Understanding the Potential Risks of Sharing Elevation Information on Fitness Applications

Ülkü Meteriz
University of Central Florida
Orlando, USA
meteriz@knights.ucf.edu

Necip Fazıl Yıldıran
University of Central Florida
Orlando, USA
yildiran@knights.ucf.edu

Joongheon Kim
Korea University
Seoul, South Korea
joongheon@korea.ac.kr

David Mohaisen
University of Central Florida
Orlando, USA
mohaisen@ucf.edu

***Abstract*—The extensive use of smartphones and wearable devices has facilitated many useful applications. For example, with Global Positioning System (GPS)-equipped smart and wearable devices, many applications can gather, process, and share rich metadata, such as geolocation, trajectories, elevation, and time. For example, fitness applications, such as Runkeeper and Strava, utilize information for activity tracking, and have recently witnessed a boom in popularity. Those fitness tracker applications have their own web platforms, and allow users to share activities on such platforms, or even with other social network platforms. To preserve privacy of users while allowing sharing, several of those platforms may allow users to disclose partial information, such as the elevation profile for an activity, which supposedly would not leak the location of the users. In this work, and as a cautionary tale, we create a proof of concept where we examine the extent to which elevation profiles can be used to predict the location of users. To tackle this problem, we devise three plausible threat settings under which the city or borough of the targets can be predicted. Those threat settings define the amount of information available to the adversary to launch the prediction attacks. Establishing that simple features of elevation profiles, e.g., spectral features, are insufficient, we devise both natural language processing (NLP)-inspired text-like representation and computer vision-inspired image-like representation of elevation profiles, and we convert the problem at hand into text and image classification problem. We use both traditional machine learning- and deep learning-based techniques, and achieve a prediction success rate ranging from 59.59% to 95.83%. The findings are alarming, and highlight that sharing elevation information may have significant location privacy risks.**

I. Introduction

From smartphones to wearable devices, various types of Internet of Things (IoT) devices are equipped with Global Positioning System (GPS), accelerometers and gyroscopes to allow applications to function or to present a better user experience by making use of *geodata*, such as location and elevation information. Specifically, fitness applications which run on smartphones and smartwatches use these systems to collect spatial, temporal, and activity-specific information to analyze, summarize and visualize users' activities. By analyzing each activity, many of those applications even deliver personalized motivations and challenges for users to meet their goals. Using social media support of these applications for sharing updates about users' activities, including training routes and elevation profiles for the routes taken for the activity (e.g., walking, running, climbing, cycling), users can have positive behavioural changes through a more active lifestyle motivated by competitions with friends and acquaintances [1].

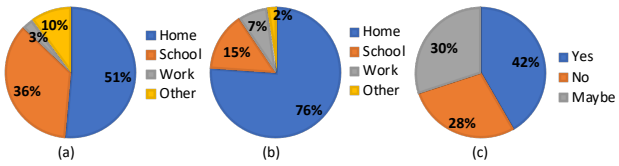


Fig. 1. Survey results for understanding users behavior: (a) starting point statistics, (b) end point statistics, and (c) not sharing location information implies privacy. While 90% of the 60 participants indicated their start of activity is either home, school, or work, an overwhelming 98% of the participant indicated those to be the end point of their activities.

Despite the broad set of advantages geodata has, geodata usage and uncontrolled sharing can pose a significant privacy risk which can be exploited in multiple attacks, including stalking [2] and cybercasing [3]. For example, with the large amount of geo-tagged data, including text, images, and videos, cybercasing allows a significant attack vector to criminals and maliciously motivated individuals. Geo-tagged photos that are frequently posted on image sharing websites, such as Flickr, or second-hand sale websites, such as craigslist, may put owners of those images at risk. For example, geo-tagged images posted on sales websites may reveal the location of the advertised product, leading to trespassing or even theft.

While geodata recorded by fitness applications is indeed important and valuable for the operation of those applications, it also can be used for launching attacks on users by breaching their privacy, since sensitive information of users such as home or workplace location can be easily inferred from such data. Even worse, a large number of users, when sharing such information, would be unaware of the ramifications of sharing, and the potential risk of inferring such contextual information (home, work location, etc.) from such shared location data. To support this argument, we conducted an online survey with 60 participants who regularly use fitness applications outdoors, with the results of the survey summarized in Figure 1. The survey results reveal that 51% of the participants start their training from their homes, 36% start from their school and 3% start from their workplace, and 76% of the participants finish their training at their homes. Moreover, for the same set of users, 42% of those users have indicated that not sharing

location information implies privacy protection, while 30% of the respondent where uncertain, and 28% were certain that not sharing would not necessarily mean their privacy is protected. This kind of mixed responses highlight the gap between reality and expectations of privacy when sharing location information online and call for further investigation.

Although it is possible not to share the location trajectory by hiding the activity map in the fitness applications, users still want to share elevation profile or certain statistics of the activity to show the roughness, technicality, and difficulty of the routes they took as a measure of their workout. For example, up until recently, users have been demanding those fitness applications to allow for fine-grained and customized access control by allowing them to share, for example, the elevation profile of an activity while masking the map that highlights the actual trajectory, which is deemed of high privacy value to them [4]–[7]. In the same survey conducted earlier, we asked our 60 subjects if "while sharing an outdoor workout record, do you think hiding the map and sharing only the statistics of your training (such as speed and elevation changes) is enough for protecting your privacy?". The results were overwhelmingly positive, with 25 of them indicating "yes", 18 indicating "maybe" (together accounting for more than 71%), while only 17 indicating "no".

However, is sharing the elevation profile of an activity enough to maintain the privacy of the users? In this paper, we argue that an approximate location, extracted from contexts of activities, and at different levels of location granularity, could still be revealed from elevation profile information. We examine this problem at length, and develop approaches that can be used to accurately associate an elevation profile with contextual information, such as the location.

Contributions. In this paper, we contribute the following:

- We translated the problem into text classification and image classification problems by encoding the elevation signals as strings and visualizing the elevation signals as images to employ the common approaches for solving image and text classification problems,
- We investigated the possible attack surface for the problem by introducing three different threat models, which we later used to evaluate the success of our approaches by simulating our methods considering each threat model,
- We proved that location information can be predicted from elevation profile using different machine/deep learning methods with accuracy in range 59.59% - 95.83% at different resolutions as our results showed.

