A Black-Box Approach for Quantifying Leakage of Trace-Based Correlated Data

Shafizur Rahman Seeam
Rochester Institute of Technology
Rochester, USA
ss6365@rit.edu

Zhengxiong Li
University of Colorado Denver
Denver, USA
zhengxiong.li@ucdenver.edu

Yidan Hu
Rochester Institute of Technology
Rochester, USA
yidan.hu@rit.edu

ABSTRACT

Quantification of information leakage is crucial, especially for privacy-preserving systems such as location-based services (LBS) with integrated privacy mechanisms. Existing quantification mainly utilized two approaches: i) the white-box approach, precise but impractical for complex systems, and ii) the black-box approach, practical but struggles with scalability for large output spaces. Recently, Machine Learning (ML) algorithms have been integrated into the black-box approach to effectively approximate information leakage for independent observations with better scalability. However, this method does not provide precise estimates for dependent observations. Intuitively, once a correlated secret is discovered, it becomes easier for an attacker to predict related secrets, leading to an underestimation of information leakage. This paper introduces an ML-based black-box approach to improve the accuracy of information leakage estimation for systems with correlated data, particularly in trace-based scenarios. Our solution uses an ML model for rough estimation and leverages data correlations to refine inferences for more accurate quantification. Evaluation results from three real-world datasets and one collected dataset confirm our solution's effectiveness in accurately and cost-effectively quantifying system leakage for correlated observations.

CCS CONCEPTS

• **Security and privacy** → *Systems security*.

KEYWORDS

Leakage Estimation, Location Privacy, Trace Privacy

PICASSO 24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11
https://doi.org/10.1145/3636534.3694722

ACM Reference Format:
Shafizur Rahman Seeam, Zhengxiong Li, and Yidan Hu. 2024. A Black-Box Approach for Quantifying Leakage of Trace-Based Correlated Data. In *International Workshop on Physics Embedded AI Solutions in Mobile Computing (PICASSO 24), November 18–22, 2024, Washington D.C., DC, USA.* The 7th International Workshop on Physics Embedded AI Solutions in Mobile Computing, Washington, D.C., USA, 6 pages. https://doi.org/10.1145/3636534.3694722

1 INTRODUCTION

The measurement of information leakage of a system is a fundamental aspect of security, particularly for privacy-preserving systems. For example, in LBS with privacy protection mechanisms [2], measuring the amount of sensitive information an adversary can obtain is of utmost importance to understand whether such leakage can be tolerated or must be considered a major security flaw. It can also serve as a guide for the selection of advanced privacy techniques and to enhance overall system security. Unfortunately, accurately quantifying information leakage remains challenging due to the system's inherent complexity.

Existing solutions for quantification of information leakage mainly include two categories: white-box approaches and black-box approaches. Specifically, white-box techniques compute the desired leakage measures with the assumption that the system channel, including the conditional probabilities of the outputs (e.g., observations) given the inputs (e.g., secrets), is known [1]. However, this assumption is often impractical due to the unknown nature of the system channel. Even if these conditional probabilities are known, analytic computation can be challenging for complex systems such as privacy-preserving LBS systems that require detailed analysis among numerous pairs of inputs and outputs. In contrast, black-box approaches [4, 5] assume that the system's internals are unknown. They are based on collecting extensive datasets of input-output pairs and using the relative frequencies of these pairs to approximate the joint probability distribution to measure information leakage. However, these methods face scalability issues in applications with large output spaces, requiring sample sizes much larger than the product of possible inputs and outputs for reliable estimates. Additionally, they struggle with novel observations that are not present in the training data.

Recently, Machine Learning (ML) algorithms have increasingly been integrated into the black-box approach due to their scalability and ability to generalize beyond the training data. In particular, Cherubin et al. [8] leverage ML algorithms to calculate the Bayes risk, which is the smallest error achievable by an adversary in predicting a secret input from the observable output. This approach is effective for assessing information leakage in privacy-preserving LBS, where an adversary infers a user's real location (i.e., secret) from the released obfuscated location (i.e., observation). However, extending this method may not be suitable for trace-based attacks, where the attacker predicts subsequent secrets based on subsequent observations. This type of attack involves sequences of location points that are linked by spatio-temporal correlations, meaning that predicting one point in the sequence provides insights that facilitate the prediction of subsequent points. Consequently, any attack model that overlooks the interdependencies among sequential observations will likely underestimate the extent of information leakage (i.e., the Bayes risk). Thus, there is a pressing need to design a more sophisticated approach to accurately measure information leakage for systems with correlated observations.

