This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



# Server-side Traffic Analysis Reveals Mobile Location Information over the Internet

Keen Sung, Joydeep Biswas, Erik Learned-Miller, Brian N. Levine, Marc Liberatore

**Abstract**—Users can attempt to thwart third-party services from discovering their location by disabling location services on their mobile device. In this paper, we show that web services can use throughput information to reveal the path taken by the phone and its owner among a set of possibilities. For example, a TCP-based music streaming service can compile a sequence of throughputs over several minutes. We collected hundreds of traces of music that we streamed to phones in two different scenarios: a user traveling to four different towns from campus (or the reverse direction); and a user traveling within our campus. We evaluate three classifiers: $k$-Nearest Neighbors ($k$-NN), which compares a test sequence with respective time points of training sequences; a Hidden Markov Model (HMM), which computes the transition and emission probabilities of different geographic areas and chooses the most likely sequence of a test trace; and a Naive Bayes Classifier with KDE-based throughput estimates (NB-KDE), which looks at the density of throughputs at each time point along a path. In our study, the $k$-NN, HMM and NB-KDE approaches can distinguish between a small number of geographic routes taken by mobile users using only throughput measurements. The NB-KDE method performed best, using throughput alone to identify the path and direction among two roads within a University campus (4 classes) with 77% accuracy, and the path and direction among 4 roads (8 classes) out of town with 83% accuracy. Furthermore, it was able to classify among 8 paths with greater than 59% accuracy after one minute. We examine the limitations of these techniques.

## 1 Introduction

While smartphones allow users to disable location services, a device's whereabouts can often be remotely deduced. A remote party communicating with a phone has a window into the complex interactions between phone and cell tower. These features can be used to reveal which path was taken among a set of possibilities by the phone and its owner. This information is leaked regardless of application-level privacy settings. Cell phone users have such experiences intuitively in many common situations. For example, a caller may be able to tell when a friend has entered an elevator based solely on call quality; or a user may notice a loss in data throughput during a subway ride.

Consider, more generally, a TCP-based streaming audio service (like that provided by Spotify or Pandora) that wishes to localize users who have disabled GPS. While location data is unavailable, the service may choose to monitor the changing throughput of a stream, which is affected by a user's geography, and use those variations in throughput as evidence of a user's location. To acquire training data, the service may leverage their observations of unwitting users who do not disable GPS while streaming music from an owned server.

• *E-mail: {ksung,joydeepb,elm,levine,liberato}@cs.umass.edu*

We show that there is a significant correlation between cellular throughput and geography through a large measurement study of mobility and cellular downloads over a large area. And we show that a classifier, from only throughput measurements, can remotely identify the geographic path of a user from across the Internet. We examine the problem from the perspective of an attacker who has an existing model of network traffic over a geographic region, and a list of possible paths the target will take. We consider this a *time-sequence multi-class classification problem* [21], using remotely recorded network traffic traces as instances to be classified, and geographic paths as classes. Using throughput measurements of music streamed via TCP to a mobile phone on a 3G/UMTS network, we develop several classifiers, and optimize their parameters using a development dataset. We evaluate results using a separate test set.

Throughput traces alone cannot pinpoint a location from among the entire Earth, but they can be combined with other partial information. For example, users that turn off GPS can still be geolocated from a non-cellular IP address. A user may leave home or work where their cell phone had a static, geo-locatable IP address. While mobile, the IP address provided by a cellular ISP cannot be geo-located. However, with the techniques we introduce in this paper, their path from (or to) a geo-located IP can be determined remotely via only a sequence of throughput.

**Contributions.** We collected hundreds of traces of music that we streamed to phones along two roads on campus, and four roads going out of town, in two directions each. Within small geographical cells, mean throughput is largely consistent and distinct. We evaluate the performance of three remote localization classifiers that leverage this consistency. We explore their limitations in terms of the amount of training data needed, and the length of test sequence.

Our naive approach, trained on the mean throughput of each path, performed better than random. We compared this method against three classifiers, described in section 4. Our best performing approach, the NB-KDE classifier, can correctly determine the path taken by the phone from one of four longer paths to neighboring suburbs with greater than 90% accuracy, and the path and direction (8 choices) with 76% accuracy. In a separate experiment involving data collected only from within a 4km$^2$ area, in and around our campus, the NB-KDE approach could identify the direction and part of campus the user was traveling with 76% accuracy.

Our contributions are summarized as follows:

- We define the Internet-based remote localization problem and demonstrate for the first time that cellular phones can be remotely localized based on the quality of an Internet connection.
- We describe our data collection methodology, and discuss some properties of our dataset. Our analysis shows statistically different throughput means among small geographic areas (0.9 km$^2$ each). Phones that move between locations travel through consistent and distinct network conditions that are remotely observable.
- We detail three classifiers: a $k$-nearest neighbors ($k$-NN) classifier, which trains on the ordered sequence of throughput values of each route, a hidden Markov model (HMM) classifier, which exploits the consistency in throughput values at each location, and a naive Bayes (NB-KDE) classifier that uses kernel density estimation of throughput at each second along a path.
- We examine the performance of the three classifiers, demonstrating that the $k$-NN, HMM and NB-KDE approaches can distinguish between a small number of geographic routes taken by mobile users using only throughput measurements. We examine two different scenarios: a user traveling to a different town from campus (or the reverse direction); a user traveling within our campus.
- We examine the limitations of these techniques. We determine the limits on the amount of data and the length of the trace required to achieve a certain accuracy. These limits provide a starting point for the development of a defense.

This paper is an extension of our previous work [17] in the following ways:

- We collected 112 new traces, in addition to the original 295. The new traces were collected from phones traveling along either direction of the campus bus loop, whose path encompasses a much smaller area compared to the paths of the original four paths out of town. As with the original traces, these were recorded by streaming music via TCP over a 3G/UMTS cellular connection.
- We designed a new hidden Markov model (HMM) that used geographic areas as hidden states, instead of throughput levels. This is intended to allow both path and direction to be classified by the HMM, which was not possible for the HMM in our previous work.
- We designed a new classifier, NB-KDE, that considers the Gaussian kernel density of throughput at each time point along a path. In fact, this new approach is the best performing.
- We analyze the limitations of the algorithms. In particular, we quantify the effect of trace length on accuracy. We find that even short traces about one minute long are still classifiable.

## 2 Problem Statement and Attacker Model

We are interested in a subset of the Internet-based remote localization problem: *Can an attacker, providing an Internet-based service to a mobile user that has disabled geolocation features, infer the path taken by the mobile user from among a limited set of paths, using only information visible at the server?* In later sections of this paper, we show that the answer is yes and with high accuracy. Here, we elaborate on the problem, our assumptions, and our approach and its limitations.

**Motivation.** Solving this subproblem of the remote localization problem is an important step toward a solution to the more general problem. Aside from the research challenge, this problem is of interest to the general public. Particular users care about their own location being determined and shared without their consent. Further, society may judge phone-based location tracking of individuals as something to be regulated or otherwise controlled. For example, a report by the U.S. Government Accountability Office notes that federal action could help protect consumer privacy [7].

**Assumptions.** We assume the attacker is a proprietor of a web service that sends data to a user in motion, and attempts to determine the user's geographic location without her permission. We assume the attacker is the remote end-point of the target's communication, as is the case for TCP-based streaming services such as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



Spotify, Pandora, and many others; or, the attacker may have access (perhaps unauthorized) to network-level traces at this location. We assume that the carrier, who can localize the mobile node by examining the cell towers to which it has associated, is not assisting the attacker. We assume that the attacker does not have direct access to the internals of the cellular infrastructure or to the mobile device used by the target, and therefore will find it nearly impossible [1] to geolocate the user from its carrier-assigned IP address. (Carriers use a small pool of addresses that are re-used across the country from one minute to the next.) We assume the attacker does not have the ability to direct the mobile device to reveal its location overtly. The attacker only passively analyzes the communication between the mobile device and its server. Our attacker uses only throughput measurements of the target's data stream and not the content, which could be encrypted or otherwise unavailable — though we do assume the attacker could link flows if the remote end-point IP address changes. This assumption is reasonable given our attacker model.

