separation threshold as a function of those parameters, we can decompose the traffic trace for each user into its constituent bursts.

We modeled personalized inter-arrival time distributions by exploring different density functions, such as exponential distribution, pareto distribution and mixtures and concatenations of these distributions (details provided in [14]). From the analysis, we found that even these general density functions are not flexible enough to accommodate highly varied and personalized inter-arrival time of different users. Thus, we concluded that even though this formalism is principled, there is a need for a more robust technique to separate within-burst and out-of-burst records, that is *independent* of the personalized distribution shape of a user.

## 5 Burst Decomposition Using Adaptive Thresholds

In this section, we propose a robust burst decomposition algorithm that is independent of the distribution shape. Our technique only relies on the general

**Algorithm 1.** Burst Decomposition

1. Divide inter-arrival times into bins  $b_i$  of length  $l$  with counts  $c_i$
2. Calculate  $k_i = \frac{c_i}{\sum_{j=1}^i c_j}$
3. Find the smallest index  $i^*$  that  $k_{i^*+j} < p \forall j = 1..J$
4. Define the inter-arrival threshold  $\tau^* = l \times i^*$
5. Group consecutive URLs with inter-arrival times  $\tau < \tau^*$  into bursts.

characteristics of the inter-arrival time distribution (cf. [14]), but not on any specific model. The only characteristic of the inter-arrival time distribution that we use is that there is a within-burst component with high arrival-rate of records (and small inter-arrival time), an out-of-burst component with low arrival-rate (high inter-arrival time) forming a long tail and that these two components are separable with a threshold. Our aim in this section is to have a threshold that adapts itself to any inter-arrival time distribution, subject to this general property.

We first observe that an optimal threshold  $\tau^*$  is expected to lay in a low probability range and should satisfy the following conditions:

- $\forall x < \tau^*$ ,  $p(x)$  should, generally, be high and show the presence of bursts
- $\forall x \geq \tau^*$ ,  $p(x)$  should, typically, have low values and imply user inactivity periods

In order to satisfy the above conditions,  $\tau^*$  has to intercept the minimum  $x$  point where the probability density function of inter-arrival time distribution decays to fairly close to zero and the density of  $p(x)$  values beyond  $\tau^*$  is minimal.

However, to quantitatively measure the significance of each  $p(x)$  value, we need a *scalar indicator* that would determine when a  $p(x)$  value is minimal. This approach would suffer from the selection of a global scalar indicator that would fail in detecting the intrinsic variations of the density proportion between the within bursts and out-of bursts components for different users.

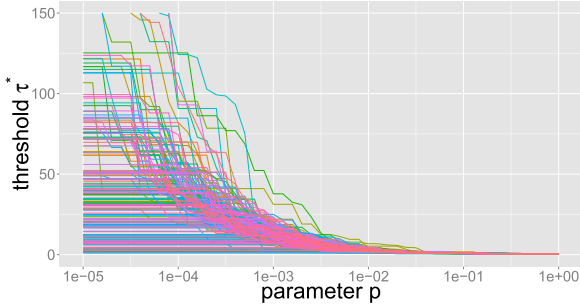
Therefore, instead of using this approach of quantifying  $p(x)$ , we leverage the conditional density, i.e.  $\frac{p(x)dx}{\int_0^x p(y)dy}$ , to determine  $\tau^*$ . Note that,  $\frac{p(x)dx}{\int_0^x p(y)dy} \approx \frac{c_i}{\sum_{j=1}^i c_j}$  which is the probability that a time sample belongs to bin  $i$ , conditioning on the fact that it belongs to a bin less than or equal to  $i$ :  $k_i = \Pr(x \in b_i | x \in b_{1:i})$ . In other words,  $k_i$  measures the contribution of the current bin to the accumulated probability.

Our Algorithm 1 searches for  $\tau^*$  by starting from the smallest value of the inter-arrival time density such that the extended probability by increasing decaying point is insignificant, compared to the accumulated probability at that point (as captured by  $k_i$ ). Specifically, the threshold  $\tau^*$  is found when the contributions of  $J$  consecutive bins are less than a predefined probability, for a pre-specified parameter  $J$ .

The burst decomposition algorithm will group all the records with inter-arrival time less than the obtained  $\tau^*$  into actual bursts.



