Learning Location From Shared Elevation Profiles in Fitness Apps: A Privacy Perspective

Ulku Meteriz-Yildiran, Senior Member, IEEE, Necip Fazil Yildiran, Joongheon Kim, Senior Member, IEEE, and David Mohaisen, Senior Member, IEEE

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 the 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 the 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 99.80%. The findings are alarming, highlighting that sharing elevation information may have significant location privacy risks.

Index Terms—Location privacy, privacy breach, privacy in social media, fitness applications, natural language processing, applied machine learning

1 INTRODUCTION

From smartphones to wearables, an increasing number 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 using geodata, such as location and elevation information. More recently, fitness applications that run on smartphones and smartwatches used these components to collect spatial, temporal, and activity-specific information to analyze, summarize, and visualize users’ activities. By analyzing each activity, many of those applications 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 an activity (e.g., walking, running, climbing, cycling), users can have positive behavioral changes through a more active lifestyle motivated by competitions with acquaintances [2].

Despite the broad set of advantages that geodata offers, geodata usage and uncontrolled sharing can pose a significant privacy risk that can be further exploited in multiple attacks, including stalking [3] and cybercasing [4]. For example, with a large amount of geotagged data, including text, images, and videos, cybercasing provides criminals and maliciously motivated individuals with a significant attack vector. 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, this data can also 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, such as 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. The
results of the survey, summarized in Fig. 1, reveal that 51% of the participants start their training from their homes, 36% start from their school, and 3% start from their workplace, while 76% of the participants finish their training at their homes. Moreover, for the same set of users (results are not shown in Fig. 1), 42% of those users have indicated that not sharing location information implies privacy protection, while 30% of the respondent were uncertain, and 28% were certain that not sharing would not necessarily mean their privacy is protected. The 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 to hide the location trajectory by removing the activity map in the fitness applications, users still want to share elevation profiles 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 the elevation profile of an activity while masking the map that highlights the actual trajectory, which is deemed of high privacy value to them [5], [6], [7], [8].

In the same survey we conducted earlier, we asked our 60 subjects “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%), and only 17 indicating “no”.

Is sharing the elevation profile of activity enough to maintain the privacy of users? In this paper, we argue that an approximate location, extracted from the contexts of activities and at different levels of location granularity, could still be revealed from the elevation profile information. We examine this problem comprehensively and develop techniques 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 investigate the possible attack surface for the problem by exploring three different threat models, which we later use to evaluate the success of our approaches by simulating our methods considering each threat model,
- we demonstrate that location information can be predicted from elevation profile using different machine/deep learning methods with accuracy in the range 80.25% - 99.80% at different resolutions.
- In this section, we provide some background information highlighting the significance of elevation profiles for athletes, the use cases, some properties of the fitness applications on the market today, and some reported privacy breach incidences of fitness applications to contextualize further the work presented in the rest of this paper.

2 BACKGROUND

In this section, we provide some background information highlighting the significance of elevation profiles for athletes, the use cases, some properties of the fitness applications on the market today, and some reported privacy breach incidences of fitness applications to contextualize further the work presented in the rest of this paper.

2.1 Elevation Profiles Importance for Athletes

Athletes who keep track of their activity records measure various modalities and attributes associated with the activities, including the distance, speed, overall time, and heart rate over the course of the activity. Based on these attributes, they adjust their training strategies to reach their goals. Elevation changes, often reported in the form of elevation gain, are one of the most significant attributes measuring the performance of a cyclist/runner and often depict how hard the run or ride is. For example, riding a bike for a 20-mile ride while climbing 1000 feet in total is significantly more challenging than biking on a flat terrain [9]. Therefore, when recording or sharing a ride/run, athletes care about the changes in the elevation, thus elevation profiles.

2.2 Fitness Applications & Privacy Breach Incidents

Fitness applications allow users to track their workout history and provide them with statistics. Moreover, some
Users of fitness applications are greater than in privacy zones. In Runtastic, one of the popular fitness applications, 95% of the users are at risk of being located. The same study also reveals the exact start and end point of a route that utilizes the privacy zone feature. The combined shared activity, including bicycles, tracking devices, shoes, etc., are shown on the leaderboard grouped by gender and age. Consequently, once this information makes them a target for robbery, and several incidents of bicycle theft are reported. The aggregated and public activities of Strava users over the past year. It is, however, shown that a dedicated adversary can deanonymize heatmap to find out users who ran in a hypothetical setting. We note, however, that those settings are also plausible if such sharing is enabled.

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 a more restrictive (powerful) model.

In TM – 1, we assume an adversary with workout history records of a target user, and the goal of the adversary is to identify the last workout location of the target user from the recently 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, where 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 previous fitness activity records, 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 location privacy.

In TM – 2, we assume an adversary with access to limited information, such as the city where 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 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

3 Threat Models

We outline the potential threat models under which this study is conducted. We describe three models under which location privacy is breached only from associated elevation profiles. 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.

Table 1: Popular Fitness Applications and Their Features

| Service    | ET | SS | SNS | PR | BU |
|------------|----|----|-----|----|----|
| Strava     |    |    |     |    |    |
| Runtastic  | ∗  |    |     |    |    |
| Runkeeper  |    |    |     |    |    |
| Nike+ Running |   |    |     |    |    |
| MapMyRun   |    |    |     |    |    |

ET: Exercise tracking, SS: Ability to share to social media. SNS: Social networking capabilities in the service. PR: Private records. BU: User blocking capability.

fitness applications have social network capabilities, as shown in Table 1, and allow users to share workout summaries that are known to motivate users and their social network connections to achieve their goals [2]. Some fitness applications also inherit user-blocking features and capabilities from social network platforms, including user privacy options such as private records—the activity records that are only visible to the user.

Although fitness applications have configurable privacy options, there have been a lot of privacy incidents concerning location data obtained from those fitness applications. We review some of those privacy breaches in the following to contextualize our work in the broader privacy literature.

**Revealing Secret U.S. Military Bases.** Strava, which is one of the most popular fitness tracking applications in the market today, collects users’ public data and publishes a heatmap of the aggregates to highlight routes frequented by users [10]. Although the aggregates in the heatmap do not explicitly contain any identity information, activities in desolate places revealed the location of many U.S. military bases, which is considered sensitive information [11], [12].