Finally, we limit our scenarios to those where at either or both ends of a path, the user obtains an IP address that is geolocatable. For example, at home a user is likely to be connected to the Internet via an IP addressed assigned by their landline/non-cellular ISP. The same is true at a workplace or other destinations, such as some stores. In essence, our assumption is that paths are defined as starting or ending by geolocatable IP addresses and use non-geolocatable IP addresses assigned by a cellular ISP while mobile. (We enumerate limitations of our evaluation in Section 5.)

These assumptions are widely applicable. TCP-based audio and video (i.e., music and movies) streaming apps are common and popular. And any installed app with network access can allow itself to accept data transfers from a server without notifying the user and generally without detection. Finally, there are no protections against a website sending streaming data to a mobile web browser in the background unbeknownst to the user while a page is open (e.g., using Ajax). For most web sites, user sessions are only a few minutes long. However, in our tests, we found that a web page can direct the browser to continue to download data in the background after the phone's screen is turned off. Such transfers are allowed for 5 minutes by Chrome on Android, though not by other browsers.

**Approach.** Our approach builds models of the effects of mobility upon network traffic, and uses these models to determine the mobility of users. Specifically, the attacker compares a trace of network traffic generated by a mobile user against a set of models representing specific paths through a targeted geography. The attacker creates these models by gathering information about TCP's performance on a set of possible routes that he assumes the target may take. The attacker may gather this information, which consists of traces of network traffic, during any period when the traffic observed would be similar — it need not be done strictly prior to the attack. This training information may be gathered from other users that have not disabled GPS. We conjecture that greater temporal locality will improve the attacker's performance, though we did not explicitly test this assumption.

**Limitations.** The traces we consider are on the order of tens of minutes long, and we also evaluate how trace length affects accuracy. The user must be traveling at a similar velocity at each location with limited deviations from a typical training trace. All our measurements were taken by users riding public buses, except one path where the measurements were taken from a car. In all cases, the vehicle went at the pace of the road and congested traffic, if any. Finally, our main result relates to selecting a path from limited possibilities rather than the full set of roads on Earth.

## 3 Data Exploration

We collected 407 measurements of mobile phones traveling around campus and to nearby towns, illustrated in Figure 1. Two hundred eighty-six of those traces — the *metro* dataset — were on paths to and from a central location to four remote locations, each about 25 minutes away (roughly a 360 $km^2$ area). One hundred twenty-one of those traces — the *campus* dataset — were on paths encircling a central location around a 4 $km^2$ area. We also collected 29 stationary traces to serve as a simple baseline to contrast the effect of motion on throughput changes.

Signal strength and throughput characteristics are tied to geography. To explore this relationship, we discretize geographic data into square areas of varying granularity (see Figures 4 and 5). In our first data set, the mean throughputs of 95% of 1 $km^2$ areas are statistically different from at least 90% of the other areas. In our second dataset, which cover a smaller area, the mean throughputs of 85% of 0.01 $km^2$ areas are different from at least 85% of other areas. The implication is that the route traveled by a mobile node is through a largely unique sequence of mean throughputs that is classifiable. Figure 2 illustrates these differences in throughputs.

### 3.1 Data Collection Methodology

We recorded traces[1] of GPS location and signal strength using four Android cell phones. A server in our building streamed music continuously to the phones during measurement trials, while TCP traces were logged at

1. Traces from our experiments are available for download from http://traces.cs.umass.edu.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



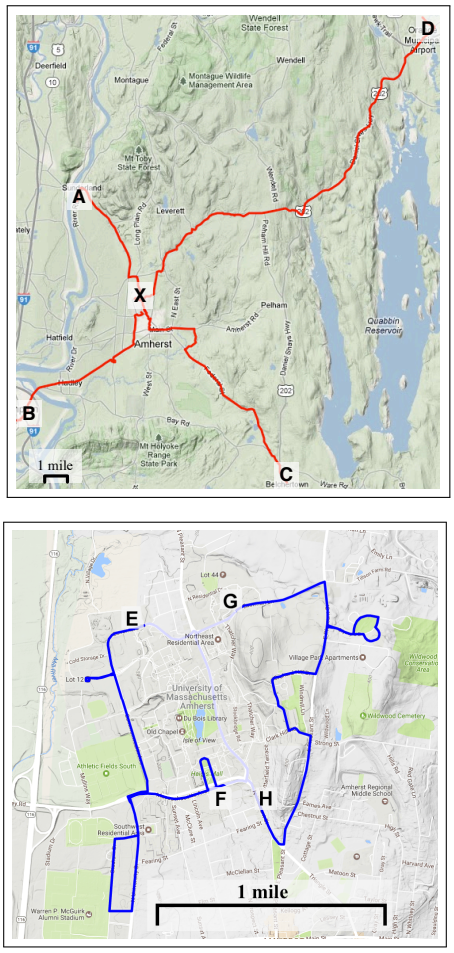


Fig. 1. Top: For the 'metro' dataset, we gathered data on four popular paths in our area, in two directions for each path. All paths intersect in Amherst, MA, labeled as "X". Bottom: For the 'campus' dataset, data was collected along the campus figure-eight bus loop around the University of Massachusetts campus, which runs in both directions. We compared both directions of paths along the West ("E"–"F") and East ("G"–"H").

the server. We later combined sets of corresponding phone and server traces, synchronized using the timestamps within the traces. Note that it is impossible without carrier participation to take measurements within the network. Moreover, our goal is to assess a weaker attacker who is without special access to cellular infrastructure, but who can take measurements at an end host. We used *Samsung Nexus S*, *Samsung Galaxy S*, *Motorola Atrix*, and *HTC Inspire* phones, all connected to the AT&T UMTS (3G) network, to record traces. The 802.11 radio on the phone remained powered off during the experiments. We collected the data under varying traffic and weather conditions.

We took three sets of traces:

- **Mobile 3G Measurement Set — 'Metro'**: We collected data during a one-month period. Each measurement was taken as a phone traveled along one of four routes going either toward or away from our central location (point X in Figure 1). The individual paths are shown on a map in Figure 1 and summary statistics appear in Figure 2. In total, we recorded 286 traces in this set.
- **Mobile 3G Measurement Set — 'Campus'**: We recorded 141 traces from the same phones, along one of two directions around a bus loop on campus. Traces were collected over a period of eight months.
- **Stationary 3G Measurement Set — Baseline**: We recorded 29 traces from stationary phones, connected to the UMTS (3G) network, located in different locations near our central location.

The phones collected traces of GPS location (with 10m accuracy) and signal strength.[2]

Each element of the traces was sampled once per second. Traces of network activity on the server consist of standard `pcap` logs. We did not limit traces to periods of cellular connectivity, and some traces consisted of several TCP connections.

In the mobile sets, we hired several persons to collect data on these specific paths, and no person was assigned to a single path or phone. Our goal was to avoid learning the phone model or user behind the movement. Each path differed in distance. In the first dataset, each took about 25 minutes on average to travel by car or bus. The travel time to location A was the shortest and D the longest. In the second dataset, two bus routes ran bidirectionally in a loop; each cycle took about 45 minutes. We later divided the bus loop data into East and West paths (see Figure 1). We discuss the implications of path duration on classifier bias in Section 5.

Because we relied on a consumer phone platform, on some occasions the experiment failed because either the end time or start time were not recorded correctly, due to a GPS failure or write-to-flash failure. We did not attempt to even out the number of traces per path after our collection period completed. Though the number of traces per route and direction varies, we did not alter which traces to collect. In our experiments, we discarded traces that did not exceed 10 kilobytes transferred (indicating a network error).