In the next section, we provide evidence that this algorithm detects meaningful bursts that significantly improve the classification accuracy in identifying the domains that represent user activities.



**Fig. 4.** Calibrating parameter  $p$  by examining the variation of user behaviors through the threshold  $\tau^*$

We estimate the values of the scalar indicator,  $p$ , used in the Algorithm 1 based on an analysis of the corresponding  $\tau^*$  values across all users. In Figure 4 we only report the  $\tau^*$  behaviors of 200 users as representative of entire  $\tau^*$  values computed across all users. It is easy to notice that for  $p = 0.01$  the  $\tau^*$  would range from 1 to 10 seconds, which is a reasonable range to separate inter-arrival time values between within burst and out-of bursts for activities such as web browsing, reading, shopping, etc. Hence, this value of  $p$  was used in our experimental analysis.

Next, we examine the results of our algorithm with respect to users with substantially different behaviors. Even though the distribution shapes and the number of records characterizing these three users are very different, the algorithm successfully finds a user specific  $\tau^*$  as shown in Figure 5.

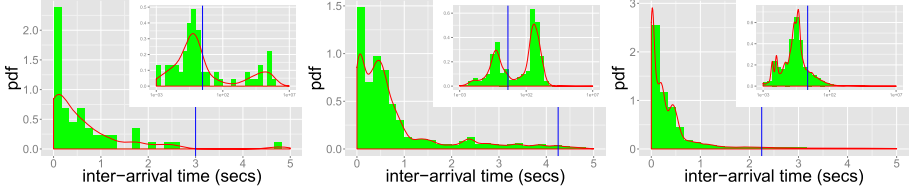
## 6 Domain Classification

In this section, we describe our classification model for identifying the representative URLs and show that it is possible to achieve very high accuracy for this task even with truncated URLs. Features extracted from the burst decomposition presented in Section 5 play a crucial role in significantly improving the accuracy of our classification model.

**Classification Model Formalization.** We use a logistic regression model for the domain classification problem. Our model for logistic regression is as follows:

$$y_i \sim \text{Bernoulli}(q_i), \quad (2)$$

$$\ln\left(\frac{q_i}{1 - q_i}\right) = \mu_i = \beta_0 + \sum_l \beta_l x_{i,l}, \quad (3)$$



(a) Low activity user: 71 records. (b) Medium activity user: 6764 records. (c) High activity user: 35116 records.

**Fig. 5.** User dependent  $\tau^*$  obtained from burst decomposition algorithm

where  $y_i$  is the binary label ( $y_i = 1$  if URL  $i$  is representative and  $y_i = 0$  otherwise) and  $x_{i,l}$  is the specific classification feature that we derive from record-level and burst-level analysis in Section 6.1 and 6.2. The representative probability  $q_i$  is computed by the *logistic function* on a linear predictor  $\mu_i$  and all the parameters are estimated by the Iteratively Re-Weighted Least Squares (IRWLS) method [20].

The domain classification follows three steps. First, we manually label 400 URLs into two classes: representative and non-representative domains. Second, we extract five sets of web traces generated out of  $2K$  random users each, perform the burst decomposition and obtain aggregated measurements independently for each set. Finally, half of the labelled URLs of the first set are used in training the classifier, which is validated by the other half of the first set and the remaining four. We use five different sets to validate the robustness of our approach.

We demonstrate the accuracy of our classification approach in two steps. We first study the accuracy obtained by only using the record-level features and ignoring the burst-level features. Then, we show the improvements we gain by adding the burst-level features which are derived upon the detected bursts from our burst decomposition algorithm.

**Record-level Features.** The key part of our modeling is feature engineering, or identifying the right set of features to achieve a high accuracy. For the record-level features, shown in Table 1, we use the aggregated measurements across all users and compute the quantile values  $k$  by ranging  $k$  from 5 to 95 with an increment step equal to 5. These features were carefully selected to achieve a high accuracy with record-level features. Specifically, for each record we collect the leading and following inter-arrival time and the upload and download size. These features are examined as covariates in our domain classification model.

## 6.1 Accuracy with Record-level Features

**Accuracy.** As shown in Table 2, the resultant accuracy with the record-level features is quite poor. For the five sets of web traces, the accuracy varies between 69.7% and 72.9%, implying that around 30% of the URLs are misclassified.