This paper introduces an ML-based black-box approach to accurately measure information leakage for systems with correlated data, such as a trace. In particular, we propose an advanced model utilizing the k-Nearest Neighbors (k-NN) algorithm that harnesses the spatio-temporal relation in sequential observations to predict subsequent secrets within a trace. Our key contributions are summarized below.

- This paper addresses a critical gap in accurately quantifying information leakage for systems with correlated data, particularly for LBS with trace-based data.
- We introduce an ML-based black-box approach that leverages spatio-temporal relationships between consecutive location points to accurately assess information leakage, quantified through Bayes risk.
- We evaluated our proposed solution on four real-world datasets, including one we collected. Our results demonstrate that the strawman approach underestimates information leakage in LBS with trace-based data, while our solution accurately quantifies this leakage.

We release the code[1] to replicate the experiments.

2 RELATED WORKS

The initial methodologies for quantifying information leakage were based on probabilistic measures that required a precise understanding of the system's behavior [16, 18]. These "white-box" approaches require detailed formal definitions and specific assumptions [6] about the systems, making them impractical for modeling trace-based distributions, where such precise knowledge is often unattainable. The frequentist (i.e., black-box) paradigm gained prominence because it can quickly quantify approximate information leakage in real-world scenarios [5, 9, 11, 15]. This approach has been further refined to include the calculation of confidence intervals for the estimated leakage [10]. However, it faces significant scalability issues, particularly when the output space is prohibitively large or involves continuous distributions. For example, LeakWatch [12] necessitates a sample size that vastly exceeds the product of input and output space sizes. Moreover, this approach also struggles with continuous output spaces, such as those in trace-based distributions.

Cherubin et al. [8] introduced an innovative approach using ML algorithms that utilize a metric on the output space to achieve faster convergence than the frequentist methods. This ML-based approach proves advantageous even when faced with i) a large output space, and ii) observations not in the training set. However, this model primarily approximates the leakage for independent observations and does not address the complexities of correlated observations, which are crucial in trace-based distributions. This gap underscores the need for advanced ML-based approaches to measure information leakage in systems with correlated data.

3 PRELIMINARIES

3.1 Location Privacy Mechanism

Planar Laplace mechanism (PL_ϵ) [2] ensures that an individual's location is indistinguishable within a certain radius r by reporting a location $z \in \mathcal{R}^2$ instead of the actual location $x \in \mathcal{R}^2$, generated randomly according to the noise function:

$$\mathbf{D}_\epsilon(x)(z) = \frac{\epsilon^2}{2\pi} e^{-\epsilon d(x,z)} \tag{1}$$

where ϵ is the privacy budget, $\frac{\epsilon^2}{2\pi}$ is a normalization factor, $d(x, z)$ is the Euclidean distance between x and z.

3.2 Leakage Measure

The Bayes risk $\mathcal{R}^*$ quantifies the minimum expected error of an optimal adversary [7, 8, 19] , who knows the true distribution $\mu(o, s)$, when predicting confidential data (i.e., actual locations) from observed outputs (i.e., perturbed locations). It is defined with respect to a loss function $\mathcal{L}$, typically a loss of 0-1, where $\mathcal{L}(s, s') = \mathcal{I}(s \neq s')$, taking the value 1 if $s \neq s'$, 0 otherwise. For a finite set of secrets $s \in \mathcal{S}$ and observed outputs $o \in \mathcal{O}$, the Bayes risk is given by:

$$\mathcal{R}^* := 1 - \sum_{o \in O} \max_{s \in S} \mu(o, s) \tag{2}$$

The true Bayes risk $\mathcal{R}^*$ is difficult to determine analytically because the actual distribution is often unknown. Instead, an estimate $\hat{\mathcal{R}}_n$ is calculated based on n examples,