### 3.2 Geographic Analysis

We grouped all server-side throughput and client-side signal strength measurements into small geographic areas (much smaller than and having no correspondence to carrier cells) to determine if each area had consistent and differentiable mean throughput. The efficacy of any throughput-based remote localization scheme depends

2. As reported by android.telephony.SignalStrength.getGsmSignalStrength().

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



| Route | Distance (mi) | Num. Traces Collected | Throughput (KB/s) mean ± s.d. | Duration (min) mean ± s.d. |
|---|---|---|---|---|
| A-to-X | 7.0 | 63 | 157.4 ± 115.2 | 16.1 ± 4.2 |
| B-to-X | 21.2 | 24 | 63.4 ± 94.9 | 30.5 ± 4.8 |
| C-to-X | 10.5 | 29 | 116.5 ± 111.4 | 34.8 ± 7.4 |
| D-to-X | 8.5 | 19 | 34.6 ± 64.4 | 36.1 ± 5.1 |
| X-to-A | 7.0 | 68 | 134.8 ± 110.0 | 16.4 ± 4.3 |
| X-to-B | 21.2 | 28 | 47.1 ± 81.3 | 33.6 ± 9.0 |
| X-to-C | 10.5 | 31 | 120.1 ± 109.4 | 32.1 ± 7.8 |
| X-to-D | 8.5 | 24 | 45.0 ± 76.1 | 34.9 ± 10.3 |
| E-to-F | 2.6 | 57 | 142.9 ± 117.4 | 9.0 ± 9.3 |
| F-to-E | 2.6 | 76 | 126.2 ± 122.4 | 7.8 ± 11.2 |
| G-to-H | 2.8 | 53 | 206.7 ± 154.0 | 11.7 ± 4.6 |
| H-to-G | 2.8 | 56 | 194.5 ± 148.1 | 12.8 ± 4.5 |
| Stationary | 0 | 29 | 206.6 ± 122.5 | 20.4 ± 5.5 |

Fig. 2. Details of the traces in our Measurement Sets. Letters refer to landmarks labeled in Figure 1. In total, we recorded 407 mobile and 29 stationary traces.

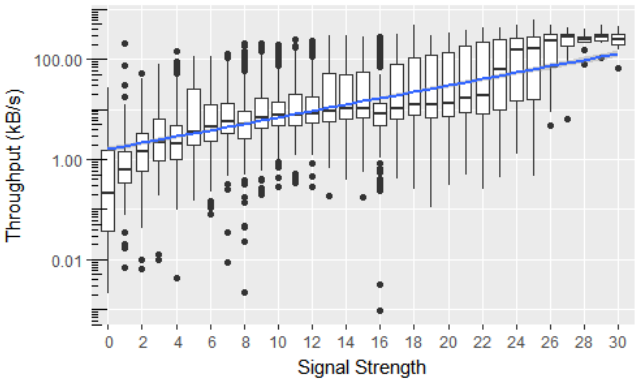


Fig. 3. On a per-second basis, the correlation between server-side throughput and client-side signal strength is 0.24. The plot shows a linear fit. Note: Android's reported signal strength correlates linearly to decibels, which is a logarithmic measure.

on such consistency. We found geographic consistency in both cases and a weak correlation between the two features.

Server-side throughput is influenced by the wireless link between the phone and cell tower [4], the network conditions and infrastructure [23] between the phone and server, the TCP algorithm [13], and other factors. Signal strength is just one factor that influences the wireless link but it is the factor with the strongest tie to geography. Received signal strength is influenced by many physical features, including occlusions between the radio and cell tower from tree foliage, the body of the person carrying the phone, buildings, and other structures. Most of these physical features do not change from one day to the next, and therefore can have a permanent effect on throughput in a particular area.

We found a weak correlation between client-side signal strength and server-side throughput of 0.24 when considering mean throughput and mean signal strength on a per-trace basis. Figure 3 shows the distribution of throughput values per signal strength value as a boxplot. The figure also plots the least-squares linear fit of the two variables as a visual guide. The figure is based on the range of signal strength values measured by the phones. These values can be converted to integers, as defined in GSM standard TS 27.007, with 0 referring to $-113$ dBm or less, 31 referring to $-51$ dBm or greater. Each value between 1 and 30 is a linear increase from $-111$ to $-53$ dBm. We discarded values of 31, as the unbounded range it captures is too large for a meaningful regression.

Figure 4 shows the mean throughput (left) and signal strength (right) of geographic areas in our measurements of paths to towns ('metro' dataset); Figure 5 shows the same information in left and right plots for the smaller measurement area within our campus. The error bars of each mean indicate the 95% confidence interval of the mean. Each plot is sorted by an increasing mean value, and therefore the order of areas in the plots is not the same. Using a two-sided, 95% confidence interval $t$-test, we performed a pairwise comparison of the mean throughput of the areas. In the 'metro' dataset, each cell is on average significantly different from 85.4% of other cells. In the 'campus' dataset, the discretized areas are smaller (i.e. finer granularity) and there is more similarity, with each cell differing from 73.6% of others. The consistency of these values and the differences among areas suggests that latent information linking throughput and geography is available for training a classifier.

The takeaway of the plots is that mobile nodes will travel through a sequence of areas that has a relatively unique signature of mean throughputs. The task of classification is to match the observed throughput to a training set that captures these means. In the simplest approach, we can classify based on the mean throughput that a mobile device obtains from visiting a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



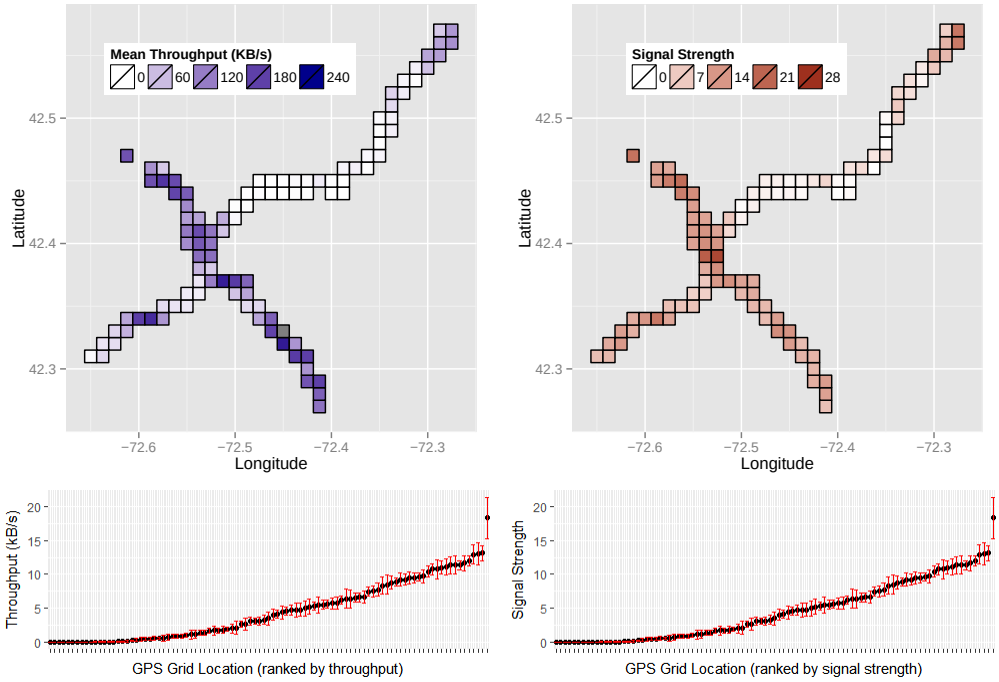


Fig. 4. The mean throughput (left) and signal strength (right) of geographic areas in our measurements of paths to surrounding towns. On average, throughput in each cell is significantly different from throughput in 85.4% of other cells. This data suggests that latent information linking throughput and geography is available for training a classifier.