II. Threat Models & Approach Overview

We outline the potential threat models under which this study is conducted. We describe three models under which the location privacy is breached only from associated elevation profiles. We then review our approach, including a pipeline that consists of data collection, preprocessing, feature extraction, and multi-class classification for location identification through elevation profiles. We briefly discuss the phases of our pipeline, each of which is explained in details in the Implementation Details section.

We note that the following threat models are only hypothetical: no attacks were actually launched on any users. As mentioned earlier, this study in its entirety is motivated by the aforementioned demands of users to have more flexibility over sharing partial data, such as elevation profiles, and examines the ramifications of such sharing in a hypothetical setting. We note, however, that those settings are also plausible if such sharing is enabled.

A. Threat Models

Our study utilizes three threat models: TM-1, TM-2, and TM-3, which we outline below with their justifications. The adversarial capabilities in TM-1 are greater than in TM-2 and TM-3, making it more a restrictive (powerful) model.

❶ **TM-1.** In TM-1, we assume an adversary with records of the workout history of a target user, and the goal of the adversary is to identify the last workout location of the target user from the shared elevation profiles. TM-1 is justified by multiple plausible scenarios in practice. For example, such an adversary might have been a previous social network connection of the target user that was later blocked. In such a scenario, the adversary may have previous workout records of the target from which the adversary may attempt to de-anonymize the target's activities. Another example might include group activities, in which two individuals (i.e., the adversary and target) may have shared the same route at some point. In either case, by knowing the target's history, the main goal of the adversary in this model is to identify recent whereabouts only from publicly shared elevation profiles in workout summaries, thus breaching the target's privacy.

❷ **TM-2.** In TM-2, we assume an adversary with access to limited information such as the city in which the target lives. Such information is easily accessible from public profile summaries, athlinks.com, public records, etc. The adversary's goal in TM-2 is to find out which region or part of a given city the target's activities are associated with. The TM-2 use scenario may include a targeted user sharing private activities, in which the route is hidden while the elevation profile is shown. The adversary, knowing the city where the target lives, would want to identify the region (e.g., borough in the city) associated with the user's activity.

❸ **TM-3.** In TM-3, we assume an adversary trying to identify the target user's city using only publicly shared elevation profiles. We assume, however, the adversary has the ability to profile the elevation of cities, with information that is easily obtained from public sources (e.g., Google Maps, OpenStreetMap). The use scenario of TM-3 may be used as a stepping stone towards launching the attack scenario in TM-2 upon narrowing down the search space to a city.

B. Approach Overview

In this subsection, we give a brief overview of our pipeline, which consists of the data collection, preprocessing, feature extraction and classification as illustrated in Figure 2.

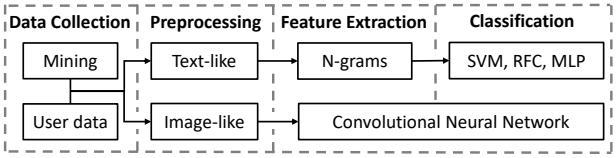


Fig. 2. The end-to-end pipeline of the approach.

Data Collection. We collected three datasets with varying and rich characteristics, namely 1) user-specific activity data collected from an athlete, 2) mined training route segments grouped at city-level, and 3) mined training route segments grouped at borough-level. For the user-specific dataset, we collected physical activity records of athletes and converted those activities to our intermediate format, the GPS Exchange Format (GPX). Then, we parsed the GPX files and manually labeled them according to the latitude and longitude information included within each file. For the second dataset, we mined training route segments from a popular fitness tracking website by specifying the location boundaries, *i.e.,* the class label of the mined data, and augmented each segment with the corresponding elevation profiles obtained from Google Maps Elevation API. Finally, the borough-level dataset is constructed in a similar manner as in the city-level dataset.

Preprocessing. We use both natural language processing and image processing techniques for extracting features from the elevation profiles in order to associate them with a given label. To this end, the preprocessing phase is two parts: text-like and image-like representations. For text-like representation, we discretized the elevation signals and computed the minimum required *word* size. We then created a mapping between each unique discrete value and a string. By mapping the string correspondents to the unique discrete values, we encoded the elevation profiles in text. Finally, we form a *vocabulary* from the text sequences of each dataset using the n-grams.

To obtain image-like representation, we converted the elevation profiles to a fixed-sized line graph where the x-axis stands for time and y-axis stands for the elevation values. The lines in the graphs are also colored to represent the elevation interval in which the elevation profiles range.

Feature Extraction. The classification algorithm operates on high quality and discriminative features, obtained from the representations of elevation profiles. For feature extraction, we utilize Natural Language Processing (NLP) and computer vision approaches. To employ NLP approaches, using previously obtained vocabulary, we represent each elevation profile as a feature vector based on the frequency of the vocabulary in the text-like representation (bag-of-words vector). To employ computer vision approaches, we utilize Convolutional Neural Network (CNN) over image-like representations. The optimal features of an image-like representation are efficiently extracted by the convolutional and pooling layers in the Convolutional Neural Network architecture.

Multi-class Classification. We use various machine learning and deep learning models for classification including Support Vector Machine (SVM) and Random Forest Classification (RFC) as machine learning approaches, and Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) as deep learning approaches.

III. Implementation Details

The implementation details of data collection, preprocessing, feature extraction and multi-class classification are addressed in the following subsections.

A. Data Collection

In this study, we compiled three datasets: the user-specific dataset, the city-level dataset, and the borough-level dataset. The user-specific dataset is retrieved from a voluntary athlete who frequently records activities. It offers a dense and thorough coverage for regions frequented by the user; those regions are used as class labels. The city- and borough-level datasets are created from scratch by collecting location trajectories that are created and frequented by the athletes. Both city-level and borough-level datasets provide sparse coverage of cities and boroughs.

1) User-Specific Dataset: For the user-specific dataset, we collected activity data including each activity's location trajectory and the corresponding elevation profile from a voluntary athlete who records activities frequently. First, the location trajectories included in the user-specific dataset are converted to GPX format to avoid confusion caused by different formats and settings. Then, to label the samples, the maximum and minimum coordinates of each location trajectory are fetched. Each sample location trajectory is encapsulated with a tight rectangle whose top right (North East) and bottom left (South West) corners are computed from the maximum and the minimum coordinates of the trajectory as illustrated in Figure 3. To classify the samples, each rectangle encapsulating the trajectory is compared with the previously created regions. If the Euclidean distance between the center of the rectangle and the center of the existing region does not exceed a predetermined threshold, the rectangle and its corresponding sample are labeled with a unique identity of the region. If there is no region that includes the trajectory, a new region is created. The final sample size distribution of user-specific dataset is shown in Table I.