**Table 1.** Aggregated measurements at record-level where  $\mathcal{R}(i)$  denotes the set of records containing domain URL  $i$ 

Record-level features (wrt $\mathcal{R}(i)$ )	
$t_{r,l,k}$	Quantile $k$ of the leading inter-arrival time
$t_{r,n,k}$	Quantile $k$ of the next inter-arrival time
$s_{r,u,k}$	Quantile $k$ of the upload size
$s_{r,d,k}$	Quantile $k$ of the download size

**Table 2.** AIC and classification accuracy with record-level features

	AIC	BIC	Classification accuracy on 5 sets of 2K users each.				
$\mathcal{C}_{\mathcal{R}}$	242.82	252.72	69.7	72.9	71.0	70.8	71.4

**Table 3.** Estimated values ( $\beta_l$ ), standard deviation ( $\sigma_\beta$ ),  $p$ -values and significance (SIG) for logistic regression model with record-level features

Feature	$\beta_l$	$\sigma_\beta$	$p$ -value	SIG
Intercept	-1.41	0.38	$1.8e-4$	***
$s_{r,d,75\%} - s_{r,d,25\%}$	-0.03	0.01	$4.7e-4$	***
$s_{r,u,50\%}$	0.86	0.26	$1.0e-3$	***

Among the analyzed features we have discovered two particularly important:  $x_{.,1} = (s_{r,d,75\%} - s_{r,d,25\%})$  and  $x_{.,2} = s_{r,u,50\%}$  by the stepwise model selection procedure. The first is the difference between the 75 and 25 quantile statistics of the download size per domain and the second is the 50 quantile statistic of the upload size. The estimated coefficients for this model is shown in Table 3, implying that domains with small variation of download size and high value of upload size have higher chance of being representative domains. However they are the most relevant features at record-level, their discriminatory capacity still remains limited.

## 6.2 Accuracy with Burst-level Features

In this section, we show how the accuracy improves with features measured at burst-level.

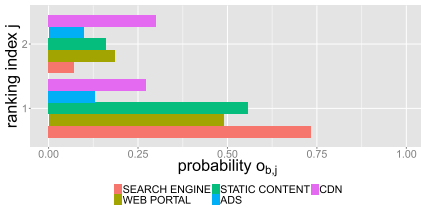
**Burst-level Features.** By leveraging the burst decomposition algorithm, we segment our web traces in a series of consecutive bursts and we measure burst-specific characteristics. Specifically, for each URL  $i$ , we choose a list of aggregated measurements, shown in Table 4, where  $\mathcal{B}(i)$  denotes the set of bursts containing URL  $i$ . We observe that two burst features, i.e.  $o_{b,j}$  and  $u_{b,j}$  ( $j = 1 : 2$ ), in Figure 6 are particularly important in improving the domain classification results. The first measure, i.e.  $o_{b,j}$ , describes the probability that a URL is ranked  $j$  in its burst and the second, i.e.  $u_{b,j}$ , quantifies the probability that there are  $j$  unique domains in the burst containing the URL ( $j = 1 : 2$ ). Similar to record-level features, these aggregated measurements are examined as covariates in our domain classification model.

**Table 4.** Aggregated measurements at burst-level where  $\mathcal{B}(i)$  denotes the set of bursts containing URL  $i$

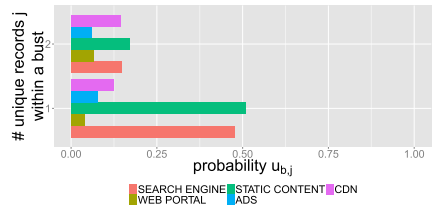
Burst-level features (wrt $\mathcal{B}(i)$ )	
$o_{b,j}$	The probability that URL $i$ is ranked $j$ -th ( $j = 1 : 9$ )
$u_{b,j}$	The probability that a burst containing URL $i$ has $j$ unique URLs ( $j = 1 : 9$ ) in the burst
$d_{b,k}$	Quantile $k$ of burst duration
$t_{b,l,k}$	Quantile $k$ of the leading inter-arrival time of a burst
$t_{b,n,k}$	Quantile $k$ of the next inter-arrival time of a burst
$s_{b,u,k}$	Quantile $k$ of the burst upload size
$s_{b,d,k}$	Quantile $k$ of the burst download size