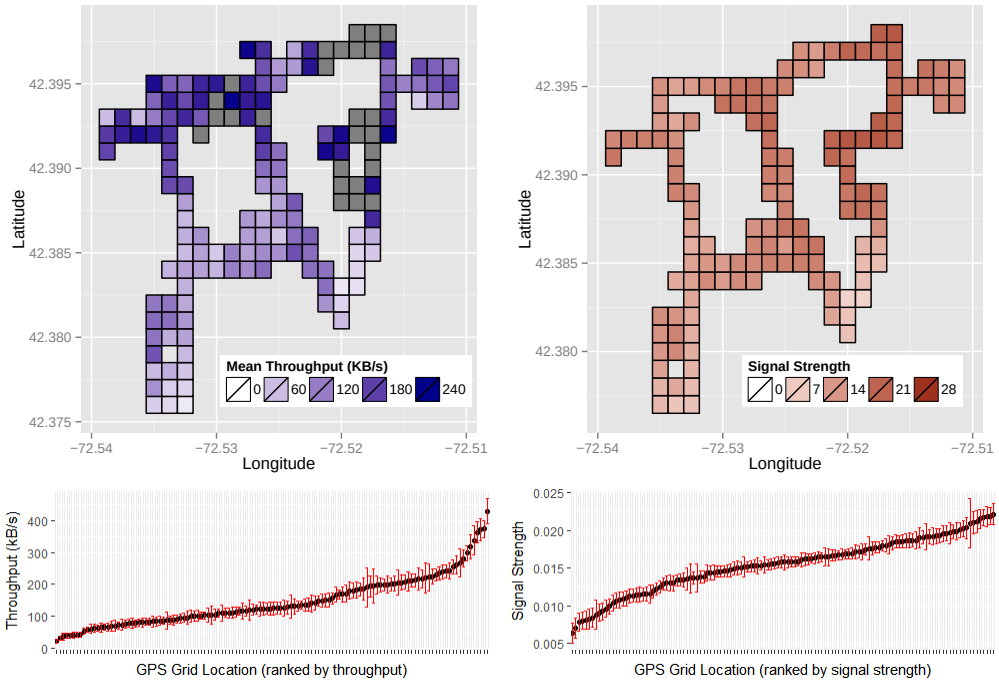


Fig. 5. The mean throughput (left) and signal strength (right) of geographic areas in our measurements of our 'campus' data (a bus traveling within a 4 $km^2$ area). Throughput in each cell is significantly different from throughput in 73.6% of other cells.

series of areas. In three more-advanced approaches, we can classify based on the per-second mean throughputs of each trace. We evaluate all three approaches, detailed further in the next section.

## 4 Classifiers for Mobile Throughput Traces

In this section, we describe several *classification* algorithms that could be used to identify which path a mobile phone user is traveling down.

Classifiers build models of labeled training instances of data, and use these models to decide which class an unlabeled test instance belongs to. The instances we considered were created from packet capture (pcap) files, and the specific data we were interested in was TCP throughput. We discretized this data into one-second intervals, and treated each instance as a sequence of per-chunk mean throughputs. With one second chunks, the index of each throughput value is the time since the start of the trace. Corresponding GPS coordinates are recorded on a per-second basis as ground truth.

We present a hidden Markov model (HMM) [5], $k$-nearest neighbors ($k$-NN), and naive Bayes (NB-KDE) classifier with respect to a motion-throughput model.

Each classifier is derived from this model, with different assumptions. The HMM classifier is most general, and attempts to maximize the estimated hidden states and transitions. The $k$-NN and NB-KDE classifiers are more rigid, and make the assumption that users move through a certain path with consistent velocity. The $k$-NN classifier finds $k$ of the closest traces by comparing the throughput at each time point of the trace. The NB-KDE classifier determines the distribution of throughputs at each time point for each path, and finds the most likely trace.

In our evaluation, we tested the identification of a path rather than a sequence of locations. Therefore, we have split the location sequences into separate path classes.

### 4.1 Model

We have a set of discrete locations, $L$, and a sequence of throughputs $\mathbf{b}$. In general, we want to associate $\mathbf{b}$ with a sequence of locations $l_0, l_1 \ldots$, where $l_i \in L$.

In fact, we are trying to match $\mathbf{b}$ with the most likely path (or class) $\mathbf{c}$ in a set of paths $C$. A path $\mathbf{c}$ represents the sequence of locations in a path: $\mathbf{c} = l_0^{\mathbf{c}}, l_1^{\mathbf{c}}, \ldots$.

In order to match a sequence of throughputs $\mathbf{b} = b_0, b_1, \ldots$, we store all observed throughput at each location in $L$, and use this information to compute the likelihood that a location exhibits a certain throughput. We denote the observations at a location $l$ as $\mathbf{o}_l = o_1^l, o_2^l, \ldots$.

Each classifier extends this model. The HMM discretizes $L$ as square cells on a geographic map. However, the $k$-NN and NB-KDE classifiers assume the location sequences are traversed at a consistent velocity between traces, so each point in $\mathbf{c}$ represents a single second along the path. In other words, each class $\mathbf{c}$ is expected to be traversed in $|\mathbf{c}|$ seconds, and has a sequence $l_{0 \ldots |\mathbf{c}|-1}$.

### 4.2 HMM

The HMM classifier can account for users traveling at variable speed down a path. However, this is a disadvantage if speed is consistent between traces, since it disregards timing information. We predicted that an HMM, which does a basic alignment of the data, would allow us to infer location with finer granularity.

We represent each class as a separate hidden Markov model. Each approximate square area (as shown in Figure 4) is represented by a state $l \in L$. Each state has a set of emission probabilities — the probability that a certain discrete level of throughput is seen at that state. We smoothed the observed throughput into sequences of $w$ second windows. The value of $w$ varies between 3-fold cross validation tests. Based on the resulting sequence, we compute the likelihood that a certain sequence of states was traversed. We choose the class that has the highest likelihood as our guess.

The states are fixed as approximate square areas on the map. Emission probabilities at each state are determined by counting the number of occurrences of each throughput level in the training data in each area and normalizing; see Figure 6. Sixteen throughput levels were represented by $e \in 0 \ldots 15$. The probability of a certain throughput level occurring at a certain location is from a categorical distribution.

In particular location, the probability that an emission level $e$ occurs is

$$E[p(e|l)] = \frac{Count(e)}{\sum_{e'} Count(e)}. \tag{1}$$

Following the Markov assumption,

$$p(\mathbf{l}|\mathbf{b}) = p(l_0|b_0)p(l_1|b_1, l_0), p(l_2|b_2, l_1, l_0) \ldots \tag{2}$$
$$= p(l_t|b_t, l_{t-1}) \tag{3}$$
$$= \prod_t p(l_t|b_t, l_{t-1}). \tag{4}$$

In each sequence, we count the number of transitions between each state, and normalize. The transition from $l_x$ to $l_y$ is denoted by $l_{x \to y}$, and its probability by $p(l_y|l_x)$. In other words, if we are in $l_x$ at time $t$, we determine the probability we are in $l_y$ at time $t+1$:

$$p(l_y|l_x) = \frac{Count(l_{x \to y})}{Count(l_x)}. \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



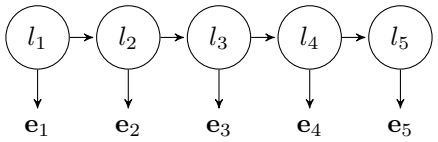


Fig. 6. Locations are represented by hidden states in a hidden Markov model, $l_i$. A user moves from one state to the next with a fixed transition probability of 0.7 multiplied by the probability an observed level of throughput is seen in that state according to the emission probability vector $\mathbf{e}_i$.

Given both the emission probabilities and transition probabilities of each state, we can now compute the likelihood of the model given a test sequence (also discretized into chunks),

$$p(\mathbf{l}|\mathbf{b}) = \prod_{t=1}^{|\mathbf{b}|} p(l_t|l_{t-1})p(b_t|l_t). \tag{6}$$

We use the log probability,

$$\log p(\mathbf{l}|\mathbf{b}) = \sum_{t=1}^{|\mathbf{b}|} \log p(l_t|l_{t-1}) + \log p(b_t|l_t). \tag{7}$$

We use the Viterbi algorithm [19] to compute this log-likelihood.