The user-specific dataset is prone to having similar location trajectory portions across its samples since the user may frequent the same set of places in his/her everyday activities, such as the location trace they follow while leaving/arriving home, or their favorite routes. Therefore, we calculated the average overlap ratio of the routes included in the user-specific dataset by comparing each sample with the other samples with the same class label. For each sample pair comparison, the overlap ratio is calculated as the intersection over union of the tight rectangles encapsulating the sample routes. The average overlap ratio of the user-specific dataset is calculated as 35%.

2) City-Level Dataset: For the city-level dataset, we mined **publicly available** training route segments in a popular fitness tracking application using its

Fig. 3. An illustration of the tight rectangle encapsulating an example route.

TABLE I
USER-SPECIFIC DATASET SAMPLE SIZE DISTRIBUTION.

Regions	Sample Size
Washington DC	366
Orlando	232
New York City	120
San Diego	18

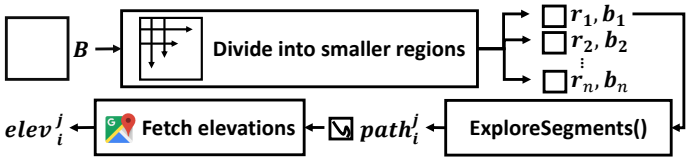


Fig. 4. An illustration of data mining pipeline.

EXPLORESEGMENTS() functionality. We note that our experiments do not put any users at risk, and are not in violation of the terms of use of the fitness tracking application: since both the trajectory (map) and elevation profiles are public, we are also not obtaining any information beyond what is provided by the users explicitly. We note that training route segments are user-created activity routes whose main purpose is to compare completion times among users who also completed the same route, and can be easily also obtained from other sources (e.g., Google street view). We note that these routes are particularly useful for the motivation of our work, since they include public location trajectories which are frequented by the actual users rather than randomly created location trajectories. During mining segments, the anonymity (thus the privacy) of the users who frequented the segments or created the segments is preserved. The overall data mining procedure consists of three steps as illustrated in Figure 4. First, we defined the cities of interest, which are also the class labels. For each city C, we defined rectangle boundaries B consisting of the top right and bottom left corner coordinates. Since EXPLORESEGMENTS() returns only the top-10 segments encapsulated by a given boundary, to obtain more data of a city C, we divided the large rectangle boundary of the city into smaller regions, each denoted by r_i, by following a grid-like structure as shown in the second phase of the Figure 4. Then, we computed region boundaries, each denoted by b_i. For each region boundary b_i, we called EXPLORESEGMENTS() and received the geolocation polyline path, $path_i^j$ where $j \in [1, 10]$, of the top-10 segments encapsulated in b_i. Finally, since the polyline paths do not include elevation profiles, we obtained the corresponding elevation profile $elev_i^j$ for each $path_i^j$ using the Google Maps Elevation API. The sample size distribution of city-level dataset can be found in Table II. Unlike user-specific dataset, city-level dataset does not include overlapped samples since each region r_i is disjoint with the other regions. A segment route that is included by more than one neighbour region are not considered since EXPLORESEGMENTS() returns the routes that are encapsulated within the given boundaries, b_i.

3) Borough-Level Dataset: For the borough-level dataset, we applied a similar mining procedure as of the city-level dataset, using the borough boundaries of a city instead of the city boundaries. Table III shows the sample size distribution of the borough-level dataset for different cities.

B. Preprocessing

We transform the samples to text- and image-like representations to facilitate feature extraction. We provide details on the preprocessing methods below.

1) Text-like Representation: To represent data as text-like contents, we use four steps, as in Figure 5: discretization, word size decision, text encoding, and vocabulary creation.

❶ In the discretization step, the original elevation signal is discretized along the y-axis, which represents the elevation values to avoid possible overhead by small differences in the precision causing a longer string correspondences and, consequently, longer vocabulary and sparse feature vectors. The discretization is done as follows. Let e_i^j be the i-th elevation value in j-th sample. The discretization functions are defined as $f(e_i^j) = \lfloor e_i^j \rfloor$ and $f(e_i^j) = \frac{\lfloor e_i^j \times 10^3 \rfloor}{10^3}$, where the first function is used for processing the user-specific dataset and the second function is used for processing the city-level and borough-level datasets. Since the user-specific dataset is dense in terms of sampling rate, using the floor function is enough to represent the routes. However, as the city-level and borough-level datasets are already sparse, losing information is undesired, so we used the second function to represent the elevations that differ in up to 3 decimal digits precision.

❷ For word size decision, we use $w = \log_l c$, where w is the word size, l is the length of the alphabet, and c is the number of unique value occurrences in the given signals.

❸ For text encoding, each unique value in all the discrete signals is mapped to a unique string with length w and each sample signal is encoded by referring to the string correspondences of each value in the discrete signal and concatenating these strings to construct a long text, *i.e.,* corpus.

TABLE II
CITY-LEVEL DATASET SAMPLE SIZE DISTRIBUTION.

Regions	Sample Size
New York City	2437
Washington DC	2129
San Francisco	743
Colorado Springs	369
Minneapolis	363
Los Angeles	280
New Jersey	266
Duluth	156
Miami	94
Tampa	83

TABLE III
BOROUGH-LEVEL DATASET SAMPLE SIZE DISTRIBUTION.