**Table 5.** Estimated values ( $\beta_l$ ), standard deviation ( $\sigma_\beta$ ),  $p$ -values and significance (SIG) for logistic regression model with all features

Feature	$\beta_l$	$\sigma_\beta$	$p$ -value	SIG
Intercept	-3.30	0.52	$2.2e - 11$	***
$\bar{o}_{b,j=1}$	22.87	3.32	$5.5e - 12$	***
$u_{b,j=2}$	-9.51	3.45	0.01	**



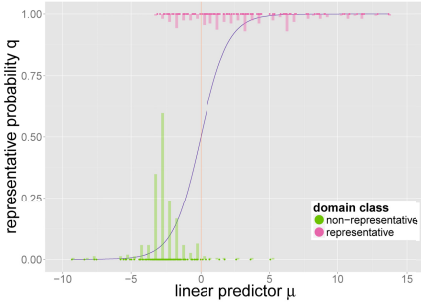
(a)



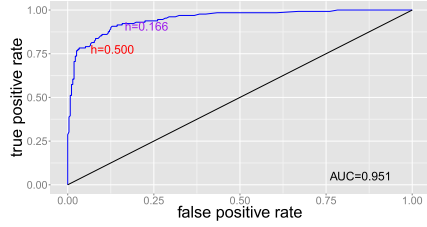
(b)

**Fig. 6.** The probability  $o_{b,j}$  that a URL is ranked  $j$  in its burst and the probability  $u_{b,j}$  that there are  $j$  records in the burst containing the URL ( $j = 1 : 2$ ) for different domains

**Discriminating Features.** We perform a model selection procedure, based on AIC, to select the most discriminating features for our classification model and starting from those listed in Table 4. We observe that the feature  $\bar{o}_{b,j=1} = o_{b,j=1} - u_{b,j=1}$  is selected with high significance. The intuition behind this is that the URLs which usually come first in bursts are more likely belonging to the representative class. Thus,  $o_{b,j=1}$  is a good distinguishing feature between representative domains (*SEARCH ENGINE*, *WEB PORTAL*) and non-representative domains (*ADS*, *CDN*) (as shown in Figure 6). Solely using this feature will misclassify domains from *STATIC CONTENT* class as representative (as these are also likely to come first in burst). This class includes many CSS HTML pages and static images on web-pages. However, the exceptions such as those from *STATIC CONTENT* class have a high probability of being alone in their bursts, as shown by  $u_{b,j=1}$  in Figure 6. Thus, the feature  $\bar{o}_{b,j=1}$  is able to distinguish between most representative and non-representative domains.



**Fig. 7.** Logistic curve of the predictor and the histograms of labelled observations



**Fig. 8.** Receiver operating characteristic curve

Note that the domains in the *SEARCH ENGINE* class have a unique characteristic, i.e. they show high values in both  $u_{b,j=1}$  and  $o_{b,j=1}$  features. However, the differenced  $\bar{o}_{b,j=1}$  can still act as a discriminator in selecting representative domains.

The corresponding estimated coefficients  $\beta_l$  are shown in Table 5 along with standard deviation and  $p$ -values, indicating all significant coefficients. As explained above, domains that have high rankings among others, i.e. do not appear alone in their bursts, are more likely to be representative domains. From the estimated value  $u_{b,j=2}$ , we can also see that domains appearing in small bursts of few unique records have smaller chance of becoming representative domains.

### 6.3 Trade-off Between Classification Metrics

The relation between the linear predictor  $\mu_i$  and representative probability  $q_i$  is plotted in Figure 7, together with the binary labelled observations and histogram of each domain class. The red vertical line represents the decision boundary such that all URL with  $q_i \geq h = 0.5$  are put into representative class and the other are in the non-representative class. Hence, the ratio between the points on the left and right of the red line at row  $q = 0$  corresponds to the ratio between true negative-ness and false positive-ness. Similarly, the ratio between true positive-ness and false negative-ness is at row  $q = 1$ .