We want to determine the most likely class:

$$\arg\max_{\mathbf{c}} \log p(\mathbf{c}|\mathbf{b}). \tag{8}$$

The HMM is the most versatile classifier, robust to varying geographical speeds because it classifies based on throughput sequence, and using a sequence to infer not only path, but location. However, if speed does not typically vary much but is unique to a certain path, then this classifier may be less useful than non-Markov models, since the former discards this information.

### 4.3 Sequence-based $k$-Nearest Neighbor Classifier

In the $k$-NN and NB-KDE classifiers, we make an assumption that subjects traveled along the path at consistent speeds. Instead of considering discrete geographic locations as we did with the HMM classifier, we represent location as equivalent to the number of seconds a user has traveled along a path. In other words, $\mathbf{c} = l_0, l_1, \ldots$ represents a virtual location for each second along a path, rather than directly mapping to a geographic location.

First, we compute a distance between the test trace $\mathbf{b}$ and training traces $\mathbf{b}^{tr} \in B^{tr}$.

$$\text{distance}(\mathbf{b}, \mathbf{b}^{tr}) = \sum_{t=0}^{|\mathbf{b}|} |\mathbf{b}_t - \mathbf{b}_t^{tr}| \tag{9}$$

Subsequently, we rank the sequences $\mathbf{b}^{tr}$ by the computed distance from lowest to highest. We classify the instance as the label (i.e., the route) present in the largest fraction of the $k$ nearest neighbors. If there is a case of a tie, we increment $k$ for that case until the tie is broken.

The choice of $k$ tunes a smoothing effect in the data: A larger $k$ reduces erroneous labeling due to matching against outliers, while too large of a $k$ can result in simply choosing the most common training label.

This classifier is powerful because it does not try to model any attributes about the geographic paths — assumptions which may turn out to be incorrect. It simply votes based on similarity to the labeled traces in the database. However, its accuracy depends on there being enough training traces to properly match the unknown trace.

### 4.4 Naive Bayes with a sequence of kernel density estimators

The NB-KDE classifier makes a similar assumption to the $k$-NN classifier and assumes that subjects are traveling at consistent speeds. The algorithm obtains a distribution of throughputs for each location state, and finds the closest matching sequence of distributions to the observed throughputs.

Similar to $k$-NN, $\mathbf{c} = l_0, l_1, \ldots$ represents seconds along the path; thus, we do not consider transitions.

We wish to determine

$$\arg\max_{\mathbf{c}} p(\mathbf{c}|\mathbf{b}). \tag{10}$$

Each location is associated with a kernel density estimator, with Gaussian kernel $K$:

$$f(b|l) = \frac{1}{N_l} \sum_{i=0}^{N_l} K(b - b_i^l). \tag{11}$$

We use the KDE to estimate the likelihood of a certain bandwidth at a location, so

$$p(b_t|l_t) = f(b_t|l_t). \tag{12}$$

We consider each second as independent features in the naive Bayes algorithm. We compute the likelihood of a class $c$ by multiplying the likelihood at each second, and choosing the most likely class.

$$\arg\max_{\mathbf{c}} p(\mathbf{c}|\mathbf{b}) = \arg\max_{\mathbf{c}} \prod_{t=0}^{|\mathbf{c}|} f(b_t|l_t^{\mathbf{c}}) \tag{13}$$

A kernel density estimator captures the varying distributions of throughputs at each location, as well as cases where there may be more than one distribution of throughputs at a location (for example, if the location is serviced by more than one possible cell tower). In other words, unlike the $k$-NN, it does not require every throughput sequence scenario to be captured in the training traces; rather, it measures the distributions at each second. By learning this information, the classifier performs much more accurately than the other two classifiers.

## 5 Experimental Results

We evaluate several scenarios against the classifiers described in Section 4. Below, we describe these scenarios with respect to our attacker model, and then discuss the limitations of our classifiers in different situations.

**Summary of Experiments and Results.** First, we show that an attacker can differentiate a mobile user from a stationary user, that is, make the binary choice of mobile or stationary. The attack succeeds with very high accuracy (100% in the *metro* dataset for NB-KDE). Next, we show that given a user's starting or ending location and choice of four paths, an attacker can determine which path a user traveled, that is, it can choose the origin or destination correctly among four suburbs. This attack also succeeds with high accuracy (83.0–93.0% for NB-KDE depending on the scenario).

We then use our data to explore how well our method will scale to more choices. We show that given just the choice of four paths, an attacker can determine both the path and direction traveled (from among the eight possibilities) with good accuracy (75.7% for NB-KDE). This problem parallels the problem of choosing from one of eight known paths given a starting or ending point, but is no easier: when forced to determine both path and direction, the attacker is choosing among paths where pairs are quite similar in some respects (since they are essentially mirror images of one another). We also show that in our data, the mirroring accounts for some drop in accuracy, though it is less pronounced in the NB-KDE classifier.

Finally, we show that an attacker can identify a user's path with a fine granularity, determining the direction and path of travel within campus with high accuracy (76.5% for NB-KDE and 81.1% for $k$-NN).

### 5.1 Overview

Our experiments take the form of classification problems, where an attacker trains a classifier on data consisting of labeled sequences of throughput as training instances, as described in Section 4, and attempts to determine the class of an unlabeled test instances. Varying the classifier and training and testing data allow us to determine how well the attacker can perform under different scenarios.

**Classifiers.** We evaluate an attacker's accuracy using the $k$-nearest neighbor, naive Bayes KDE (NB-KDE), and throughput-based HMM classifiers described in Section 4. We also evaluate two simplistic approaches to classification. In the first simplistic approach, called *Throughput*, the classifier models each path as the mean of the mean of the throughputs associated with each path. This is among the simplest approaches to modeling a path that actually uses observed throughputs. In the other simplistic approach, called *Frequency*, the classifier simply chooses the class that was most common in the training data. Performing no better than either simplistic approach, which use no information from the data stream, implies that a classifier models the situation poorly.

In all experiments, we use the standard definition of *accuracy* for a multi-class problem: the sum of correct classifications divided by the total number of classifications. We determined optimal parameters for each classifier in separate training sets using 3-fold cross-validation. These were the value of $k$ for the $k$-NN, the bandwidth selection method (Silverman's [16] or Scott's Rule [15]) for the NB-KDE, or the window size $w$ for the HMM. Using these parameters within the held-out test set, testing and training was done using leave-one-out cross-validation. We evaluate the named classifiers themselves, not any of the specific parameters.

**Assumptions and Limitations.** Our attacker model and assumptions are described in Section 2. Our techniques assume the attacker knows the starting (or ending) location of the user. We assume the attacker has trained on all possible paths the user could have taken from (or to) that location.

Our measurement study is limited to one geographic location that is a small town with few tall buildings, as well as several paths to surrounding towns. Other small cities may be different, and cities replete with tall buildings may have very different characteristics. The speed of the mobile node was dictated by local traffic. We have no data on walking or bicycling targets.