**Deanonymization Through Strava Segments.** In Strava, the heatmap feature was used to show “heat” made by the aggregated and public activities of Strava users over the past year. It is, however, shown that a dedicated adversary can deanonymize heatmap to find out users who ran in a specified route [13]. For example, by selecting a route from the heatmap, a registered user can manually create a GPS eXchange (GPX) track file and create a segment using it on Strava. A segment is a portion of a road or a trail where athletes compare their finishing times. Consequently, once this segment is created, the users who previously ran that route are shown on the leaderboard grouped by gender and age. This feature is then leveraged to identify individuals who ran that particular place.

**Tracking and Bicycle Theft.** Users of fitness applications can share information related to the equipment used for the activity, including bicycles, tracking devices, shoes, etc., along with the routes frequented. The combined shared information makes them a target for robbery, and several such incidents of bicycle theft are reported [14], [15], [16], [17].

**Attack on Privacy Zone.** To cope with the increasing privacy risks, Strava features privacy zones, a technique to obfuscate the exact start and end points of a route. A recent study [18] has demonstrated that it is possible to reveal the exact start and end point of a route that utilizes the privacy zone feature. The same study also claimed that around 95% of the users are at risk of revealing their location information.

**Live Activity Breach.** In Runtastic, one of the popular activity-tracking applications, users can share their live activities. In theory, users should be able to configure the privacy settings for their activities such that only privileged users, such as connections on the application platform, can track the shared live activity session. However, it has been demonstrated [19] that the selected privacy settings are not correctly applied to a live session. As a result, everyone can go through live sessions and track Runtastic users in real time, even though the associated privacy options should have prevented this type of breach. Based on this incident, it would be easy to stalk and locate a user, e.g., a lone runner or cyclist with expensive equipment, in real time.
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 through fitness applications. It offers dense and thorough coverage of regions frequented by the user; those regions are used as class labels. The city-level 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.

### 5.1 Data Collection

The implementation details of data collection, preprocessing, feature extraction, and multi-class classification are addressed in the following subsections.
5.1.1 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 through fitness applications. First, the location trajectories included in the user-specific dataset are converted to GPX format to avoid confusion caused by different formats and settings across the activity records. 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 minimum coordinates of the trajectory as illustrated in Fig. 4. 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. Then, we annotated the region labels, such as Orlando, Washington DC etc., based on the manual observation on the map. If no region includes the trajectory, a new region is created. The final sample size distribution of the user-specific dataset is shown in Table 2.

The user-specific dataset is prone to have 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%.

5.1.2 City-Level Dataset

For the city-level dataset, we mined publicly available training route segments in a popular fitness tracking application using its 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 information, we are also not obtaining any information beyond what is provided by the users explicitly. We also note that the training route segments are user-created activity routes whose main purpose is to compare completion times among users who also completed the same route.

They are particularly useful for our purposes since they include public location trajectories that are frequented by the actual users rather than randomly created location trajectories that may not necessarily be of privacy value. During the segments mining, the anonymity—thus the privacy—of the users who frequented the segments or created the segments is maintained.

The overall data mining procedure consists of three steps, as illustrated in Fig. 3. First, we define the cities of interest, which we also use as the class labels per our threat model. For each city, we define the rectangle geolocation boundary box \( B \) consisting of the top right and bottom left corner coordinates in the boundary selection phase. In the segmentation phase, and since EXPLORESEGMENTS() returns only the ten most frequented segments encapsulated by a given boundary, and to obtain more data of a geolocation boundary box, we divide the large rectangle boundary of the city into smaller region boundaries, each denoted by \( b_i \), by following a grid-like structure as shown in the second phase of the Fig. 3. For each region boundary \( b_i \), we call EXPLORESEGMENTS() and receive the geolocation polyline path, \( path_i^j \) where \( j \in [1, 10] \), of the 10 most frequented segments encapsulated in \( b_i \), as shown in the segment exploration phase. Finally, since the polyline paths do not include elevation profiles, we obtain the associated elevation profile \( ele_{i,j} \) for each \( path_i^j \) using the Google Maps Elevation API through the elevation retrieval phase. The sample size distribution of the city-level dataset can be found in Table 3.

Unlike the user-specific dataset, the city-level dataset does not include any overlapped samples since each region \( r_i \) is disjoint with the other regions. A segment route included by more than one neighboring region is not...
considered since \texttt{ExploreSegments()} returns the routes encapsulated within the given boundaries, $b_i$.

### 5.1.3 Borough-Level Dataset

For the borough-level dataset, we apply a similar mining procedure as we have done with the city-level dataset, using the borough boundaries instead of the city boundaries. Table 4 shows the sample size distribution of the borough-level dataset for different cities.

### 5.2 Preprocessing

A key design element in our pipeline is the representation modality of the elevation profile, which will significantly impact the performance of our elevation-location mapping, as we show later. We transform the samples into text-like and image-like representations to facilitate feature extraction and feed them into our classification module. In this section, we describe the details of the utilized preprocessing methods.

#### 5.2.1 Text-Like Representation

For our text-like representation, our approach follows four steps, as shown in Fig. 5: discretization, word size decision, text encoding, and vocabulary creation.

**Discretization.** 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 longer string correspondences and, consequently, longer vocabulary and sparse feature vectors. The discretization is done as follows. Let $e_{ji}^t$ be the $i$th elevation value in $j$-th sample. The discretization functions are defined as $f(e_{ji}^t) = \lfloor e_{ji}^t \rfloor$ and $f(e_{ji}^t) = \lfloor e_{ji}^t / 10^3 \rfloor$, 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. Having more fine-level details when the samples are sparse would help our models to learn more and thus increase the accuracy. To demonstrate the effect of discretization, we measured the vocabulary size of the smallest class of the User-specific dataset, i.e., San Diego. For a class as small as the San Diego class, using the second function results in a vocabulary of size 12,870, and using the first function results in a vocabulary of size 3,155. Such difference in the vocabulary size demonstrates the effect and necessity of discretization.

**Word size decision.** For the word size decision, we use $w = \log_c l$, 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.

**Text encoding.** 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.

**Vocabulary creation.** 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 Fig. 6, a window with size $W = w \times n$ is slid 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 construct the vocabulary consisting of unique $n$-grams of the given dataset after traversing the corpus by $n$ times with different window sizes.