Cities	Regions	Sample Size
Los Angeles (LA)	Downtown	280
	Santa Monica	128
	Chinatown	46
	Beverly Hills	38
Miami (MIA)	Downtown	67
	Miami Beach	44
	Virginia Key	18
New Jersey (NJ)	Jersey City	266
	West New York	23
	Newark	28
New York City (NYC)	Manhattan	2437
	Queens	353
	Brooklyn(South)	239
	Brooklyn(North)	205
	Bronx	142
	Staten Island	119
San Francisco (SF)	South West	743
	South East	144
	North West	130
	North East	86
Washington DC (WDC)	District of Columbia	2129
	Baltimore	218

❹ To create our *vocabulary*, we consider the corpus created from all encoded signals regardless of labels. Each line in the corpus represents a sample signal, and each word in a line represents the text correspondence of an elevation value in the sample signal. We build a vocabulary from the unique word-based n-grams of the document. As illustrated in Figure 6, a window with size $W = w \times n$ is slided throughout the corpus and each window content is appended to the vocabulary set. Since the vocabulary set does not contain duplicate entries by definition, we constructed the vocabulary consisting of unique

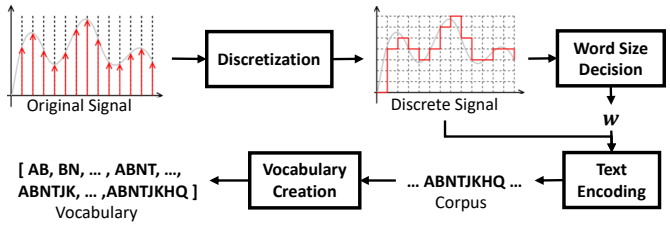


Fig. 5. Illustration of the flow of text-like preprocessing. The signal is discretized by eliminating the small elevation fluctuations. The discretized signal is also used for deciding the word size of the encoding. The discrete signal is then encoded in text and a vocabulary is built.

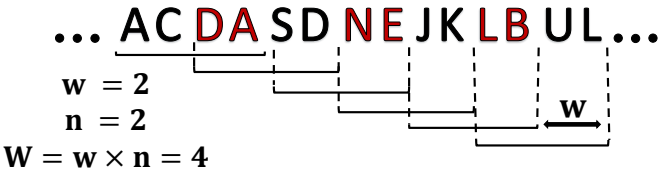


Fig. 6. Illustration of bi-gram creation where the word size is $w = 2$ and window size is $W = 4$.

n-grams of the given dataset after traversing the corpus by n times with different window sizes.

2) Image-like Representation: In image-like transformation, the elevation signals are drawn as line graphs. To draw a line graph, the maximum and minimum values for y-axis are set to be the extremes of each elevation signal, and the lines are colored to encode the value interval in which elevation signal ranges. This method has multiple advantages over other methods—e.g., the alterations of an elevation signal are more visible, and the method results in an efficient utilization of the feature space—which we examined to reach this design choice, but we omit due to lack of space. We use 200 elevation values for each, obtained by dividing the elevation signal into equal-sized parts.

C. Feature Extraction

To classify elevation profiles accurately, we extract discriminative features from the elevation profile representations.

Text-like. In text-like feature extraction, words and non-overlapping occurrences of word sequences are counted, a feature vector for each sample is created with each unique word sequence count being a feature. Finally, the feature vectors are normalized where each feature represents the probability of occurrence of each word in the given sample.

When the dataset is large and diverse, the vocabulary and, consequently, the feature vectors becomes too large and sparse to do computations with. In feature selection phase, in case the length of feature vectors are too long, some rarely occurred features from vocabulary are discarded according to the specified feature frequency threshold. Features are ordered by term frequency across the corpus and the features whose term frequency is under the specified threshold are discarded and a new vocabulary is created.

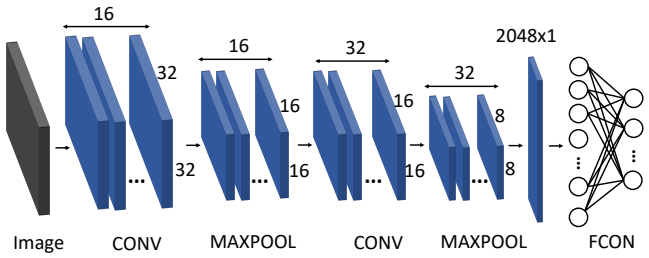


Fig. 7. The CNN architecture used for classification. The input image is passed to a CONV layer. The output is then forwarded to the MAXPOOL layer to fetch the most important feature in a kernel. The data is then passed through following CONV and MAXPOOL layers. The output retrieved from the last MAXPOOL layer is flattened to be passed to a FCON layer whose output is the class probabilities.

Image-like. Since we use CNN for images, it is unnecessary to explicitly extract features, since the convolutional layer kernels do that already by learning the filters optimally and efficiently. Therefore, the actual feature extraction mechanism is discussed in the context of classification.

D. Multi-Class Classification

For classification, SVM, RFC, MLP, and CNN are used.

SVM. We use the standard SVM, where the objective is to find the best hyperplane separating classes from one another.

RFC. We utilized the standard RFC, with 100 trees, and a majority voting is taken over the outcomes of those trees.

MLP. We use the standard MLP with 100 hidden layers and Adam solver [8] for weight optimization, since it is claimed to work well for large feature space. MLP is shown to outperform the decision trees [9], [10].

CNN. Figure 7 illustrates the employed CNN architecture. Two consecutive convolution layers (CONV) are used along with the ReLU activation function and MAX pooling layers (MAXPOOL) before a fully connected layer (FCON). For both of the convolution layers, kernel, stride and padding sizes are decided as 5, 1 and 2 respectively. The distinctive features are selected at the max pooling layers with kernel and stride size of 2, which reduce the dimensions from (32x32) to (8x8) at two passes. The loss is calculated by Cross Entropy Loss function and for optimization, the Adam optimizer is used.

IV. Evaluation, Results, and Discussion

We performed evaluations for each dataset and threat models. First, we performed evaluations with text-like representation. For TM-1, we performed experiments using 5- and 10-fold cross-validation methods and by fixing the dimension of n-grams to 8 on the user-specific dataset with SVM, MLP and RFC techniques. For TM-2, we tried to find the borough information of a given elevation profile whose city information is known. For this experiment, we use 10-fold cross validation, $n = 8$ for the n-gram, and SVM, MLP and RFC are employed. For TM-3, we use the same settings as in TM-2. The user-specific dataset contains overlapped and repetitive portions by nature. In the Simulations subsection, we simulated the same behaviour on the mined datasets and performed the same evaluations for comparison.

TABLE IV
TM-1 evaluation on user-specific dataset. Prediction accuracy (%) with different configurations. 4-class = [Washington DC, Orlando, New York City, San Diego], 3-class = [Washington DC, Orlando, New York City], 2-class = [Washington DC, Orlando]. 5-f: 5-fold cross-validation. 10-f: 10-fold cross-validation. C: the number of classes in the classification. S: sample size of each class.