We used Subsonic[3] to stream a constant bitrate mp3 from our server to the phone, resetting the cache before each run. However, a real target might not stream data the entire length of the path (or at all), and hence we gave the attacker an advantage. Further, we did not model the complexities of commercial streaming services, which may not stream from a single location on the network. Finally, our measurements do not include any competing traffic flows to or from the

3. http://www.subsonic.org

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



| Path | Classes | NB-KDE | $k$-NN | HMM | Throughput | Frequency |
|---|---|---|---|---|---|---|
| A-to-X vs stationary | 2 | **100.0** | **97.7** | 67.8 | 65.2 | 68.5 |
| B-to-X vs stationary | 2 | **100.0** | 92.2 | 45.1 | 86.8 | 54.7 |
| C-to-X vs stationary | 2 | **100.0** | **96.5** | 49.1 | 74.1 | 50.0 |
| D-to-X vs stationary | 2 | **100.0** | **93.8** | 39.6 | 87.5 | 60.4 |
| X-to-A vs stationary | 2 | **100.0** | 93.3 | 68.9 | 69.1 | 70.1 |
| X-to-B vs stationary | 2 | **100.0** | **96.3** | 48.1 | 89.5 | 50.9 |
| X-to-C vs stationary | 2 | **100.0** | 93.0 | 49.1 | 83.3 | 51.7 |
| X-to-D vs stationary | 2 | **100.0** | **94.1** | 43.1 | 77.4 | 54.7 |
| E-to-F vs stationary | 2 | **68.6** | **72.5** | 47.1 | **60.5** | **66.3** |
| F-to-E vs stationary | 2 | **68.9** | **64.4** | 37.8 | **76.2** | **72.4** |
| G-to-H vs stationary | 2 | **87.0** | 60.9 | 58.0 | 54.9 | 64.6 |
| H-to-G vs stationary | 2 | **96.2** | 74.4 | 64.1 | 56.5 | 65.9 |

Fig. 7. Classification accuracy for differentiating *stationary* vs. *mobile users*. Bolded entries have the highest accuracy; also bolded are entries that are not statistically different from the highest rate (two-sided, two-sample proportion test; 95% c.i.) The NB-KDE classifier performs flawlessly for the "metro" traces.

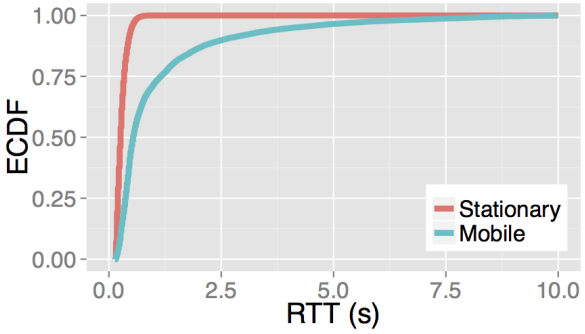


Fig. 8. The empirical CDF of server-side roundtrip time (RTT) estimates for long-running TCP connections when a mobile device is static or moving. The same device is used for both cases. The static scenario demonstrates a noticeably different distribution of estimated RTTs compared to the mobile scenario; RTT is a primary factor in TCP throughput [13].

phone during trace collection, which may complicate classification in practice.

### 5.2 Differentiating Stationary and Mobile Users

In our first experiment, we compare our instances of data from stationary phones against those from mobile phones. In this set of experiments, all stationary traces are one class, and mobile traces corresponding to each of the routes shown in Figure 1 are treated as the other class. The results are shown in Figure 7. We see that the NB-KDE classifier works perfectly, and the $k$-NN classifier is near perfect. Both are noticeably better than simply comparing raw throughput information. As we show in Figure 8, there are obvious differences in RTT estimates, and RTT is a prominent factor in TCP throughput [13]. Dramatic variations of the estimated RTT are likely the result of the increased number of local link-layer retransmissions, which seek to mitigate the impact of wireless losses on TCP [4]. These retransmissions are more common in our mobile scenarios.

### 5.3 Determining a User's Path

In our next set of experiments, we assume that the user's starting or ending location is known, and that our goal is to determine which of four paths were taken by the device. In these experiments, we truncated each trace to the first ten minutes to avoid biases introduced by the varying trace length between each class. In the *metro* situation, we train and test classifiers using only traces that are *Inward* to the central location ("... to X") described in Section 3; then we do so again, using only *Outward* ("X to ...") traces. Each set of these experiments considers four classes. We also classified path and direction (8 classes). The results of these experiments are shown in Figure 9.

The NB-KDE and classifier significantly outperform the naive classifiers. The HMM does not work when there is a time-independent sequence of throughputs that are similar between two paths. This is particularly the case for paths X-to-C and X-to-A, which are both paths away from the more urban location X. If the value of $k$ is tuned, the classifier achieves greater than 70% accuracy [17]; however, it does not perform well if $k$ is chosen using a separate training set. Figure 12 shows the confusion matrix of the NB-KDE results. It misclassified the "A-to-X" path direction particularly often, indicating that the first and last ten minutes of this path are relatively symmetrical.

In the *campus* situation, both the $k$-NN and NB-KDE performed well. As shown in Figure 12, The NB-KDE classifier often misclassified the "F-to-E" as "H-to-G"; these are both south-to-north paths, which go from low-throughput to high-throughput. In both cases, the HMM performed poorly because while it

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



| Data Set | Experiment | Classes | NB-KDE | $k$-NN | HMM | Throughput | Frequency |
|---|---|---|---|---|---|---|---|
| Metro | 4 paths× 1 (Outward) | 4 | **92.0** | 52.9 | 29.7 | 48.3 | 45.0 |
| | 4 paths× 1 (Inward) | 4 | **93.0** | 47.3 | 69.8 | 48.9 | 46.7 |
| | 4 paths × 2 | 8 | **83.0** | 35.6 | 11.5 | 26.6 | 23.8 |
| Campus | 2 paths × 2 | 4 | **76.5** | **81.1** | 43.2 | 48.8 | 31.4 |

Fig. 9. Classification accuracy depending on which roads are included in the experiment. Bolded entries are not significantly less accurate than the most accurate result. Outward indicates "X"-to-. . . , and inward indicates the opposite. Traces were truncated to 10 minutes long.

| Data Set | Experiment | Classes | NB-KDE | $k$-NN | HMM | Throughput | Frequency |
|---|---|---|---|---|---|---|---|
| Metro | A-to-X vs X-to-A | 2 | **88.3** | 61.4 | 33.3 | 61.1 | 51.9 |
| | B-to-X vs X-to-B | 2 | **96.8** | 62.6 | 43.5 | 57.7 | 53.8 |
| | C-to-X vs X-to-C | 2 | **94.3** | 66.5 | 29.6 | 50.0 | 51.7 |
| | D-to-X vs X-to-D | 2 | **95.5** | 63.8 | 90.2 | 67.4 | 55.8 |
| Campus | E-to-F vs F-to-E | 2 | **88.4** | 55.1 | **78.3** | 63.2 | 57.1 |
| | H-to-G vs G-to-H | 2 | **99.0** | 82.3 | 74.0 | 59.6 | 51.4 |

Fig. 10. The NB-KDE was able to differentiate between directions for most paths. The exception was "X-to-D" which was relatively symmetrical. Bolded entries are not significantly less accurate than the most accurate result.

may be able to learn the sequence of throughputs associated with each path and direction, it discards all timing information. Upon further analysis of the guessed states, there were many cases where the HMM assumed that the subject traveled either exceptionally quickly or slowly.

As we discussed in Section 3, the reason classification is possible is that throughput is geographically consistent in our dataset (see Figure 4). The experiments in this section demonstrate that path-level classification can make meaningful use of such consistency. In line with our intuition, we find that naive measurements of throughput alone does not adequately differentiate routes, but classifiers perform significantly better.

We found a positive correlation of 0.15 between throughput and trace, shown in Figure 3. The lower correlation — compared to a correlation of 0.24 at the per-second level — speaks to the challenge of this task: network performance is generally consistent for a path but it weakens significantly for shorter time scales. Additional features from the network traffic are likely needed to advance classification accuracy to work with finer time scales or geographies. Mobile usage patterns may also be useful in accounting for variances between traces [9]. A non-Markovian hidden state model [10] may be able to account for varying speeds, while still preserving timing information that is consistent within a class.

**Determining direction given a path.** As Figure 10 shows, the NB-KDE is best able to classify the direction given a path. The NB-KDE performs well because it differentiates the unique seconds along a path in each direction. Occasionally, the HMM does well: it benefits from the duration and length of both directions in each class being similar, thus it is unlikely to incorrectly infer a class due to misalignment. As well, the general trend of throughput decreasing in the "X-to-. . . " *metro* traces, or north-to-south *campus* traces is easily modeled. The $k$-NN sometimes fails by erroneously matching the middle of each path, which are likely consistent in both directions.