[1] https://github.com/shafizurRseeam/BBoxPicasso

$\{(o_1, s_1), \ldots, (o_n, s_n)\}$, sampled from the joint distribution $\mathcal{S} \times \mathcal{O}$. Specifically, let f_n be a decision rule learned by an adversary from n examples. The estimated Bayes risk [8] is:

$$\hat{\mathcal{R}_{f_n}} = 1 - \sum_{o \in O} \max_{s \in \mathcal{S}} \Pr(f(o), s) \tag{3}$$

where $f_n(o)$ is the secret predicted for object o, $\Pr(f_n(o), s) = \Pr(o|s) \times \pi(s)$, and $\pi(s)$ is a prior probability of having a secret s, obtained via the frequentist [5].

3.3 ML Estimation of the Bayes Risk

In ML, $f_n : \mathcal{O} \to \mathcal{S}$ is a classifier that maps observations in $\mathcal{O}$ to predictions in $\mathcal{S}$. Let $\mathcal{F} = \{f_n \mid f_n : \mathcal{O} \to \mathcal{S}\}$ denote the set of classifiers, and let μ be a distribution on $\mathcal{S} \times \mathcal{O}$. A learning rule uses a training set $\{(o_1, s_1), \ldots, (o_n, s_n)\}$ to select a classifier f_n from $\mathcal{F}$ aiming to minimize expected loss $\mathbb{E}[\mathcal{L}(f_n(o), s)]$ for new example (o, s) sampled from μ.

Definition 3.1. (Universally Consistent (UC) Learning Rule). Let $\mathcal{A}$ be a learning rule, $f_n \in \mathcal{F}$ be a classifier selected by $\mathcal{A}$ trained from sample μ, and $\hat{\mathcal{R}_{f_n}}$ be the expected error of f_n. $\mathcal{A}$ is consistent if $\hat{\mathcal{R}_{f_n}} \to \mathcal{R}^*$ as $n \to \infty$. $\mathcal{A}$ is universally consistent if it is consistent for all distributions μ.

k-Nearest Neighbor (k-NN) is a simple yet efficient ML algorithm that predicts the output by taking a majority vote from the secrets of its k nearest neighbors.

Definition 3.2 (k_n-NN rule [8]). Given a training set of n examples, the k_n-NN rule selects a k-NN classifier, where k is chosen so that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$.

k_n-NN rule is universally consistent [20], which means that the expected error of the k_n-NN rule converges to $\mathcal{R}^*$ as $n \to \infty$, where n is the size of the training sample.

For more on UC learning rule, see Devroye et al. [14].

3.4 Problem Formulation

Consider a privacy-preserving system that obfuscates a user's real location by a location privacy mechanism, such as PL_ϵ. The obfuscated locations are then submitted to a third-party service provider for an LBS. Let $\mathcal{S}$ denote a set of users' possible true locations (i.e., secrets), and $\mathcal{O}$ denote the obfuscated location (i.e., observations) generated by $\mathcal{M}$. As of time m, a user that records continuous real locations forms a trace $\mathcal{T}_s = \{s_1, s_2, \ldots, s_m\}$, where $s_i \in \mathcal{S}$, and $\mathcal{T}_o = \{o_1, o_2, \ldots, o_m\}$ denotes the corresponding obfuscated locations generated by $\mathcal{M}$, where $o_i = \mathcal{M}(s_i)$ and $o_i \in \mathcal{O}$. Assume that we have multiple such pairs of traces $(\mathcal{T}_o, \mathcal{T}_s)$ in any given LBS system. We aim to estimate Bayes risk, $\hat{\mathcal{R}}_{f_n}$ as in Eq. 3 that is close to the real Bayes risk as in Eq. 2, for this LBS system with traces to quantify the accurate information leakage using an ML-based decision rule f_n.

4 ML-BASED TRACE INFERENCE

This section outlines the rationale behind our design and provides an overview of our solution.