		SVM		RFC		MLP	
C	S	5-f	10-f	5-f	10-f	5-f	10-f
2	232	97.8	97.8	96.5	97.2	98.0	98.5
3	120	98.3	98.5	96.3	97.0	97.4	97.6
4	18	86.8	87.5	91.0	94.4	93.0	95.8

For evaluations using image-like representations, we employed three methods in CNN: unweighted loss function, weighted loss function and fine-tuning. In the unweighted and weighted loss function evaluations, we split the test data from the dataset by considering the sample size of the classes; we assigned probabilities for each class considering the inverse proportion to its size and then randomly select test data with the associated probabilities. In fine-tuning evaluations, we performed a 10-fold validation at the last round where all the classes have the same sample size.

A. Text-like Data Evaluation

❶ **TM-1.** We trained and tested models with the user-specific dataset. As shown in Table I, the user-specific dataset has unbalanced sample size across classes. To mitigate bias, we use the same sample size for each class and change the number of classes at each step. The accuracy results are shown in Table IV. Due to the limited number of samples, the accuracy of 4-class classification is lower than 3- and 2-class classification. We observe a higher accuracy with k=10 than when k=5 in the k-fold cross-validation, perhaps due to the large training data capturing the population's distribution in the first case than in the latter. The results show 95.83% accuracy with MLP and 4-class classification. With 3-class classification, we obtained 98.51% accuracy with SVM. With 2-class classification, we obtained 98.49% accuracy with MLP.

Since the user-specific dataset is compiled from actual users, exhibiting mobility patterns, about 35% of the routes are overlapped. In a repetitive and overlapped setting, both training and testing splits may contain similar patterns leading to the high accuracy scores. The results prove that a targeted attack on a person whose activity history is known will be successful with accuracy between 86.80% and 98.51%.

❷ **TM-2.** While evaluating TM-2, the borough-level dataset is used. A model is created for each of the cities, Los Angeles, Miami, New Jersey, New York City, San Francisco, and Washington DC, by labeling the data as the name of the corresponding borough and evaluated separately. Figure 8 shows the accuracy, precision, recall and F1 scores of the each model in bar charts. All of the accuracy scores of the

TABLE V
TM-3 EVALUATION ON CITY-LEVEL DATASET. PREDICTION ACCURACY (**A**), RECALL (**R**), F1 SCORE (**F1**) WITH DIFFERENT CLASSIFICATION TECHNIQUES AND SAMPLE SIZE. **C** COLUMN STANDS FOR INDICATING THE NUMBER OF CLASSES IN THE CLASSIFICATION AND **S** COLUMN SHOWS SAMPLE SIZE OF EACH CLASS.

		SVM			RFC			MLP		
C	**S**	**A**	**R**	**F1**	**A**	**R**	**F1**	**A**	**R**	**F1**
3	743	80.0	69.8	70.2	79.1	68.4	68.4	80.9	71.2	71.6
5	362	90.7	77.7	78.4	89.4	74.8	76.0	90.5	77.4	78.4
7	266	90.7	66.7	66.5	89.0	61.1	61.0	90.0	64.3	64.4
8	155	91.9	68.6	68.5	88.9	57.0	60.3	90.9	65.1	64.5
10	82	93.9	70.2	70.4	92.4	58.1	57.5	92.9	63.7	63.3

TABLE VI
TM-3 EVALUATION ON CITY-LEVEL DATASET WHEN 35% OVERLAP IS INTRODUCED. PREDICTION ACCURACY (**A**), RECALL (**R**), F1 SCORE (**F1**) WITH DIFFERENT CONFIGURATIONS. **C** COLUMN STANDS FOR INDICATING THE NUMBER OF CLASSES IN THE CLASSIFICATION AND **S** COLUMN SHOWS SAMPLE SIZE OF EACH CLASS.

		SVM			RFC			MLP		
C	**S**	**A**	**R**	**F1**	**A**	**R**	**F1**	**A**	**R**	**F1**
3	966	91.7	82.7	82.8	89.0	77.8	79.1	92.4	84.0	84.1
5	470	94.6	81.6	81.2	93.7	78.7	78.4	95.6	85.0	84.7
7	338	93.6	72.1	72.5	92.4	68.4	68.8	93.9	73.4	73.4
8	202	94.7	75.4	74.9	93.2	67.8	66.9	94.6	74.9	74.2
10	107	94.4	71.4	72.5	93.6	67.7	66.9	93.6	68.9	69.8

models are above 55% while precision, recall and F1 scores are varying across each model. The main justification for this is that since there is no overlapped or repetitive routes among the mined segments in borough-level dataset, and the elevation differences and elevation sequences are not distinctive enough within a city to decide in which borough is the given test data is. This fact also clarifies the difference between the results of TM-1 and TM-2. The results of the simulated behaviour will be discussed in the simulations subsection.

❷ **TM-3.** In TM-3 evaluations, due to sample size differences across the labels in city-level dataset, we followed the same procedure in TM-1 evaluations. A fixed number of samples was randomly selected from each class for training and testing. Table V shows the results of the evaluation, where we employed 10-fold cross-validation and averaged results of the 10 folds. Per the reported results, we were able to predict the city of an elevation profile among 10 cities with an accuracy of 93.9%, among 8 cities with an accuracy of 91.93%, among 7 cities with an accuracy of 90.67%, among 5 cities with an accuracy of 90.71%, and among 3 cities with an accuracy of 80.86%. The success of the city-level estimations when compared to the borough-level estimations is based on the elevation range and sequence differences across cities, which is reasonable, even though the dataset is mined in a similar fashion as in the borough-level dataset. This mining indicates that city-level dataset also does not contain comprehensive, repetitive and overlapped samples. The results of the simulated evaluation will be discussed in the Simulations subsection.

1) Simulations: The mined datasets do not contain overlapped or duplicate samples as in the user-specific dataset. In this evaluation, we simulated overlapped mined datasets and performed evaluations under the same threat models.

Simulation of TM-2. For the city-level estimation evaluations, we rebuilt a simulation dataset with 30 - 34% overlap ratio for each region within the cities. The same evaluation procedures are then followed as the original mined dataset, which is 10-fold cross validation with fixed n-grams size of 8. Figure 9 shows comparison between the results of MLP classification, confirming our previous hypothesis that having overlapped route samples would increase the accuracy. Since the mined dataset is not specific to any target user's behaviour, it is anticipated to result in less accuracy than the TM-1 evaluation accuracy scores.