In Figure 7, we use the boundary value  $h = 0.5$ , corresponding to the case of minimizing the number of misclassification cases to separate between representative and non-representative records. However, from a decision theory point-of-view, as the false-positive and false-negative penalties are usually different, we may want to optimize the boundary value by customizing these penalties based on application specific requirements. For instance, let's consider an application that generates users's profiles. These kind of applications may want to put a higher penalty for false negatives (representative URLs incorrectly classified as non-representatives) than for false positives (non-representative URLs correctly classified as representative). This is because when determining the activity of a burst, there is an opportunity to prune out the noise (non-representative URLs)

**Table 6.** Trade-off between accuracy, sensitivity, precision and specificity

$h$	0.500	0.166
Precision	0.93	0.77
Negative Predictive Value	0.90	0.95
Sensitivity	0.78	0.91
Specificity	0.97	0.87
Accuracy	0.91	0.88

**Table 7.** AIC and classification accuracy

	AIC	BIC	Classification accuracy					
$\mathcal{C}_B$	112.80	122.69	90.0	88.3	90.2	89.2	90.5	
$\mathcal{C}_R$	242.82	252.72	69.7	72.9	71.0	70.8	71.4	

further, while the representative URLs lost in the process are unlikely to be re-inserted later on. Thus, these applications should be calibrated to improve sensitivity, i.e. the ratio of correctly classified representative URLs to the total number of representative URLs.

Because the penalties are problem-specific and not obvious in many contexts, we show the trade-off between true positive rate (complement of false negative rate) and false positive rate with the receiver operating characteristic curve (ROC) in Figure 8. The figure also illustrates another optimal boundary point in purple for the case of minimizing the sum of false positive and false negative rates. The high value of area under the curve (AUC), [9], again confirms the good performance of our classifier.

Table 6 provides further statistics on the trade-off between false positives and true negatives for two different values of the boundary, that were shown in Figure 8. These trade-offs are characterized in terms of precision, negative predictive value, sensitivity, specificity and accuracy. We note that while the value of  $h = 0.5$  results in higher precision and accuracy, the  $h = 0.166$  results in better sensitivity. Thus, for applications with more emphasis on accuracy, we may choose  $h = 0.5$ , while for applications where sensitivity is crucial, we may select  $h = 0.166$ .

**Accuracy.** The usage of burst-level features, and in particular  $\bar{o}_{b,j=1}$ , results in significant improvement in the accuracy of the classification model. As shown in Table 7, the accuracy improves from around 70% to around 90%; AIC value drops from 242.82 without burst-level features to just 112.8 with these features; and finally, the BIC drops from 252.72 to 122.69.

Altogether, these results show that it is possible to achieve a 90% accuracy in segregating representative URLs from non-representative URLs by using burst-level features on truncated URL web-traces. In other words, the burst decomposition and the extraction of specific features from the bursts overcome the information lost due to URL truncation. Note that this accuracy is in terms of the number of URLs correctly classified as being representative or non-representative. Popular URLs are more likely to be correctly classified by our methodology and thus, the accuracy in terms of number of records, download size (e.g., to answer questions like how much download is generated corresponding to each activity type) or number of bursts (identifying the activity for each burst) is likely to be significantly higher.

## 7 Related Work

The past research related to identifying URLs representing user activities falls into the following categories:

1. Filter out the unintentional traffic by relying on URL suffixes (e.g., .mp3, .js etc.), URL header patterns, HTTP referrers and HTTP blacklists. This category critically relies on full HTTP web-traces. For some papers (e.g., [24]) in this category, the setting even allows to take a peek into a user’s full network traffic (including a deep packet inspection of the content). The full HTTP URLs can reveal highly sensitive user information and their usage raises serious privacy concerns. For instance, Song et al. [16] highlighted a practical privacy attack that exploits seemingly-anonymous recorded information of shortened URL service such as HTTP referrer URLs, countries, browsers, platform, etc to infer the clicking pattern of a specific user. In contrast, our focus is on inferring medium-grained user behavior analysis from minimal traffic traces (truncated-URLs) and on techniques that will allow us to offset the accuracy loss due to URL truncation.
2. Providing an activity description at a very coarse level (e.g., peer-to-peer networking, HTTP browsing, chatting etc.) by filtering out URLs based on connection port number, packet payload, statistical traffic patterns etc. [15,25], primarily for the purpose of network traffic analysis and traffic classification (e.g. for CDN). This is clearly different from our HTTP domain-level segregation.
3. Manual blacklisting of URLs to filter out spam, adult content or advertisements. This has obvious scalability limitations and it is very expensive to manually maintain the blacklists. Furthermore, the existing blacklists (see [23] for a list of many manual blacklists) are for specific purposes (such as spam, adult content, advertisements) and do not contain all non-representative URLs (such as multimedia associated with the main content).