### 5.2.2 Image-Like Representation

In the image-like representation, the elevation signals are drawn as line graphs and saved as a $32 \times 32$ images. To draw a line graph, the maximum and minimum values for the y-axis are set to be the maximum and minimum of each elevation signal, and the lines are colored to encode the value interval in which the elevation signal ranges. We note that the image-like representation with colors has multiple advantages over the black-white representation where the y-axis is set to the range of a whole dataset. First, as illustrated in Fig. 7, the alterations of an elevation signal are more visible with the color encoding method, which could be a more discriminative feature to learn. Second, the color encoding method results in efficient utilization of the feature space. We use 200 elevation values for each image, obtained by dividing the elevation signal into equal-sized parts.

### 5.3 Feature Extraction

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

#### Text-Like

In the text-like feature extraction, we utilized two methods: (i) n-grams, (ii) tf-idf.

For the n-grams method, words and non-overlapping occurrences of word sequences are counted, and 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 occurrence probability of each word in the given sample.

The tf-idf is a statistical feature signaling the importance of a word in a document. The tf-idf values proportionally increase as the number of appearances of a word in a document increases. Technically, the tf-idf for a word is the multiplication of two metrics: (i) term frequency (tf) and (ii) inverse document frequency (idf). Tf-idf of a word $t$ in a document $d$ included in the set of documents $D$ of which

$$
tf-idf(t, d, D) = tf(t, d) \cdot idf(t, D),
$$

$$
tf(t, d) = \log(1 + freq(t, d)),
$$

$$
idf(t, D) = \log \left( \frac{N}{\text{count}(d \in D : t \in d)} \right).
$$

The higher the tf-idf means that the word $t$ is more relevant to the particular document $d$.

#### Feature Selection

When the dataset is large and diverse, the vocabulary and, consequently, the feature vector representation become too large to process, compute, and learn from. With a feature selection phase to address long feature vectors, some rarely occurring features in the vocabulary are discarded according to a pre-specified feature frequency threshold. For selection, the features are ordered by their term frequency across the corpus, the features whose term frequency is below a specified threshold are discarded, and a new vocabulary is created. For both feature extraction methods of the text-like representations, the term frequency threshold is set such that the size of the eventual vector representation is 5,000.

Image-like We use the CNN for processing and classifying the $32 \times 32$ images, thus 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 for the image-like representation is discussed in the context of the classification phase.

### 5.4 Multi-Class Classification

For classification, SVM, RF, MLP, LSTM, and CNN are used. SVM SVM is a supervised classification technique. The main challenge in SVM is finding the best hyperplane that divides the classes from each other considering a
given margin. SVM with the linear kernel is able to distinguish the classes more successfully when the features are multi-dimensional and numerous [20], [21]. In linear kernel settings, when an optimal hyperplane is found while representing a class, only the features around the hyperplane within the given margin are considered and the other features are simply ignored. As such, the complexity of SVM is independent of the number of features. Since we consider n-grams up to n=5 in the feature extraction phase, the number of features is numerous which makes the usage of SVM legitimate in terms of efficiency and success.

SVM with a linear kernel is generally used for binary classification. To utilize it for the multi-class classification problem, we use the one-versus-rest method. In this method, an individual model is trained for each class and the label of the most confident model is outputted. For the penalization, we use the L2 norm, as it is a standard for linear SVMs. As for the loss function, we utilized the square of hinge loss, and the hyperparameters are decided through grid search tuning.

RF. RF is based on decision trees, which are ensemble learning methods for classification. The main feature of ensemble methods is further improving the generality and robustness of a single estimator by combining several base estimators that are built with a given learning algorithm. In decision trees, features are represented by tree nodes and each branch between two nodes represents what the immediate ancestor node returned. Since building an optimal binary decision tree from given features is an NP-complete problem, using a Random Decision Forest with different tree configurations and efficient heuristics is a way to alleviate the NP-completeness for classification problems. While creating our random forest, perturb-and-combine techniques are used. Perturb-and-combine techniques are designed specifically for decision trees to improve their accuracy by creating several (different) versions of the estimator by perturbing the training set, then combining these different versions into a single estimator [22]. Further details on this approach can be found in [23] and [22].

In this study, we use 20 decision trees for the RF model. The final prediction is then done by averaging the tree predictions. We do not set any upper limit on the number of leaf nodes or the depth of the tree, i.e., there were no time optimization concerns during training the process, so we leave the trees to grow to their maximum depth.

MLP. MLP is a feed-forward fully-connected neural network that utilizes backpropagation for training and is used for supervised learning.

Recent studies on comparing multi-layer neural networks and decision trees [24], [25] concluded the following:

- Multi-layer neural networks allow incremental learning, in which the model’s knowledge is continuously extended, more easily than the decision trees.
- The training time of the multi-layer neural networks is much longer than decision trees.
- The multi-layer neural network predictions are generally as good as the predictions produced by the decision trees, although they can perform better in certain cases.

Given the aforementioned potential for MLP to perform better than RF, we used MLP with 20 hidden layers in our experiments. We used Adam solver [26] for weight optimization since it is shown to perform well in terms of both training time and validation score for large feature spaces. We also utilize ReLU as the activation function, 0.001 as the learning rate, and 200 as the epoch size. For regularization, we use L2 norm with 0.0001 penalty parameter. The hyperparameters are decided through grid search tuning.

LSTM. LSTM is a recurrent neural network architecture. Unlike the standard neural networks, LSTMs are capable of keeping track of long-term dependencies in the input sequences using feedback connections. Such long dependencies are handled through feature extraction in n-grams and tf-idf vectors for standard neural networks. LSTM, on the other hand, handles dependencies internally, without requiring an explicit feature extraction. LSTM is also particularly useful for capturing the order dependence in sequence prediction problems. For LSTM, the input sequence can be a time series, a sentence from a given language, or a text-like representation as in our application case. Fig. 8 depicts the LSTM architecture employed in this work, where the input is directly passed through the LSTM layer with four hidden layers, and two fully-connected layers following the LSTM layer. Each input sample (vector representation of an elevation profile for n-grams and tf-idf, or individual values for the raw data) is passed through the LSTM unit. When all sample vectors (or elevation values) are passed through the LSTM units, the final hidden state vector is passed to the subsequent fully-connected layer. In the LSTM unit, hyperbolic tangent and sigmoid (depicted as T and S, respectively, in Fig. 8) are used as the activation function. For the fully-connected layers, ReLU and softmax activation functions are used, respectively.