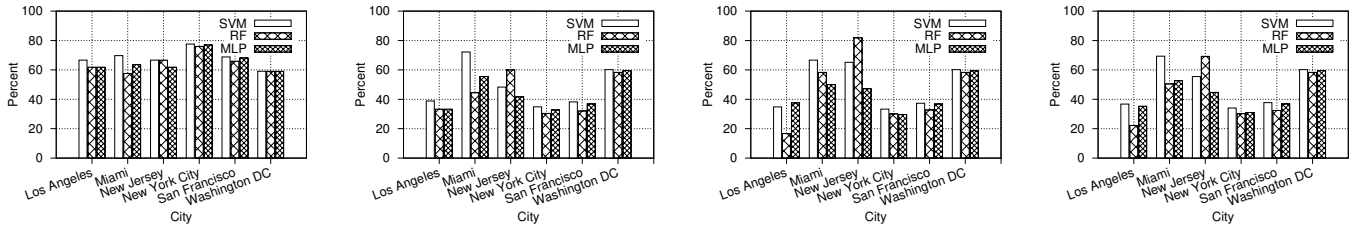
Simulation of TM-3. For TM-3's simulated evaluations, we rebuilt a simulation dataset with 35% overlap ratio for each city and performed the same evaluation with 10-fold cross validation and 8-grams. Table VI shows the results. By comparing Table V to Table VI, we notice that the accuracy, recall, precision, and F1 scores have increased for all classification techniques as expected. The improvements prove our previous hypothesis that having similar patterns in a dataset affects the success of the attack.

B. Image-like Data Evaluations

Dealing with Unbalanced Dataset. The original datasets are unbalanced and there are various methods to deal with unbalanced datasets, including downsampling, oversampling, and creating synthetic samples from existing ones. Among these methods, downsampling and oversampling are the easiest ones to explore, although downsampling leads to losing great amount of data and oversampling rises the chances of getting lower accuracy as the misclassified duplicated samples increases the false ratio.

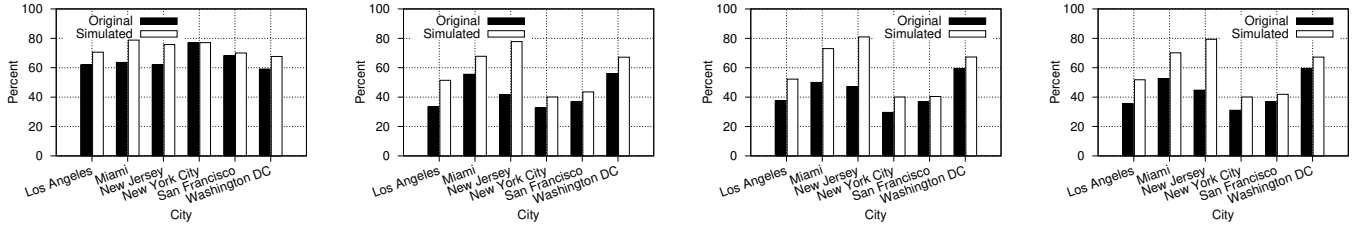
Weighted Loss Function. For the unbalanced dataset, we utilized a weighted loss function while training the CNN and used all the data in the dataset. By assigning a class weight that is inversely proportional to the sample size of the class, we signify samples of small classes while calculating the loss, thus their effect does not easily wear off.

Fine-Tuning with Different Samples. Fine-tuning is a common technique in deep learning, and is used for re-training a complex pretrained model with another dataset. To address the unbalanced dataset, we take advantage of fine-tuning in a different manner. Namely, we introduced rounds and created a set of small datasets from the unbalanced datasets for each round. As illustrated in Figure 10, several small and balanced datasets are created by randomly selecting samples. For each consecutive rounds, samples of one or more classes are discarded, and the round dataset is created from the remaining classes. After round dataset creation, the model is trained with the round dataset that contains *the least number of classes*, *i.e.,* the lattermost created round dataset. At each step, the model is re-trained using the same or different hyperparameters until all the rounds expire. The dataset ordering of the rounds are reversed, since the impact of the smallest dataset would wear off if the model is trained with the same order of round



(a) Accuracy (b) Recall (c) Precision (d) F1 score

Fig. 8. Accuracy, precision, recall and F1 score of TM-2 evaluation with different classification techniques.



(a) Accuracy (b) Recall (c) Precision (d) F1 score

Fig. 9. Comparison of the TM-2 simulated dataset evaluation accuracy scores by using Multi-Layer Perceptron classification and the original dataset evaluation accuracy scores by using Multi-Layer Perceptron classification.

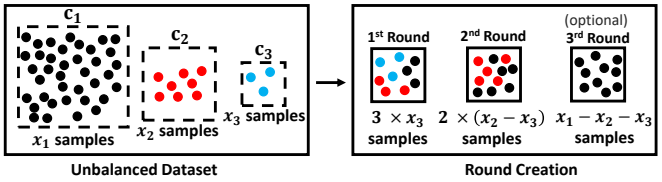


Fig. 10. An illustration of round creation from an unbalanced dataset of 3 classes.

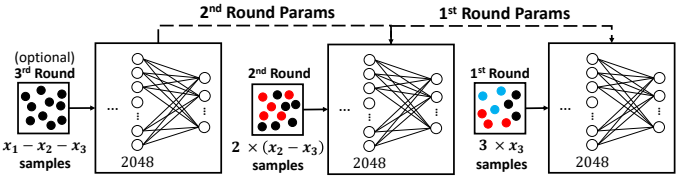


Fig. 11. An illustration of the fine-tuning pipeline for an unbalanced dataset of 3 classes.

TABLE VII
COMPARISON OF MAXIMUM ACHIEVED ACCURACY ACROSS DIFFERENT METHODS. THE UNWEIGHTED LOSS(UW) COLUMN IS NOT CONSIDERED WHILE DECIDING THE MAXIMUM ACCURACY, AS THE RESULTS ARE BIASED. THE MAXIMUM ACCURACY OF EACH EVALUATION IS WRITTEN **BOLD**, THE RESULTS THAT ARE NOT CONSIDERED ARE WRITTEN *italic*. LOCATION ABBREVIATIONS ARE IN TABLE III.