Our work deals with a large, diverse, but noisy traffic trace from a network-side. This allows us to perform a detailed study that is not limited to a few domains or restricted to a few volunteer users. This is in contrast with many publications on behavior analysis that deal with data from users or service-providers.

**Privacy Preserving User Profiling.** There has been considerable work in recent years on privacy preserving personalization. Herein, we list a few approaches:

Bilenko and Richardson [5] also consider the problem of user profiling while mitigating the privacy concerns. However, their approach is based on storing the sensitive information on the client side, in the form of cookies or browser local storage. The storage of this sensitive information still leaves a user vulnerable to privacy violations. On the other hand, our user profiling does not require the sensitive information to be stored at all. Similar client-side approaches in the context of personalized search (e.g., [21]) and online advertisements (e.g., [19]) have also been studied.

Nandi et al. [3] take an alternative approach to privacy preserving personalization. They replace the user traces by traces of group of similar users. However, this requires user segmentation, which in turn, requires significant historical data. Also, this results in an aggregate level personalization and not an individual level personalization.

Also, there are some theoretical solutions based on k-anonymity [17] and l-diversity [13] for preserving privacy. However, it is not clear if they can be useful for profiling personalized time-series data. There are also some approaches (e.g. [12]) that add random or correlated noise to the data to preserve the privacy. However, such approaches also introduce more noise in the user profiles.

**Burst Detection.** Kleinberg [11] proposed a discrete state space model as a burst detection algorithm, with applications in email streams. However, the focus of this solution is the varying exponential distribution's rate, modelled by the hidden state. The rate characterizes the email arrival of a temporal local period but does not provide a clear distinction between within-burst and out-of-burst records. For example, even when the rate goes down to the smallest value, the positive skewness of exponential distribution still favours small inter-arrival time samples. Such an approach is unlikely to work for our problem of segregating two inter-arrival time classes.

Karagiannis et al. [10] showed that the accuracy of exponential distribution varies with different backbone packet traces. In general, exponential distribution has nice mathematical features such as memorylessness and closed-form solutions of sum-concat-minimum operators. However, its light tailed property may not be a good match to some datasets.

## 8 Discussion

We have proposed a novel methodology to identify URLs representing user activities from a truncated URL web-trace. Our statistical methodology offsets the loss in accuracy due to URL truncation by considering additional features derived from the burst measurements. To enable the computation of burst-level features, we propose a novel technique for burst decomposition.

Once the set of representative URLs is identified, one can compare the (live) streaming web-traces of users to infer medium grained activities in real-time and offer personalized services. Burst decomposition can play a critical role in this part as well. Once the user-adaptive thresholds are identified, our burst decomposition algorithm can be used to decompose the streaming trace into bursts and a unique URL representing the activity in that burst can be identified using the identified set of representative URLs.

We consider that our methodology can be very useful for providing personalized services, while being considerate about more sensitive user privacy data. For instance, state-of-the-art techniques to predict click-through-rate (CTR) rely on behavioral targeting of fine-grained user data [6], such as advertisement clicks, web-page clicks, page views and search query data. A medium-grained user profiling, such as the one created by our technique, can be used to provide good



CTR predictions while preserving privacy considerations. Similarly, user segmentation based on behavioral targeted advertisement (e.g., [22]) can also benefit from our medium-grained profiling. Our profiling can also be employed to re-rank the search results for a more personalized experience (similar to the approach in [21]). More generally, we hope that our work will lead to deeper studies on the usage of truncated URL traces, as a means to striking the fine balance between personalized services and user privacy.