CNN. CNN is similar to neural networks in the mechanism, as both of them consist of neurons with learned parameters, weights, and biases. The improvement of CNN, however, is in the form of convolution layers, which apply forward passes that decrease the number of parameters of the neural network considerably. Convolution layers also facilitate the processing of high-dimensional data, such as images. They prepare high-dimensional data for a fully-connected layer, which cannot process high-dimensional data efficiently, by highlighting the important spatial features along the way.
First, we performed evaluations on the raw data. As
Second, we performed evaluations with text-

We trained and tested models with the
We performed experiments using 10-fold cross-validation and by fixing the dimension of n-grams to 5 for all datasets and associated threat models. With tf-idf features, we performed 10-fold cross-validation and fixed the dimension of n-grams to 5 for all datasets and threat models.

The user-specific dataset contains overlapped and repetitive portions by nature. In the Simulations subsection, we simulated the same behavior on the mined datasets and performed the same evaluations for comparison.

In this study, we utilize two CNN architectures, differing in the convolution layers dimensions. Fig. 9 illustrates the employed CNN architectures.

In the first architecture, we use two consecutive 2D convolution layers (CONV2D) 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 determined as 5, 1, and 2, respectively, based on the performance. The distinctive features are selected at the max-pooling layers with a kernel and a stride size of 2, which reduce the dimensions from $(32 \times 32)$ to $(8 \times 8)$ at two passes.

In the second architecture, we used two consecutive 1D convolution layers (CONV1D) along with the ReLU activation function. A dropout (DROP) layer is added to alleviate the overfitting problem. Then, a max-pooling layer (MAXPOOL) and a fully connected layer (FCON) are added.

For both architectures, the softmax function is used as an activation function at the output layer and the Categorical Cross Entropy is used as a loss function. For parameters optimization, we used the Adam optimizer.

6 Evaluation Results

We performed experiments for each dataset, data representation, and threat model. We categorize the evaluations into three: Raw, Text-like, and Image-like.

Raw. First, we performed evaluations on the raw data. As the multi-class classification models require a fixed input shape and an arbitrary elevation profile does not have a specific length, we divided the elevation profiles into equal-length (32) chunks and use the raw data to train and test the models. For all datasets and threat models, we used a slightly modified version of the soft voting ensemble method while testing with raw data. In conventional ensemble learning techniques, the input is passed to different models and the final prediction is assigned based on the decisions of the models. However, in this study, we passed the equal-length chunks of a single input, i.e., elevation profile, to a single model, and then assigned the final prediction based on the decisions on the input chunks.

Soft voting sums up the predicted probabilities for each class label for each input chunk and returns the class label with the highest probability as the final prediction.

Text-like. Second, we performed evaluations with text-like features: n-gram, and tf-idf. With n-gram features, we performed experiments using 10-fold cross-validation and by fixing the dimension of n-grams to 5 for all datasets and associated threat models. With tf-idf features, we performed 10-fold cross-validation and fixed the dimension of n-grams to 5 for all datasets and threat models.

The user-specific dataset contains overlapped and repetitive portions by nature. In the Simulations subsection, we simulated the same behavior on the mined datasets and performed the same evaluations for comparison.

Image-like. For the experiments on the 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 selected test data with the associated probabilities. In fine-tuning evaluations, we performed 10-fold cross-validation in the last round where all the classes have the same sample size.

6.1 Raw and Text-like Data Evaluation

6.1.1 Direct Evaluation

Evaluating TM = 1. We trained and tested models with the user-specific dataset. As shown in Table 2, the user-specific dataset has an 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 evaluation results are shown in Table 5. Due to the limited number of samples, the accuracy decreases as the number of classes increases. The only exception is C1D with n-grams and tf-idf. One-dimensional convolutions were able to capture the characteristics of elevation profiles even with a limited number of samples. The results show 99.80% accuracy with C1D, n-grams, and 4-class classification. With tf-idf and C1D, we obtained 99.00% and 99.94% accuracy with 3-class and 2-class classification, respectively.

LSTM gives better accuracy with raw data compared to the other text-like representations. LSTM performs better on the data where the ordering is decisive. With the text-like representations, the original ordering of the values is encoded separately, thus LSTM could not extract much information through the ordering.

For TM = 1, RF also performs better with raw data. Since extracting features from the range and the ordering of the values are less demanding for decision trees, it is reasonable to observe such a pattern.

Other classification methods, i.e., SVM, MLP, and C1D, benefit more from the n-grams and tf-idf features.

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 high accuracy scores. The results prove that a targeted attack on a person whose activity history is known will be successful with accuracy as high as 99.80%.
Evaluating TM – 2. While evaluating TM – 2, the borough-level dataset is used. Individual models are created for each of the cities, by labeling the data as the name of the corresponding borough and evaluated separately. Similar to the user-specific dataset, the borough-level dataset also has an unbalanced sample size across the classes. To avoid biased results, we fix the sample size to that of the smallest class for all classes. At each fold, we randomly select train and test data for the classes with more samples. Table 6 shows the accuracy results of each model.

Los Angeles. model reaches up to 78.02% accuracy with n-grams and SVM. Similar results are obtained with other classification methods and data representation pairs, such as raw and RF, n-grams and MLP, tf-idf, and SVM. For Los Angeles, C1D could not become prominent; the less complex models perform better on this dataset.

With Miami, we reach up to 99.54% accuracy with tf-idf and C1D. Overall, tf-idf is shown to be a better representation for this particular dataset. Combining a complex model with a representative feature, we achieved high accuracy.

For New Jersey, we achieve 92.43% accuracy with tf-idf and C1D. According to the results, tf-idf features better represent the New Jersey dataset.

In New York City, we reach up to 84.26% accuracy with raw data and RF. When we examine the dataset, we observed that the elevations fluctuate mostly between 13 ft and 95 ft. When such a small range is considered, decimal digit precision plays an important role. Since we do not discard any precision in the raw dataset, it is reasonable to have better accuracy with raw data. Although the highest accuracy is obtained with raw data and RF, tf-idf is a better choice for other classification methods.

In San Francisco, we achieve 80.25% accuracy with n-grams and MLP. Both text-like representations present similar accuracy patterns.

For Washington DC, we obtain 89.61% accuracy with n-grams and C1D. For both text-like representations, C1D shows better performance than other methods.

Overall, we can clearly observe the difference between TM – 1 results and TM – 2 results. The two main reasons for this performance gap are that (i) there are no overlapped or repetitive routes among the mined segments in the borough-level dataset, and (ii) the elevation differences and elevation sequences are not distinctive enough within a city to decide on which borough is the given test data is. The results of the simulated behavior will be discussed in the simulations subsection.