### 5.4 Effect of trace duration on accuracy

A potential target may wish to protect her privacy by limiting her session length. We investigate the accuracy of each classifier, given different trace lengths.

Figure 11 shows the accuracy the classifiers given trace durations of 1–600 seconds. All three classifiers classify the *campus* paths more accurately when traces are longer. Most traces were classified within five minutes. In the 8-class *metro* experiment, the NB-KDE far surpassed the accuracy of the two other classifiers at all lengths. The HMM did not work in this case — wide variations in throughput of all traces strongly biased the classifier towards one class (i.e., path). The $k$-NN classifier improved linearly with trace length. The NB-KDE notably could identify traces with approximately 60% accuracy within one minute. This indicates that even short connection sessions may be enough to reveal information about a person's location.

### 5.5 Approaches to Enhancing Privacy

Besides limiting trace duration, we did not test any approaches for enhancing the privacy of users against this attack, but many existing techniques are likely to be effective. To prevent revealing their travel paths to nosy remote servers, phone users will need to traffic shape or otherwise perturb their data transmission, incurring a performance penalty.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2857777, IEEE Transactions on Mobile Computing



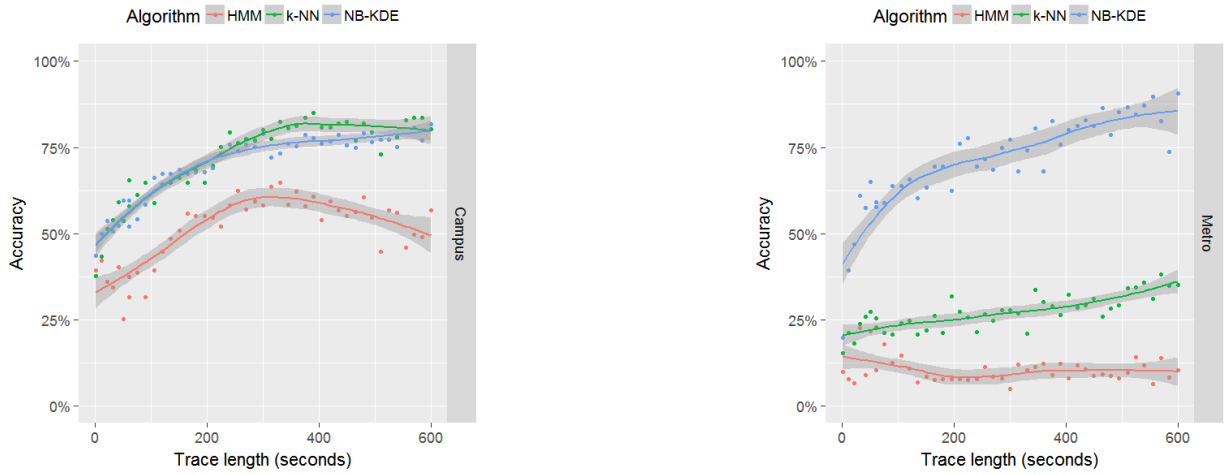


Fig. 11. We analyzed trace length vs accuracy to evaluate the efficacy of the attack if a connection session length is limited. The left graph indicates the *campus* experiment, and the right graph indicates the 8-class *metro* experiment (Figure 9 indicates the results at 600 seconds). The HMM performs poorly in both cases because it confuses classes in which similar sequences, regardless of timing, occur. This was particularly true among the outgoing paths in the *metro* data.

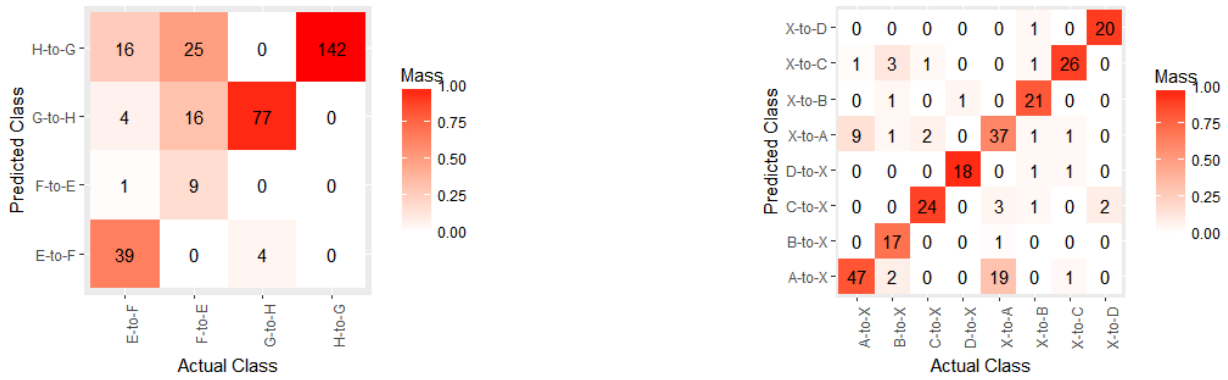


Fig. 12. Confusion matrices for the NB-KDE classifier in the corresponding *campus* and *metro* experiments for 600 seconds.

For example, a trusted proxy located outside the cellular network can re-shape traffic before reaching the attacker. As noted in Section 2, we assume the carrier is not assisting the attacker. The proxy could be set up as a VPN as a traffic shapper, which we suggest not for the encryption but because it is a protocol widely supported by smart phones as a transparent method of redirecting traffic.

It is feasible that the mobile device could reshape the traffic on its own. Most simply, it could limit throughput to a peak level that is reasonable across a wide area of the cell network. Or it could enforce regions of zero throughput. Of course, the challenge is to shape traffic in a way that does not overall reduce throughput or the interactivity needed by the application. This type of shaping would be easy for bulk file transfer where perhaps no interactivity is required. However, traffic shaping would be more challenging for interactive audio and video calls, where users are most sensitive to throughput limitations and network delay jitter. Traffic shaping of interactive web browsing may also be a challenge, but caching and pre-fetching may help mask throughput ceilings.

## 6 Related Work

**Mobile Phone Localization.** Precisely localizing mobile phones or other similar devices on the basis of GSM and other location-explicit information is an active area of research. However, these works use information available only to the mobile user (such as which 802.11 base stations or cell towers are in range [8], [18]) or their carrier (such as the pattern of handoffs [2] or other administrative details [22]). In our work, we focus on

*remotely* localizing another party based only on a TCP traffic stream rather than local information.

Kune et al. [12] propose a technique to test if a user is present within a small area or absent from a large area by simply listening on the broadcast GSM channel. The focus of their work is on lower layers of the GSM communication stack. We did not extend our study to analyze lower layers of 3G, because it is a legal violation in our jurisdiction. And again, our study is concerned with remote observation of network streams over cellular links, and largely treats the cellular infrastructure as a black box.

Xu et al. [22] present an approach for localizing performance measurements in 3G networks. They exploit the predictability of users' mobility pattern to develop a clustering algorithm for grouping related cell sectors and assigning IP performance measurements to fine-grained geographic regions. The proposed technique requires access to the cellular infrastructure. In contrast, our technique for network-based localization requires remote passive observation of the target, and data collection independent from the target and internals of the infrastructure.

Balakrishnan et al. [1] show that individual cell phones can expose different IP addresses to servers within time spans of a few minutes, and find that IP-based geo-localization is "impossible" in cellular networks. They show that application-level latencies can differ greatly among cities thousands of miles apart. Moreover, they show that the variation of latencies in short time spans is not high. Our work is complementary and extends similar notions further: we show the consistency of throughput at the finer granularity of square-kilometer regions, and we demonstrate successful classification experiments using such features.