Methods	Text-like	Image-like		
	DS	UWL *(biased)*	WL	FT
TM-1	**95.83**	*96.98*	95.23	87.93
TM-2: LA	65.13	*68.85*	**68.39**	63.63
TM-2: MIA	68.65	*88.96*	**86.80**	62.50
TM-2: NJ	63.52	*93.45*	**79.42**	57.14
TM-2: NYC	78.85	*74.20*	**79.37**	72.79
TM-2: SF	64.52	*67.20*	**78.70**	65.38
TM-2: WDC	60.79	*62.79*	70.28	**71.50**
TM-3	**93.90**	*92.51*	92.82	89.00

dataset creation, which conflicts with the whole idea. As illustrated in Figure 11, while re-training, the parameters of the previous model are passed to the model of the next round. The hyperparameters of each round can be tuned accordingly. For instance, for the last round, where we include all of the classes, the learning rate can be reduced to find the loss minima.

To evaluate our attacks on the image-like data, the elevation profiles are converted into a dataset of images and rounds using the configurations and steps discussed above. Table VII highlights the maximum achieved prediction accuracy along with comparisons among methods.

Weighted vs. Unweighted Loss Function. To observe the impact of the weighted loss function, we conducted evaluations without giving any weight to the classes in the loss function while using an unbalanced dataset. We note that the unweighted loss function evaluation results are biased due to the unbalanced dataset. Table VII shows the maximum achieved accuracy for each dataset and method. Even though the weighted loss function evaluation results are biased, which *seems* successful in outputting the largest class used during training and testing, the biased results remain behind 4 evaluations out of 8. For the remaining evaluations, and *except* TM-2*: NJ*, the difference between the results is insignificant. When the biased results are excluded for TM-2 evaluations, the weighted loss function performed better than text-like and fine-tuning methods. However, in TM-1 and TM-3, the accuracy scores between methods are considerably close. Thus, we conclude the weighted loss function improved the prediction

TABLE VIII
THE FINE-TUNING RESULTS FOR TM-1 AND TM-3 AS THE EPOCH SIZE CHANGES.

	TM-1			TM-3		
Epoch Size	**500**	**1000**	**2000**	**500**	**1000**	**2000**
Accuracy	79.3	87.9	82.7	86.0	89.0	87.8
Recall	55.8	67.5	63.1	29.7	45.3	38.9
Specificity	86.3	92.6	88.4	92.2	93.9	93.2
F1 Score	58.6	68.2	63.3	36.2	45.4	41.1

TABLE IX
THE FINE-TUNING RESULTS FOR TM-2 AS THE EPOCH SIZE IS 1000 AND LEARNING RATE IS 0.001 FOR ALL ROUNDS.

	LA	**MIA**	**NJ**	**NYC**	**SF**	**WDC**
Accuracy	63.6	62.5	57.1	72.8	65.4	71.5
Recall	28.0	25.6	40.0	18.1	30.7	73.2
Specificity	75.8	75.9	66.7	83.4	76.3	73.2
F1 Score	28.8	28.6	37.5	18.4	31.4	73.4

performance primarily for TM-2.

Fine-tuning. Round datasets are created from the original data. For TM-1, with 4 classes, 3 rounds are created. For TM-3, with 10 classes, 5 rounds were created by eliminating 1, 2, 1 and 2 classes at each round, respectively. The dataset of TM-2 can be considered as a compilation of the dataset of 6 cities: Los Angeles (3 rounds), Miami (3 rounds), New Jersey (2 rounds), New York City (4 rounds), San Francisco (2 rounds), and Washington DC (1 round). Even though the main idea is using all the data we have, we decided to downsample the classes with large sample size. For instance, in the evaluation of TM-2: New York City, the biggest class has 5,455 samples where the second biggest class has 960 samples. In such cases, we did not create additional round for only one class as this round would have strong influence over the predictions, *i.e.,* may cause overfitting.

Table VII shows the fine-tuning method outperformed other methods only for TM-2: Washington DC. The difference between the fine-tuning evaluation of Washington DC and others is that we were able to create only one round from the data in the former. Overall, according to the results shown in Table VIII and Table IX, the fine-tuning evaluation is not as successful as the weighted loss function evaluation, since we still lost some data while creating rounds.

V. RELATED WORK

In this work, we addressed *location privacy* in activity trackers using side channel information from publicly shared elevation profile, a topic that is related to various pieces of the literature. In the following, we review some of those studies.

Most location privacy breaches are caused since users do not know why or how to preserve location privacy. [11] developed a tool to examine possible privacy exposures of users in their social networks where the data is mostly collected from wearable devices. Using this tool, the authors aimed to enhance the awareness of information leakage in social networks, particularly fitness apps in which the data retrieved from wearable devices is shared on social networks. [12] aimed to increase awareness of location privacy on geo-social networks by surveying 186 users, where 77% of them indicated they use location-based services often, several times a day, and 47% of them reported that they were not aware that the location-based apps collect and store location information when users select the private location option. Moreover, 43% of respondents were not aware that application may share the location information with third parties.

Despite the methods employed to preserve location privacy, several attacks are devised to uncover supposedly protected locations. Experiments for revealing exact locations from trajectories with private zones are conducted on Strava [13]. Researchers found the exact end points associated with users, even when such users selected the private zone option when sharing the training route. In another study, location trajectories of users are recovered from publicly available aggregated mobility data obtained from GSM operators [14]. The attack relies on tracking the regularity—i.e., coming across the same location trace in the aggregated data regularly—and uniqueness—i.e., the location trace belongs to a unique user—of the user mobility traces to recover trajectories.

As our study exemplifies, online social networks lays under the scope of privacy breach risks for users. [15] shows that sharing data which reveals spatiotemporal features of users' mobility patterns on online social networks reveal sensitive information such as home location, using a different form of data, *i.e.,* multimedia. [16] shows that location-based social networks are vulnerable to identity privacy breaches by revealing the identity of users by observing their mobility patterns.

Several attacks against general location privacy methods are proposed [17]. The homogeneity attack [18] is an attack on k-anonymity to infer data of interest from other shared data. [18] illustrated a scenario where an adversary infers the illness of a target person from available information, the zip code, age, etc. The same method can be applied to infer location data. In location distribution attacks [19], the adversary exploits the fact that users are mostly not uniformly distributed in the location space. Another attack [20] utilized the aggregated traffic statistics and environmental context information. The attack scenario includes an adversary who tries to reveal the possible location of the target by making use of the fact that the probability of target's whereabouts is not uniformly distributed. Map matching methods [21] aim to restrict the obfuscated area to a smaller but plausible area by removing irrelevant areas. Movement boundary attacks were explored [22], where the adversary aims to calculate the movement boundary of a target by chasing the position queries and updates of the target. After calculating the boundary, the location of interest, such as home or work place, is inferred and the irrelevant locations are discarded.