4.1 Design Rationale and Overview

Strawman Approach: We form a training set of n samples $\{(o_1, s_1), \cdots, (o_n, s_n)\}$ from pairs of traces $(\mathcal{T}_o, \mathcal{T}_s)$ to train an ML classifier (e.g., k-NN classifier) by minimizing the expected loss [21]. Next, for any observed trace $\mathcal{T}_o' = \{o_1, o_2, \cdots, o_m\}$, we independently adopted the trained classifier to infer the true location. In particular, for any o_i, $1 \le i \le m$, we infer the most probable location $\hat{s}_i$ as:

$$\hat{s}_i = \arg\max_{s \in S} Pr(s|o_i) \tag{4}$$

where $Pr(s|o_i)$ is the probability of the secret s given the observation o_i. We finally compute the estimated Bayes risk $\hat{\mathcal{R}}_{f_n}$ according to Eq. (3), where $\hat{s}_i$ is regarded as the secret achieving the highest joint probability, i.e., $Pr(f_n(o_i), \hat{s}_i)$ for observed location o_i in the trace $\mathcal{T}_o'$.

Design Rationale: Although the strawman approach can efficiently estimate the Bayes risk in a black-box manner, it tends to underestimate the Bayes risk within the locations in the traces due to its disregard for the inherent spatio-temporal correlations. We extend the machine learning model from single-point predictions to trace-based predictions by capitalizing on the inherent correlations within sequential location data. This extension involves integrating a higher-order transition probability matrix to model movements between successive locations in a trace effectively.

4.2 Proposed Framework

We first train an ML classifier as in the strawman approach and formulate spatio-temporal correlations among locations as a transition probability matrix, which is obtained by counting the number of transitions from each location to every other location and converting the counts to probabilities. Next, for any observed trace $\mathcal{T}_o' = \{o_1, o_2, \cdots, o_m\}$ with m sequential locations, we infer the corresponding real locations $\hat{s}_t$, $t \in \{1, \cdots, m\}$ as follows.

For the initial location o_1 on the observed trace, as in the strawman approach, we directly adopt the trained classifier to predict the most probable location $\hat{s}_1$ using Eq. 4 with the initial observation o_1.

For each subsequent point o_t, where $t = \{2, \cdots, m\}$, the prediction $\hat{s}_t$ is determined not only by the current observation o_t but also by the most recent r previous predictions forming the historical context $\mathcal{H}_t = \{\hat{s}_{t-1}, \hat{s}_{t-2}, \cdots, \hat{s}_{t-r}\}$. Denote by $P(s|o_t, \mathcal{H}_t)$ the inference procedure that naturally integrates the spatio-temporal correlation among locations in a trace into the ML-based classification considering both

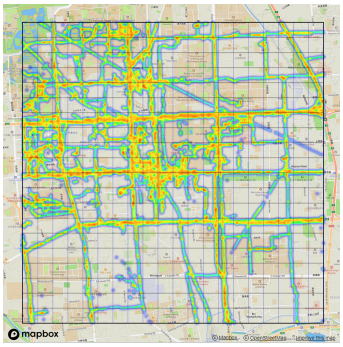


Figure 1: $6 \times 6\ km^2$ area centered around Beijing, China from Geolife dataset.

Table 1: Characteristics of Preprocessed Datasets

Dataset	$\overline{Dis\mathcal{T}}$(km)	$\overline{Dis}$(m)	$\overline{SR}$(s)	$\overline{Loc}$
Geolife [23]	1,320	29	2	124,570
T-drive [22]	48,660	1,011	181	70,762
GPS [13]	408	46	6	11,122
Collected	312	6	1	63,771

$\mathcal{H}_t$ and o_t simultaneously. We infer the most probable location $\hat{s}_t$ as:

$$\hat{s}_t = \arg\max_{s \in S} P(s|o_t, \mathcal{H}_t) \tag{5}$$

The corresponding estimated Bayes Risk could be obtained according to Eq. 3, where $f_n(o)$ is estimated by Eq. 5.

Now a practical procedure to compute $P(s|o_t, \mathcal{H}_t)$ is defined by:

$$P(s|o_t, \mathcal{H}_t) = Pr(s|o_t) \times Pr(s|\mathcal{H}_t) \tag{6}$$

where $Pr(s|o_t)$ is the probability of having s predicted by the trained classifier with input o_t. $Pr(s|\mathcal{H}_t)$ is the transition probability that represents the conditional probability of transition to state s given a historical context $\mathcal{H}_t$.