Evaluating TM – 3. In TM – 3 evaluations, due to sample size differences across the labels in the city-level dataset, we follow the same procedure in TM – 1 evaluations. A fixed number of samples is randomly selected from each class for training and testing. Table 7 shows the results of the evaluation. Per the reported results, we are able to predict the city of an elevation profile among 10 cities with an accuracy of 95.36%, among 8 cities with an accuracy of 95.98%, among 7 cities with an accuracy of 94.94%, among 5 cities with an accuracy of 93.00%, and among 3 cities with an accuracy of 88.99%. For TM – 3, for all number of classes, we find the best performing configuration as raw data and RF. When we look into the reason for the fact that raw data with RF outperforms every other configuration, we observe that the elevation range of the different classes in this dataset plays an important role, similar to TM – 2: NYC. Decision trees in RF are able to capture the features firsthand, without any representation needed in the middle.

When we consider the text-like representations, we observe that tf-idf features better represent the dataset. The success of the city-level estimations, when compared to the borough-level estimations (TM – 2), is due to 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 the city-level dataset also does not contain comprehensive, repetitive, and overlapped samples. The results of the simulated evaluation will be discussed in the following.

6.1.2 Simulations
The mined datasets do not contain overlapped or duplicate samples as in the user-specific dataset. In this set of evaluations, we simulate overlapped mined datasets and perform evaluations under the same threat models.

Simulation of TM – 2. For the city-level estimation evaluations, we rebuild a simulation dataset with a 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 a fixed n-grams size of 5. Figs. 10a and Fig. 10b show the comparison between the best-achieved result in the original evaluation and the best-achieved result in the simulations. The increase in the accuracy confirms our previous hypothesis that having overlapped route samples would increase the accuracy. Since the mined dataset is not specific to any target user’s mobility pattern, 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 rebuild a simulation dataset with a 35% overlap ratio for each city and performed the same evaluation with 10-fold cross-validation and 5-grams. Fig. 10c shows the comparison of the best-achieved accuracy results in original evaluations and simulations. As expected, the accuracy is increased in the simulations proving our previous hypothesis that having similar patterns in a dataset affects the success of the attack.

6.2 Image-Like Data Evaluations
In this set of evaluations, we perform experiments on the image-like representations of the data. Since the original data is unbalanced, the dataset built with the image-like representation also inherits the problem. In this section, we explain the methods to avoid bias due to an unbalanced dataset and discuss the associated results.

Dealing with Unbalanced Dataset. 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 a great amount of data, and oversampling raises the chances of getting lower accuracy as the misclassified duplicated samples increase the false ratio. Therefore, we explore other alternatives: (i) weighted loss function and (ii) fine-tuning with different samples.

Weighted Loss Function. For the unbalanced dataset, we utilize a weighted loss function while training the CNN and use 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 introduce rounds and create a set of small datasets from the unbalanced datasets for each round. As illustrated in Fig. 11, several small and balanced datasets are created by randomly selecting samples. For each consecutive round, 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 is reversed since the impact of the

| C | SVM | RF | MLP | C1D | LSTM | SVM | RF | MLP | C1D | LSTM | SVM | RF | MLP | C1D | LSTM |
|---|----|----|-----|-----|-----|----|----|-----|-----|-----|----|----|-----|-----|-----|
| 3 | 61.53 | 88.99 | 64.29 | 82.53 | 56.15 | 76.70 | 78.04 | 77.20 | 65.23 | 33.92 | 87.42 | 81.33 | 81.81 | 65.99 | 20.11 |
| 5 | 73.14 | 93.00 | 73.53 | 65.65 | 70.11 | 80.33 | 78.67 | 79.53 | 53.92 | 20.28 | 86.18 | 82.78 | 84.11 | 52.38 | 20.11 |
| 7 | 77.58 | 94.94 | 80.60 | 52.58 | 64.98 | 85.22 | 84.73 | 84.82 | 43.50 | 13.86 | 88.59 | 88.01 | 87.27 | 48.14 | 14.23 |
| 8 | 80.60 | 95.98 | 80.50 | 44.57 | 67.55 | 84.77 | 84.85 | 85.12 | 43.79 | 14.03 | 87.19 | 86.24 | 86.55 | 50.28 | 12.50 |
| 10 | 83.72 | 95.36 | 84.11 | 40.81 | 59.20 | 87.46 | 87.78 | 87.12 | 31.22 | 16.95 | 89.48 | 87.99 | 88.41 | 43.86 | 12.56 |

Table 7: The Overall Evaluation Results for TM – 3

Accuracy (%) with different data representations and classification models.

Fig. 10. Some selected simulation results of TM – 2 and TM – 3. The maximum achieved accuracy results are compared.
successful in outputting the largest class and can be considered as a compilation of the dataset of 6 cit-

In this study, we are strictly concerned with the

The Unweighted Loss (UWL) column is not considered while deciding the maximum achieved accuracy for each dataset and method. Even due to the unbalanced dataset. Table 8 shows the maximum achieved prediction accuracy for each method 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 is 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 8 highlights the maximum achieved prediction accuracy along with comparisons with other methods.

Weighted Versus Unweighted Loss Function. To observe the impact of the weighted loss function, we conduct 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 8 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. In TM – 1 and TM – 3, the accuracy scores of unweighted and weighted loss functions are considerably close. Thus, we conclude that the weighted loss function improved the prediction performance primarily for TM – 2.

Fine-Tuning Versus Weighted Loss Function. For the fine-tuning evaluations, 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 to use all the data we have, we decided to downsample the classes with a large sample size. For instance, in the evaluation of TM – 2: New York City, the biggest class has 5,455 samples whereas the second biggest class has 960 samples. In such cases, we did not create an additional round for only one class as this round would have a strong influence over the predictions, i.e., overfitting.

Table 8 shows the fine-tuning method outperformed the weighted loss function method only for TM – 2: WDC. 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. Overall, according to the results shown in Tables 9 and 10, the fine-tuning evaluation is not as successful as the weighted loss function evaluation, since we still lose some data while creating rounds.

When we compare text-like and image-like representations, we can conclude that text-like representation is a better choice for such attacks. For all evaluations except TM – 3, text-like representation outperformed image-like representation. For TM – 3 and TM – 2:NYC, the raw data and RF configuration is the best choice.