Although we did not directly touch upon preserving the location privacy in our study, there has been a few related studies in this space. The fast-growing need of preserving location privacy over the aforementioned attacks excited researchers'

attention. Researchers introduce obfuscation methods such as decreasing the quality of the location by introducing inaccuracy and imprecision [23]. Additionally, the term k-anonymity is defined as obscuring the location information of individuals with k number of other individuals within the region [24], [25].

VI. Conclusion

In this paper, we presented new attacks on location privacy using only elevation profiles. The attacks are categorized into three types: predicting location by knowing the activity history of the target, predicting the borough by knowing the city of the target, and predicting the city of the target without any prior knowledge. The key contributions of our work are proving the concept that hiding the route of a workout and sharing only the elevation profile is not sufficient to preserve location privacy, defining a new attack surface by creating scenarios for possible threat models, and providing a machine-learning approach to realize such threat as attacks. To validate our attacks we created three datasets by collecting data from athletes, and mining data from a popular fitness tracking website and Google Elevation API. We preprocessed the datasets by employing Natural Language Processing and Computer Vision approaches, and then employed classification techniques to predict the location from elevation profiles. En route, we defined three threat models and evaluated each of them individually on the different datasets. As a result of the evaluations, we were able to identify the corresponding location of an elevation profile with accuracy between 59.59% and 95.83%. In the future, we will explore compatible defenses such as devising and using route statistics that serves the same purpose as sharing elevation profile; demonstrating the roughness of the route, while preserving users' privacy.

Acknowledgement. This work was supported by NSF under grant CNS-1809000, CyberFlorida Collaborative Seed Award, CyberFlorida Capacity Building Award, and NRF under grant 2016K1A1A2912757. We thank the reviewers of ICDCS 2020 and ICWSM 2020 for their valuable comments that significantly improved this work.

References

[1] J. P Higgins, "Smartphone applications for patients' health & fitness," *The American journal of medicine*, vol. 129, 06 2015.

[2] I. Polakis, G. Argyros, T. Petsios, S. Sivakorn, and A. D. Keromytis, "Where's wally?: Precise user discovery attacks in location proximity services," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: ACM, 2015, pp. 817–828. [Online]. Available: http://doi.acm.org/10.1145/2810103.2813605

[3] G. Friedland and R. Sommer, "Cybercasing the joint: On the privacy implications of geo-tagging," 08 2010, pp. 1–8.

[4] "Map privacy," https://support.strava.com/hc/en-us/community/posts/208125208-Map-Privacy, accessed: 2020-01-11.

[5] "How do i remove display map from my activity list?" https://support.strava.com/hc/en-us/community/posts/360041573811-How-Do-I-remove-, accessed: 2020-01-11.

[6] "Hide map," https://support.strava.com/hc/en-us/community/posts/360039162451-Hide-Map, accessed: 2020-01-11.

[7] "Privacy setting to hide activity map from non-followers," https://support.strava.com/hc/en-us/community/posts/208848117-Privacy-setting-to-, accessed: 2020-01-11.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[9] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine Learning*, vol. 40, no. 3, pp. 203–228, Sep 2000. [Online]. Available: https://doi.org/10.1023/A:1007608224229

[10] P. W. Eklund, "A performance survey of public domain supervised machine learning algorithms," Tech. Rep., 2002.

[11] A. Aktypi, J. Nurse, and M. Goldsmith, "Unwinding ariadne's identity thread: Privacy risks with fitness trackers and online social networks." Association for Computing Machinery, 2017, pp. 1–11.

[12] A. I. Abdelmoty and F. Alrayes, "Towards understanding location privacy awareness on geo-social networks," *ISPRS International Journal of Geo-Information*, vol. 6, no. 4, 2017. [Online]. Available: http://www.mdpi.com/2220-9964/6/4/109

[13] W. U. Hassan, S. Hussain, and A. Bates, "Analysis of privacy protections in fitness tracking social networks -or- you can run, but can you hide?" in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, 2018, pp. 497–512. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/hassan

[14] Z. Tu, F. Xu, Y. Li, P. Zhang, and D. Jin, "A new privacy breach: User trajectory recovery from aggregated mobility data," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1446–1459, June 2018.

[15] D. Zheng, T. Hu, Q. You, H. A. Kautz, and J. Luo, "Towards lifestyle understanding: Predicting home and vacation locations from user's online photo collections," in *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, 2015, pp. 553–561. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10487

[16] L. Rossi, M. J. Williams, C. Stich, and M. Musolesi, "Privacy and the city: User identification and location semantics in location-based social networks," in *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, 2015, pp. 387–396. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10498

[17] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, Jan 2014. [Online]. Available: https://doi.org/10.1007/s00779-012-0633-z

[18] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," in *22nd International Conference on Data Engineering (ICDE'06)*, April 2006, pp. 24–24.

[19] M. F. Mokbel, "Privacy in location-based services: State-of-the-art and research directions," in *2007 International Conference on Mobile Data Management*, May 2007, pp. 228–228.

[20] R. Shokri, G. Theodorakopoulos, J. Le Boudec, and J. Hubaux, "Quantifying location privacy," in *2011 IEEE Symposium on Security and Privacy*, May 2011, pp. 247–262.

[21] J. Krumm, "Inference attacks on location tracks," in *Pervasive Computing*, A. LaMarca, M. Langheinrich, and K. N. Truong, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 127–143.

[22] G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino, "Preventing velocity-based linkage attacks in location-aware applications," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '09. New York, NY, USA: ACM, 2009, pp. 246–255. [Online]. Available: http://doi.acm.org/10.1145/1653771.1653807

[23] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proceedings of the Third International Conference on Pervasive Computing*, ser. PERVASIVE'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 152–170. [Online]. Available: http://dx.doi.org/10.1007/11428572_10

[24] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 571–588, Oct. 2002. [Online]. Available: http://dx.doi.org/10.1142/S021848850200165X

[25] A. Gkoulalas-Divanis, P. Kalnis, and V. S. Verykios, "Providing k-anonymity in location based services," *SIGKDD Explor. Newsl.*, vol. 12, no. 1, pp. 3–10, Nov. 2010. [Online]. Available: http://doi.acm.org/10.1145/1882471.1882473