Xu et al. [23] show that in contrast to wired Internet traffic, current cellular data traffic traverses through a limited number (4–6) of Gateway GPRS Support Node (GGSNs), which is the first IP hop of a data connection. The authors show that local DNS servers provide an appropriate approximation to estimate a user's network location (i.e., one of the 6 GGSNs) for purposes of mobile content placement and server selection, due to the restricted routing in cellular networks. By assuming availability of partial information about the possible routes a user could be on, our approach aims at a much more granular localization than the DNS method proposed by Xu et al., which is limited to finding approximate network locations.

**Other Remote Attacks.** Kohno et al. [11] present a technique for fingerprinting a physical device remotely by exploiting clock skews. Their approach could be used to remotely identify the same device connected to the Internet at different times or using different IP addresses. Our approach, which is focused on detecting the routes taken by a mobile node, is orthogonal to this work and could benefit from it when locating the endpoints of a target's travel path.

NAT and firewall policies of cellular carriers are explored in the work by Wang et al. [20]. They identify a set of such policies that directly impact performance, energy, and security of mobile devices. For instance, they show that NAT boxes and firewalls set timeouts for idle TCP connections, which sometimes lead to a significant waste of energy on the mobile device. The authors show that in spite of the deployment of firewalls, cellular networks are still vulnerable to denial of service and battery draining attacks. In contrast, we explore another type of attack on the location privacy of mobile cellular users.

Perta et al. [14] identify mobile phone IP addreses by sending several PUSH notifications to target devices and measuring round-trip times, assuming the network carrier assigns public IP addresses. They exploit variations in the cell network architecture to narrow down a user's IP to about 1000 addresses within 20 messages. This may be enough to identify coarse grained location information about the user.

## 7 Conclusion and Open Problems

We have demonstrated that the patterns of data transmission between a server on the Internet and a moving cell phone can reveal the geographic travel path of that phone. While the GPS and location-awareness features on phones explicitly share this information, phone users will likely be surprised to learn that disabling these features does not suffice to prevent a remote server from determining their general mobility. Our work shows that a naive Bayes classifier with sequences of kernel density distributions can discover and exploit features of the geography surrounding possible travel paths to determine the path a phone took, using only data visible at the remote server on the Internet and training data collected independently.

While we had hypothesized that simpler alignment methods such as dynamic time warping [3] or hidden Markov models could improve accuracy, these failed because they discard timing information that is consistent within a path. More complex alignment such as hidden semi-Markov models [10] may be more of a threat in this regard. If the attacker has access to much more training information (for example, via a popular mobile application), they may be able to model a user's position in 2-D space using a Gaussian process [6], rather than only along a specific path.

In the present article, we have demonstrated that the attack can be successful with traces that are only one minute long. It is an open and important problem to quantify the extent to which a user's location can be compromised in this fashion — with greater accuracy and among larger numbers of paths and

different geographies — and to determine just how much information is needed to make these inferences.

**Acknowledgements.** This work was supported in part by NSF award CNS-0905349. We thank Hamed Soroush for his work on the original version [17] of this paper, and Mark Corner for early discussions of this work. Non-student authors are listed alphabetically.

## REFERENCES

[1] Balakrishnan, M., Mohomed, I., Ramasubramanian, V.: Where's that phone? Geolocating IP addresses on 3G networks. In: ACM IMC. pp. 294–300 (2009)
[2] Becker, R.A., Caceres, R., Hanson, K., Loh, J.M., Urbanek, S., Varshavsky, A., Volinsky, C.: Route classification using cellular handoff patterns. In: ACM UbiComp. pp. 123–132 (2011)
[3] Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD workshop. vol. 10, pp. 359–370. Seattle, WA (1994)
[4] Chan, M.C., Ramjee, R.: TCP/IP performance over 3G wireless links with rate and delay variation. In: ACM MobiCom. pp. 71–82 (2002)
[5] Eddy, S.R.: Hidden markov models. Current opinion in structural biology 6(3), 361–365 (1996)
[6] Ferris, B., Fox, D., Lawrence, N.D.: Wifi-slam using gaussian process latent variable models. In: IJCAI. vol. 7, pp. 2480–2485 (2007)
[7] Goldstein, M.L., et al.: Mobile Device Location Data (United States Government Accountability Office). http://www.gao.gov/assets/650/648044.pdf (Sept 2012)
[8] Hightower, J., LaMarca, A., Smith, I.: Practical Lessons from Place Lab. IEEE Pervasive Computing 5(3), 32 –39 (July-Sept 2006)
[9] Jo, H.H., Karsai, M., Kertész, J., Kaski, K.: Circadian pattern and burstiness in mobile phone communication. New Journal of Physics 14(1), 013055 (2012)
[10] Johnson, M.J., Willsky, A.S.: Bayesian nonparametric hidden semi-markov models. Journal of Machine Learning Research 14(Feb), 673–701 (2013)
[11] Kohno, T., Broido, A., Claffy, K.: Remote physical device fingerprinting. IEEE Trans. on Dependable and Secure Computing 2(2), 93–108 (May 2005)
[12] Kune, D.F., Koelndorfer, J., Hopper, N., Kim, Y.: Location leaks on the GSM Air Interface. In: ISOC NDSS (Feb 2012)
[13] Padhye, J., Firoiu, V., Towsley, D.F., Kurose, J.F.: Modeling TCP Reno Performance. IEEE/ACM Trans. Netw. 8(2), 133–145 (Apr 2000)
[14] Perta, V.C., Barbera, M.V., Mei, A.: Exploiting delay patterns for user ips identification in cellular networks. In: International Symposium on Privacy Enhancing Technologies Symposium. pp. 224–243. Springer (2014)
[15] Scott, D.W.: Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons (2015)
[16] Silverman, B.W.: Density estimation for statistics and data analysis, vol. 26. CRC press (1986)
[17] Soroush, H., Sung, K., Learned-Miller, E., Levine, B.N., Liberatore, M.: Turning Off GPS is Not Enough: Cellular location leaks over the Internet. In: Proc. Privacy Enhancing Technologies Symposium (PETS). pp. 103–122 (July 2013), http://forensics.umass.edu/pubs/soroush.pets.2013.pdf
[18] Thiagarajan, A., Ravindranath, L., Balakrishnan, H., Madden, S., Girod, L.: Accurate, Low-Energy Trajectory Mapping for Mobile Devices. In: USENIX NSDI (Mar 2011)
[19] Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13(2), 260–269 (April 1967), http://dx.doi.org/10.1109/TIT.1967.1054010
[20] Wang, Z., Qian, Z., Xu, Q., Mao, Z., Zhang, M.: An untold story of middleboxes in cellular networks. In: ACM SIGCOMM. pp. 374–385 (Aug 2011)
[21] Xing, Z., Pei, J., Keogh, E.: A brief survey on sequence classification. SIGKDD Explor. Newsl. 12(1), 40–48 (Nov 2010)
[22] Xu, Q., Gerber, A., Mao, Z., Pang, J.: AccuLoc: practical localization of performance measurements in 3G networks. In: ACM MobiSys. pp. 183–196 (Aug 2011)
[23] Xu, Q., Huang, J., Wang, Z., Qian, F., Gerber, A., Mao, Z.: Cellular data network infrastructure characterization and implication on mobile content placement. In: ACM SIGMETRICS. pp. 317–328 (2011)

**Keen Sung** is a PhD Candidate in the College of Information and Computer Sciences at UMass Amherst. His research focuses on time series data analysis and mobile privacy.

**Joydeep Biswas** is an Assistant Professor in the College of Information and Computer Sciences at UMass Amherst. His research interests include perception, planning, and control of autonomous mobile robots.

**Erik Learned-Miller** is a Professor in the College of Information and Computer Sciences at UMass Amherst. His research focuses on various aspects of computer vision.

**Brian Neil Levine** is a Professor in the College of Information and Computer Sciences at UMass Amherst. His research focuses on privacy, digital forensics, networking, blockchains, and the Internet.

**Marc Liberatore** is Teaching Faculty in the College of Information and Computer Sciences at UMass Amherst. His research interests include digital forensics, privacy, and p2p systems.