7 DISCUSSION

Defenses. In this study, we are strictly concerned with the elevation profile as a representation of the location. We note
that the location itself is the eventual modality of interest, but the exposed information to the adversary from which the adversary will make the inferences is the elevation. To that end, however, the same technique used for location perturbation [27] to defend against location privacy breaches can be applied on the elevation profile, although more straightforwardly and effectively. Namely, two broad classes of defenses could be applied to our problem domain: perturbation and aggregation.

Perturbation as a Defense To thwart our inference attacks, we can perturb the elevation while maintaining its overall statistical features used for the original application by adding a carefully crafted Gaussian noise driven from the elevation distribution (zero mean and a standard deviation of the original data). We hypothesize that the noise will affect the classifier but will not impact the validity of some of the driven insight from the elevation for the user (e.g., total elevation\(^2\)). To highlight the validity of this approach, we conducted a limited experiment to perturb 10\% of the original raw data used in the original classification problem in Table 5. To keep the elevation signal plausible and convincing, we superimposed dependently generated noise over an epoch of time (10 seconds) and “clipped” the generated noise with one standard deviation of a moving average in the current segment\(^3\). As a result, we were able to reduce the accuracy from 95.29\% (in the case of SVM, C=2) to 72.46\%. Similar performance degradation is observed with the n-grams and tf-idf in the same settings: by generating those features from the perturbed data, we were able to achieve an accuracy of 67.19\% and 70.31\% with the n-grams and tf-idf, respectively. While not totally subverting the inference, the approach shows an initial promising direction.

Aggregation as a Defense Another defense is realized by aggregating the elevation profile information into application-compliant statistics, e.g., total ascend, descend, mean ascend and descend, their standard deviation, minimum or maximum (over quantized elevation profile) or sampled elevation signal, which would reduce the effectiveness of our inference attack or block it altogether. The objective of the defense can be realized by rounding (10s of feet of elevation) so that the number of possible segments associated with the given statistical features is large enough to provide plausible deniability through anonymity (e.g., the correlation between the aggregate and location is weakened).

Feature-level Defenses and Caveats We note that other approaches that directly perturb the feature representation modality (e.g., images [28]) may not be as practical, since the image itself is a constrained domain, and not every pixel in the image domain is a valid perturbation candidate. We emphasize, however, that this issue is not particular to this problem space we address in this paper, but applicable to a range of problems in general, such as software [29], [30], network domains [32], where the feature representation used for implementing the machine learning algorithm transforms the input by upholding a dependency among the features, which is not the case in the original image modality used in computer vision applications [33].

Why Elevation and Actual Implications. We emphasize that we consider the elevation profile information as our inference input because this feature modality might be viewed, even to the most privacy-savvy individuals, as an innocent modality that does not necessarily expose much information about a specific location. Technically, however, it is not far-fetched to assume that there is an infinite number of mappings between a given elevation profile and location segments, which is shown to be not the case in this study under various plausible adversarial settings and objectives.

Moreover, we recognize the subjectivity of valuing location privacy by users [34], [35]. For instance, while some users might not care about sharing their exact location information all the time, some others may not feel comfortable sharing such information [36]. We demonstrate that such users, even when under the impression that they are not exposing their direct location, would be allowing an adversary to infer that location indirectly from the shared elevation. The sharing itself might put those users at risk, particularly when coupled with context. Imagine, for instance, an adversary who knows where a victim lives, but is able to infer that the activity of the victim, posted live, is associated with a location where the victim does not live, which would allow the adversary to stalk, or even break into the victim’s house. By the same token, an adversary who is able to precisely locate such an activity, in real-time, might be able to launch the physical attack possible under the sharing of the direct location (e.g., theft of expensive biking gear [4]).

Other Activities and Associated Risks. We note that, in general, mobility patterns are both individual and activity-dependent [37]. For instance, the mobility of a salesperson would be totally different from the activity of a student. However, we note that our study is chiefly concerned with mobility patterns of specific activities: exercise. It is very difficult to envision a plausible scenario where a salesperson, for instance, would share the elevation profile of their activities moving door-to-door to sell a product. Similarly, while activities, in general, are tracked by various smartwatches (for everyday use), such tracking is limited to the high-level aggregate (e.g., total, sampled over epochs of time), and is not shareable directly. To this end, activities that are not exercise-related (on public trackable roads) are considered out of the scope of our use and attack models.

Explaining the Performance of Our Techniques. We note that, generally, the image and text representations outperformed the raw data used directly for our inference attack (except in the case of LSTM and random forest). The reason why LSTM performed well over the raw data is because of the natural mapping between such raw data (i.e., time series) and the operation of LSTM with attention to long sequences and their dependencies. In other words, LSTM is capable of representing the features of time series well. On the other hand, the random forest technique is known to be tolerant to noise, so one plausible explanation for its superior performance is its implicit feature selection and representation (by creating decision points on ranges). On the other hand, CNN worked well with image-like representations for its
power in the representation of features from such modality. By the same token, SVM and MLP are shown to be superior in the text-like features for their tolerance to noise achieved by adjusting decision margins.

In general, we also note that CONV1D performed better for TM – 1, but not for TM – 2 and TM – 3, by demonstrating the power of CNN in general for capturing repeating patterns: TM – 1 is concerned with profiling persons with a number of activities, some of which might be overlapping or even repeated, and CNN is known to extract high quality, representative, and discriminative features in that space, in contrast to the scenarios of TM – 2 and TM – 3, where such patterns are less manifested.

Finally, we note that natural languages in general have various semantic and syntactic characteristics that are not manifested in the non-constrained domain of elevation profiles. To that end, while LSTM, when applied to features driven from the tf-idf or n-grams of a natural language utterance, would perform well, the same technique is not guaranteed to perform as well using the same feature representations, as we demonstrate in this study. One plausible explanation also for the superior performance of LSTM over the raw data is that the raw data explicitly and fully maintains an ordering that is essential for the operation of LSTM, whereas that ordering is implicit and for very short utterances in tf-idf and n-grams. Losing such valuable information as a result of feature extraction would (although marginally) affect the performance, as shown in this study.

8 RELATED WORK

In this work, we addressed the problem of location privacy in activity trackers using the side-channel information obtained from publicly shared elevation profiles. While there is no work that explicitly addresses this topic, there is has some studies on various topics that are related in the broad literature [38], [39], [40]. 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. [41] 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. Abdelmoty and Alrayes [42] aimed to increase the awareness of information leakage in social networks, particularly fitness apps in which the data is mostly not uniformly distributed. Map matching methods [50] aim to restrict the obfuscated area to a smaller but plausible area by removing irrelevant areas. Movement boundary attacks were explored [51], 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 workplace, is inferred and the irrelevant locations are discarded.