The computation of $P(s|\mathcal{H}_t)$ is facilitated through a Markovian model, which can accommodate longer dependencies and more sophisticated state transition dynamics. Specifically, this model weights the transitions from previous states to the current state, where more recent states are given higher weights in a decaying fashion:

$$P(s|\mathcal{H}_t) = \sum_{j=1}^{r} w_j Pr(s|\hat{s}_{t-j}) \tag{7}$$

where w_j are the weights that sum to 1 and decrease exponentially for older states in the history $\mathcal{H}_t$. $Pr(s|\hat{s}_{t-j})$ could be obtained by looking for the transition probability matrix.

Algorithm 1: Grid-Based Location Encoding

Input: $\mathcal{L} = \{(\phi_1, \lambda_1), \ldots, (\phi_n, \lambda_n)\}$, Grid size N
Output: Encoded Locations $\mathcal{S} = \{s_1, \ldots, s_n\}$

for *each* (ϕ, λ) *in* $\mathcal{L}$ **do**
  $\Delta\phi = \frac{\phi_{\max} - \phi_{\min}}{N}$, $\Delta\lambda = \frac{\lambda_{\max} - \lambda_{\min}}{N}$
  $j = \left\lfloor \frac{\phi - \phi_{\min}}{\Delta\phi} \right\rfloor$, $i = \left\lfloor \frac{\lambda - \lambda_{\min}}{\Delta\lambda} \right\rfloor$
  if $j < N$ **then**
    $j = N - 1$
  if $i < N$ **then**
    $i = N - 1$
  $s = i \cdot N + j$
  Append s to $\mathcal{S}$

5 EVALUATION

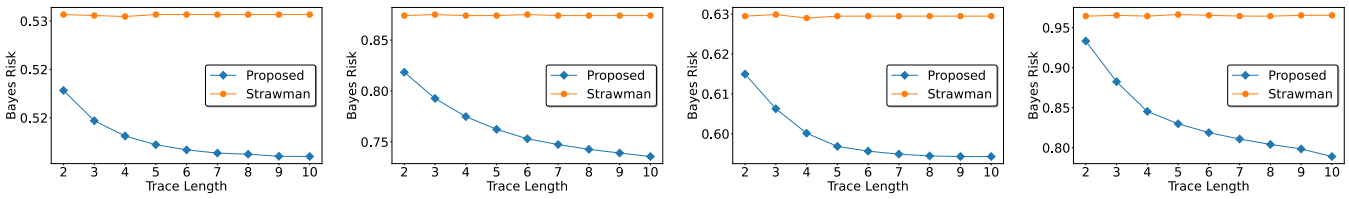
5.1 Datasets Preprocessing

We conducted experiments using four datasets: three public datasets and one proprietary dataset. Our data preprocessing workflow includes subsampling, perturbation, and encoding.

5.1.1 Subsampling: Initially, we randomly select a subset of each dataset for computational efficiency. They are summarized in Table. 1, where $\overline{Dis}$(m) is the median distance in meters between two adjacent locations in a trace, $\overline{Dis\mathcal{T}}$(km) is the total distance traveled in kilometers, $\overline{SR}$(s) is the median sampling rate in seconds, and $\overline{Loc}$ is the total number of locations. Next, we define a $6 \times 6\ km^2$ area with sufficient examples from each dataset and discretize it into 20×20 cells, each $300 \times 300\ m^2$. The secret space for the Geolife dataset is shown in Fig. 1, plotted using Mapbox [17]. Each trace has multiple adjacent locations, where each location consists of a pair of latitude (ϕ) and longitude (λ).

5.1.2 Perturbation: Assume that there are m traces or subtraces in the $6{\times}6\ km^2$ subarea, i.e., all locations in those traces are in the subarea. For each trace, we distorted the locations using PL_ϵ [2] with privacy budget $\epsilon = 1$. Specifically, let $o_i = (\phi_i, \lambda_i)$ be an i^{th} original location in the trace, where ϕ_i and λ_i are latitude and longitude, respectively. The corresponding obfuscated location generated by the PL_ϵ is $o'_i = (\phi'_i, \lambda'_i)$. These coordinates define the observation space O, which is accessible to the attacker. It is worth mentioning that the output domain of PL_ϵ is infinite and the perturbed outputs (ϕ'_i, λ'_i) may extend beyond the $6 \times 6\ km^2$ square region.