## References

1. European Communities (Electronic Communications Networks and Services) (Privacy and Electronic Communications) Regulations (2011). [http://dataprotection.ie/documents/guidance/Electronic\\_Communications\\_Guidance.pdf](http://dataprotection.ie/documents/guidance/Electronic_Communications_Guidance.pdf)
2. Alexa: Actionable Analytics for the Web (2015). <http://www.alexa.com>
3. Nandi, A., Aghasaryan, A., Bouzid, M.: P3: a privacy preserving personalization middleware for recommendation-based services. In: Proceedings of 4th Hot Topics in Privacy Enhancing Technologies Symposium (HotPETS 2011) (2011)
4. BBC (2014). <http://www.bbc.com/news/technology-25825690> (accessed November 2014)
5. Bilenko, M., Richardson, M.: Predictive client-side profiles for personalized advertising. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 413–421. ACM (2011)
6. Chen, Y., Pavlov, D., Canny, J.F.: Large-scale behavioral targeting. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 209–218. ACM (2009)
7. European Data Protection Supervisor (2011). [https://secure.edps.europa.eu/EDPSWEB/webdav/site/mySite/shared/Documents/Consultation/Opinions/2011/11-05-30\\_Evaluation\\_Report\\_DRD\\_EN.pdf](https://secure.edps.europa.eu/EDPSWEB/webdav/site/mySite/shared/Documents/Consultation/Opinions/2011/11-05-30_Evaluation_Report_DRD_EN.pdf) (accessed November 2014)
8. Facebook: Facebook and the Irish Data Protection Commission (2011). <https://www.facebook.com/notes/facebook-public-policy-europe/facebook-and-the-irish-data-protection-commission/288934714486394> (accessed November 2014)
9. Fawcett, T.: An introduction to ROC analysis. *Pattern Recognition Letters* **27**(8), 861–874 (2006)
10. Karagiannis, T., Molle, M., Faloutsos, M., Broido, A.: A nonstationary poisson view of internet traffic. In: INFOCOM (2004)
11. Kleinberg, J.M.: Bursty and hierarchical structure in streams. In: KDD, pp. 91–101 (2002)
12. Li, F., Sun, J., Papadimitriou, S., Mihaila, G.A., Stanoi, I.: Hiding in the crowd: privacy preservation on evolving streams through correlation tracking. In: Proceedings of the 23rd International Conference on Data Engineering, ICDE, pp. 686–695. IEEE (2007)
13. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkatasubramanian, M.: L-diversity: privacy beyond k-anonymity. *TKDD* **1**(1) (2007)
14. Mai, T., Ajwani, D., Sala, A.: Profiling user activities with minimal traffic traces (2015). ArXiv e-prints
15. Nguyen, T.T.T., Armitage, G.J.: A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys and Tutorials* **10**(1–4), 56–76 (2008)

16. Song, J., Lee, S., Kim, J.: I know the shortened urls you clicked on twitter: inference attack using public click analytics and twitter metadata. In: Proceedings of the 22Nd International Conference on World Wide Web, pp. 1191–1200. WWW 2013, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2013). <http://dl.acm.org/citation.cfm?id=2488388.2488492>
17. Sweeney, L.: k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **10**(5), 557–570 (2002)
18. TechCrunch (2015). <http://techcrunch.com/2014/10/01/hamburg-google/> (accessed March 2015)
19. Toubiana, V., Narayanan, A., Boneh, D., Nissenbaum, H., Barocas, S.: Adnostic: privacy preserving targeted advertising. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2010. The Internet Society (2010)
20. Wood, S.N.: Generalized additive models: an introduction with R. Chapman and Hall/CRC Texts in Statistical Science Series. Chapman and Hall/CRC Press (2006)
21. Xu, Y., Wang, K., Zhang, B., Chen, Z.: Privacy-enhancing personalized web search. In: Proceedings of the 16th International Conference on World Wide Web, WWW, pp. 591–600. ACM (2007)
22. Yan, J., Liu, N., Wang, G., Zhang, W., Jiang, Y., Chen, Z.: How much can behavioral targeting help online advertising? In: Proceedings of the 18th International Conference on World Wide Web, WWW, pp. 261–270. ACM (2009)
23. Zeltser, L. (2014). <http://zeltser.com/combating-malicious-software/malicious-ip-blocklists.html>
24. Zhang, F., He, W., Liu, X., Bridges, P.G.: Inferring users' online activities through traffic analysis. In: Proceedings of the Fourth ACM Conference on Wireless Network Security, pp. 59–70. WiSec 2011. ACM (2011)
25. Zhang, J., Xiang, Y., Wang, Y., Zhou, W., Xiang, Y., Guan, Y.: Network traffic classification using correlation information. *IEEE Trans. Parallel Distrib. Syst.* **24**(1), 104–117 (2013)
26. Zuckerberg, M.: Our commitment to the facebook community (2011). <https://www.facebook.com/notes/facebook/our-commitment-to-the-facebook-community/10150378701937131> (accessed November 2014)