Although we did not directly touch upon preserving the location privacy in our study, there have 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 [52]. Additionally, the term k-anonymity is defined as obscuring the location information of individuals with k number of other individuals within the region [53], [54].

9 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%.

While this work highlights the clear trade-offs provided by the various defenses, their usability is largely unexplored. In our future work, we will explore the usability of 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 and acceptance.

References
[1] Ü. Meteriz, N. F. Yiotaldiotaran, J. Kim, and D. A. Mohaisen, “Understanding the potential risks of sharing elevation information on fitness applications,” in Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst., 2020, pp. 464–473.
[2] J. P. Higgins, “Smartphone applications for patients’ health & fitness,” Amer. J. Med., vol. 129, no. 1, pp. 11–19, 2015.
[3] I. Polakis, G. Argyros, S. Sivakorn, and A. D. Keromytis, “Where’s wally?: Precise user discovery attacks in location proximity services,” in Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur., 2015, pp. 817–828.
[4] G. Friedland and R. Sommer, “Cybercasing the joint: On the privacy implications of geo-tagging,” in Proc. 5th USENIX Conf. Hot Topics Secur., 2010, pp. 1–8.
[5] “Map/privacy,” 2020. [Online]. Available: https://bit.ly/2TgwWJtu
[6] “How do i remove display map from my activity list?,” 2020. [Online]. Available: https://bit.ly/2TPYyDF
[7] “Hide map,” 2020. [Online]. Available: https://bit.ly/3qaqKwe.
[8] “Privacy setting to hide activity map from non-followers,” 2020. [Online]. Available: https://bit.ly/3q4Ey9T
[9] S. Padilla, I. Juantxia, G. Cuesta, and J. Goirierna, “Level ground and uphill cycling ability in professional road cycling,” Med. Sci. Sports Exercise, vol. 31 no. 6, pp. 878–85, 1999.
[10] “Strava global heatmap,” 2019. [Online]. Available: https://www.strava.com/heatmap
[11] “U.S. soldiers are revealing sensitive and dangerous information by jogging,” 2018. Accessed: Apr. 20, 2021. [Online]. Available: https://bit.ly/UM3SVJ
[12] “Fitness tracking app strava gives away location of secret US army bases,” 2018. Accessed: Apr. 20, 2019. [Online]. Available: https://bit.ly/3vdfVNT
[13] S. Loughran, “Advanced deanonymization through strava,” Accessed: Apr. 20, 2019, 2019. [Online]. Available: https://bit.ly/3eURNiH
[14] “Strava honeypot app led thieves to my £12,000 bike collection.” 2018. Accessed: Apr. 20, 2021. [Online]. Available: https://bit.ly/35wvkcB
[15] “Cyclists warned to beware sharing data on ride-tracking apps,” 2018. Accessed: Apr. 20, 2021. [Online]. Available: https://bit.ly/3g66fRe
[16] “Cyclist who had five bikes stolen says thieves are looking for quick times on strava to try and find high-end bikes – warns other users to check their privacy settings,” 2018. Accessed: Apr. 20, 2021. [Online]. Available: https://bit.ly/3gp8xtp.
[17] “Organised criminals using fitness apps to track expensive bicycles and equipment,” 2020. Accessed: Apr. 20, 2021. [Online]. Available: https://bit.ly/2SR1In1
[18] 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 Proc. 27th USENIX Secur. Symp., 2018, pp. 497–512. [Online]. Available: [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/hassan
[19] “Runtastic deserves more ‘heat’ than strava!” 2018. Accessed: Apr. 20, 2021. [Online]. Available: https://bit.ly/2sqgCyD
[20] I. Maglogiannis, K. Karpouzis, M. Wallace, and J. Soldatos Eds., Emerging Artificial Intelligence Applications in Computer Engineering - Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, vol. 160. Amsterdam, The Netherlands: IOS Press, 2022.
[21] C. Y. Hsu, C.-C. Chang, and C.-J. Lin, “A practical guide to support vector classification,” Data Sci. Assoc., pp. 1396–1400, 2003.
[22] L. Breiman, “Aricing classifiers,” Ann. Statist., vol. 26, pp. 801–823, 1998.
[23] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001.
[24] 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,” Mach. Learn., vol. 40, no. 3, pp. 203–228, Sep 2000.
[25] P. W. Eklund and A. Hoang, “A performance survey of public domain supervised machine learning algorithms,” AUS. J. Intell. Inform. Syst., vol. 9, no. 1, pp. 1–47, 2002.
[26] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” CoRR abs/1412.6980
[27] R. Doshi, “Local differential perturbations: Location privacy under approximate knowledge attackers,” IEEE Trans. Mobile Comput., vol. 12, no. 12, pp. 2360–2372, Dec. 2013.
[28] S. J. Oh, M. Fritz, and B. Schiele, “Adversarial image perturbation for privacy protection a game theory perspective,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 1491–1500.
[29] H. Alasmary et al., “Analyzing and detecting emerging internet of things malware: A graph-based approach,” IEEE Internet Things J., vol. 6, no. 5, pp. 8977–8988, Oct. 2019.
[30] A. Khormali, A. Abusnaina, S. Chen, D. Nyang, and D. Mohaisen, “From blue-sky to practical adversarial learning,” in Proc. IEEE 2nd Int. Conf. Trust, Privacy Secur. Intell. Syst. Appl., 2020, pp. 118–127.
[31] H. Alasmary et al., “Soteria: Detecting adversarial examples in control flow graph-based malware classifiers,” in Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst., 2020, pp. 886–889.
[32] A. Cherkinova and A. Oprea, “Fence: Feasible evasion attacks on neural networks in constrained environments,” ACM Trans. Privacy Secur., vol. 25, no. 4, pp. 1–34, 2022.
[33] A. Abusnaina et al., “Adversarial example detection using latent neighborhood graph,” in Proc. IEEE/ICVF Int. Conf. Comput. Vis., 2020, pp. 7675–7685.
[34] K. Fawaz, H. Feng, and K. G. Shin, “Anatomization and protection of mobile Apps’ location privacy threats,” in Proc. 24th [USENIX] Secur. Symp., 2015, pp. 753–768.
[35] B. P. Knijnenburg, A. Kobsa, and H. Jin, “Preference-based location sharing: Are more privacy options really better?” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2013, pp. 2667–2676.
[36] J. Krumm, “A survey of computational location privacy,” Pers. Ubiquitous Comput., vol. 13, no. 6, pp. 391–399, 2009.
[37] S. Hasan, X. Zhan, and S. V. Ukkusuri, “Understanding urban human activity and mobility patterns using large-scale location-based data from online social media,” in Proc. 2nd ACM SIGKDD Workshop Urban Comput., 2013, pp. 1–8.
[38] S. Narain, A. Ranganathan, and G. Noubir, “Security of GPS/INS based on-road location tracking systems,” in Proc. IEEE Symp. Secure Commun. Privacy, 2019, pp. 837–841.
[39] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, “The perils of user tracking using zero-permission mobile apps,” IEEE Secur. Privacy, vol. 15, no. 2, pp. 32–41, Mar./Apr. 2017.
[40] S. Narain and G. Noubir, “Mitigating location privacy attacks on mobile devices using dynamic app sandboxing,” Proc. Privacy Enhancing Technol., vol. 2019, no. 2, pp. 66–87, 2019.
[41] A. N. Akpetyu, J. Niu, C.-C. Chang, and A. N. Goldsmith, “Unwinding anariade’s identity threat: Privacy risks with fitness trackers and online social networks,” Proc. Multimedia Privacy Secur., 2017, pp. 1–11.
[42] A. I. Abdelmoty and F. Alrayes, “Towards understanding location privacy awareness on geo-social networks,” ISPRS Int. J. Geo-Inf., vol. 6, no. 4, 2017, Art. no. 109. [Online]. Available: http://www.mdpi.com/2220-9964/6/4/109
[43] Z. Tu, F. Xu, Y. Li, P. Zhang, and D. Jin, “A new privacy breach: User trajectory recovery from aggregated mobility data,” IEEE/ACM Trans. Netw., vol. 26, no. 3, pp. 1446–1459, Jun. 2018.