5.1.3 Encoding: We employ a grid-based encoding method that maps each real location (ϕ_i, λ_i) in a trace to a corresponding one-dimensional value $s \in \mathcal{S}$, to facilitate the training of the machine learning model. The encoding method is summarized in Algorithm 1, where $\phi_{\max}$, $\phi_{\min}$, $\lambda_{\max}$, and $\lambda_{\min}$ on



(a) T-drive (b) Geolife (c) GPS (d) Collected

Figure 2: Estimated Bayes risk across four datasets with varied trace length

line 2 are the maximal and minimal latitude and longitude in the selected area. Each s is the index of a cell after encoding, and the machine learning model predicts the encoded values $s \in \mathcal{S}$ from the perturbed observation space $o \in \mathcal{O}$.

5.2 Experimental Setting

We employ k-NN classifier as the ML model to estimate Bayes risk. First, we split dataset $\mathcal{D} = \{(o_1, s_1), \cdots, (o_n, s_n)\}$ into two distinct subsets: a training set $\mathcal{D}_t$, and a validation set $\mathcal{D}_v$. Next, we train the k-NN classifier following the same setting in [8] e.g., $k = \log(n)$ using the training set $\mathcal{D}_t$.

Next, we compute the estimated Bayes risk $\hat{\mathcal{R}}_{f_n}$ across the validation set $\mathcal{D}_v$ that includes τ pairs of traces and each trace consists of n_j observation-secret pairs.

$$\hat{\mathcal{R}}_{f_n} = \frac{1}{\tau} \sum_{j=1}^{\tau} \left(\frac{1}{n_j} \sum_{i=1}^{n_j} I(f(o_{ji}) \neq s_{ji}) \right) \tag{8}$$

where $I(\cdot)$ is 0-1 loss function, and $\hat{s}_{ji}$ is predicted location.

This estimate is usually biased with respect to a particular $(\mathcal{D}_t, \mathcal{D}_v)$, and thus we compute this repeatedly using different train-validation splits and then average their estimates. We set r as the trace length, that is, we consider all historical predictions to infer the location at the current time slot.

While we used k-NN classifier, it is important to evaluate various classifier with particular settings as certain classifiers may underperform in specific contexts [3]. All experiments carried out were performed on a MacBook Pro with an M2 chip and 16GB RAM. All experiments were carried out 100 times, and only the average was reported.

5.3 Results on Information Leakage

Fig. 2 show the estimated Bayes risks under strawman and our solution with trace lengths increasing from 2 to 10 across four datasets. We can see that the Bayes risk under the Strawman approach remains consistent as the trace length increases across all datasets, which is expected. This stability arises because the adversary treats each observation o_i in the observed trace $\mathcal{T}_o$ as independent, not incorporating the correlation in the inference of the true trace $\hat{\mathcal{T}}_s = \{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_m\}$. In contrast, our proposed method exhibits a clear trend: the Bayes risk decreases as the trace length increases. The reason is that for a trace with a longer trace length, the adversary could effectively utilize the increased side information, $\mathcal{H}_t$, to predict secrets s_i in $\mathcal{T}_s$ with greater accuracy. We can also see that the rate of decrease in Bayes risk is pronounced in Figs. 2b and 2d, compared to Figs. 2a and 2c. The median distance between two consecutive points on a trace, denoted as $\overline{Dis}$, is notably shorter in the Geolife and our collected datasets, at 29 meters and 6 meters, respectively. These datasets also contain a relatively higher number of locations, with 124,570 and 63,771 points. These characteristics enable Geolife and our collected datasets to have more samples in each grid on average, allowing adversaries to estimate the transition probability matrix more accurately. This leads to more precise calculations of $Pr(s|\mathcal{H}_t)$ and $P(s|o_t, \mathcal{H}_t)$, resulting in accurate predictions and a smaller Bayes risk.

The results demonstrate that traditional black-box approaches significantly underestimate the extent of information leakage, while ours provides a more accurate estimate.

6 CONCLUSION

This paper addresses a critical gap in accurately quantifying information leakage for systems with correlated data, particularly for an LBS in trace-based scenarios. Existing ML-based black-box approaches, though theoretically robust, often underestimate information leakage when dealing with correlated data in trace-based scenarios. We propose an ML-based black-box mechanism that leverages spatio-temporal correlations among locations in traces to accurately estimate Bayes risk, a key metric for assessing vulnerabilities in LBS. Thorough testing on real-world datasets confirms our model's superior accuracy and effectiveness in quantifying information leakage for LBS systems.

ACKNOWLEDGEMENTS

This work was supported in part by the US National Science Foundation under grants CNS-2245689 (CRII) and 2426470, as well as the 2022 Meta Research Award for Privacy-Enhancing Technologies.

REFERENCES

[1] M'rio S. Alvim, Kostas Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. 2012. Measuring Information Leakage Using Generalized Gain Functions. In *IEEE 25th Computer Security Foundations Symposium*. 265–279.

[2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 901–914.

[3] András Antos, Luc Devroye, and Laszlo Gyorfi. 1999. Lower bounds for Bayes error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 7 (1999), 643–645.

[4] Michele Boreale and Michela Paolini. 2014. On formally bounding information leakage by statistical estimation. In *Information Security: 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings 17*. Springer, 216–236.

[5] Konstantinos Chatzikokolakis, Tom Chothia, and Apratim Guha. 2010. Statistical measurement of information leakage. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 390–404.

[6] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. 2008. Anonymity protocols as noisy channels. *Information and Computation* 206, 2-4 (2008), 378–401.

[7] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. 2008. On the Bayes risk in information-hiding protocols. *Journal of Computer Security* 16, 5 (2008), 531–571.

[8] Giovanni Cherubin, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2019. F-BLEAU: fast black-box leakage estimation. In *IEEE Symposium on Security and Privacy (SP)*. 835–852.

[9] Tom Chothia and Apratim Guha. 2011. A statistical test for information leaks using continuous mutual information. In *2011 IEEE 24th Computer Security Foundations Symposium*. IEEE, 177–190.

[10] Tom Chothia and Yusuke Kawamoto. 2014. Statistical estimation of min-entropy leakage. *Manuscript available at http://www. cs. bham. ac. uk/research/projects/infotools/leakiest* (2014).

[11] Tom Chothia, Yusuke Kawamoto, and Chris Novakovic. 2013. A tool for estimating information leakage. In *International Conference on Computer Aided Verification*. Springer, 690–695.

[12] Tom Chothia, Yusuke Kawamoto, and Chris Novakovic. 2014. Leakwatch: Estimating information leakage from java programs. In *Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19*. Springer, 219–236.

[13] Macedo H. Barreto R. Cruz, M. and A. Guimares. 2016. GPS Trajectories. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C54S5Z.

[14] Luc Devroye, László Györfi, and Gábor Lugosi. 2013. *A probabilistic theory of pattern recognition*. Vol. 31. Springer Science & Business Media.

[15] Boris Köpf and Andrey Rybalchenko. 2013. Automation of quantitative information-flow analysis. In *International School on Formal Methods for the Design of Computer, Communication and Software Systems*. Springer, 1–28.

[16] Heiko Mantel and Henning Sudbrock. 2009. Information-theoretic modeling and analysis of interrupt-related covert channels. In *Formal Aspects in Security and Trust: 5th International Workshop, FAST 2008 Malaga, Spain, October 9-10, 2008 Revised Selected Papers 5*. Springer, 67–81.

[17] Mapbox. [n. d.]. Mapbox | Maps, Navigation, Search, and Data. https://www.mapbox.com/. Accessed: September 9, 2024.

[18] Annabelle McIver and Carroll Morgan. 2003. A probabilistic approach to information hiding. In *Programming methodology*. Springer, 441–460.

[19] Marco Romanelli, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Pablo Piantanida. 2020. Estimating g-leakage via machine learning. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 697–716.

[20] Charles J Stone. 1977. Consistent nonparametric regression. *The annals of statistics* (1977), 595–620.

[21] Vladimir Vapnik. 2013. *The nature of statistical learning theory*. Springer science & business media.

[22] Yu Zheng. 2011. T-Drive trajectory data sample. https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/ T-Drive sample dataset.

[23] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. 2008. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing*. 312–321.