[44] 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 Proc. 5th Int. Conf. Web Soc. Media, 2015, pp. 553–561. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10487

[45] 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 Proc. 9th Int. Conf. Web Soc. Media, 2015, pp. 387–396. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM15/paper/view/10498

[46] M. Wernke, P. Skvorstov, F. Dürr, and K. Rothermel, “A classification of location privacy attacks and approaches,” Pers. Ubiquitous Comput., vol. 18, no. 1, pp. 163–175, Jan. 2014.

[47] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “L-diversity: Privacy beyond k-anonymity,” in Proc. 22nd Int. Conf. Data Eng., 2006, pp. 24–24.

[48] M. F. Mokbel, “Privacy in location-based services: State-of-the-art and research directions,” in Proc. Int. Conf. Mobile Data Manage., 2007, pp. 228–228.

[49] R. Shokri, G. Theodorakopoulos, J. Le Boudec, and J. Hubaux, “Quantifying location privacy,” in Proc. IEEE Symp. Secur. Privacy, 2011, pp. 247–262.

[50] J. Krumm, “Inference attacks on location tracks,” in Pervasive Computing, A. LaMarca, M. Langheinrich, and K. N. Truong, Eds., Berlin, Heidelberg: Springer, 2007, pp. 127–145.

[51] G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino, “Preventing velocity-based linkage attacks in location-aware applications,” in Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2009, pp. 246–255.

[52] M. Duckham and L. Kulik, “A formal model of obfuscation and negotiation for location privacy,” in Proc. 3rd Int. Conf. Pervasive Comput., 2005, pp. 152–170.

[53] 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.

[54] A. Gkoulalas-Divanis, P. Kahnis, and V. S. Verykios, “Providing K-anonymity in location based services,” SIGKDD Explorations Newslett., vol. 12, no. 1, pp. 3–10, Nov. 2010.

Ulku Meteriz-Yildiran (Senior Member, IEEE) received the BSc degree in computer engineering from Middle East Technical University, in 2018, and the MSc and PhD degrees in computer science with a machine learning concentration from the University of Central Florida, in 2022 and 2022, respectively. Her research interests are in applied machine learning on the privacy and security of wearable devices. She is a member of the Security and Analytics Lab (SEAL) with the University of Central Florida since 2018. In 2021, she interned with Amazon Web Services (AWS) as a software development engineer and is currently a machine learning researcher with Meta.

Necip Fazıl Yıldırım received the BSc degree in computer engineering from Middle East Technical University, in 2018, the MSc degree in computer science from the University of Central Florida, in 2020 with a concentration on applied machine learning on the security and privacy of IoT, and the PhD degree from the University of Central Florida, in 2022 with a research focus on the analysis of highly configurable software. He interned with Google as a software engineer, in 2020 and 2021, where he is currently a full-time software engineer.

Joongheon Kim (Senior Member, IEEE) received the BS and MS degrees in computer science and engineering from Korea University, Seoul, Korea, in 2004 and 2006, respectively, and the PhD degree in computer science from the University of Southern California (USC), Los Angeles, CA, USA, in 2014. He has been with the School of Electrical Engineering, Korea University, Seoul, Korea, since 2019, where he is currently an associate professor. Before joining Korea University, he was with LG Electronics (Seoul, Korea, 2006–2009), InterDigital (San Diego, CA, USA, 2012), Intel Corporation (Santa Clara in Silicon Valley, CA, USA, 2013–2016), and Chung-Ang University (Seoul, Korea, 2016–2019). He serves as an associate editor for IEEE Transactions on Vehicular Technology, In the UT more than 90 journals, 110 conference papers, and 6 book chapters. He also holds more than 50 granted patents.

David Mohaisen (Senior Member, IEEE) received the MSc and PhD degrees from the University of Minnesota, in 2011 and 2012, respectively. He is a full professor of computer science with the University of Central Florida, where he leads the Security and Analytics Lab (SEAL) and has been since 2017. Previously, he was an assistant professor with SUNY Buffalo in 2015-2017, and a senior scientist with Verisign Labs, in 2012-2015. His research interests are in the broad area of applied security and privacy, covering aspects of computer and networked systems, software systems, IoT and AR/VR, and machine learning. His research has been supported by NSF, NRF, AFRL, AFOSR, etc., and has been published in top conferences and journals alike, with multiple best paper awards. His work was featured in the new scientist, MIT Technology Review, ACM Tech News, Science Daily, etc. Among other services, he is currently an associate editor of IEEE Transactions on Mobile Computing and IEEE Transactions on Parallel and Distributed Systems. He is a senior member of ACM (2018).

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.