Finding the Stars in the Fireworks: Deep Understanding of Motion Sensor Fingerprint

Huiqi Liu*, Xiang-Yang Li*, Lan Zhang*, Yaochen Xie*, Zhenan Wu*, Qian Dai*, Ge Chen†, Chunxiao Wan†
* School of Computer Science and Technology, University of Science and Technology of China
†Tencent OMG AdTech, P.R. China

Abstract—**With the proliferation of mobile devices and various sensors (*e.g.*, GPS, magnetometer, accelerometers, gyroscopes) equipped, richer services, *e.g* location based services, are provided to users. A series of methods have been proposed to protect the users' privacy, especially the trajectory privacy. Hardware fingerprinting has been demonstrated to be a surprising and effective source for identifying/authenticating devices. In this work, we show that a few data samples collected from the motion sensors are enough to uniquely identify the source mobile device, *i.e.*, the raw motion sensor data serves as a fingerprint of the mobile device. Specifically, we first analytically understand the fingerprinting capacity using features extracted from hardware data. To capture the essential device feature automatically, we design a multi-LSTM neural network to fingerprint mobile device sensor in real-life uses, instead of using handcrafted features by existing work. Using data collected over 6 months, for arbitrary user movements, our fingerprinting model achieves 93% F-score given one second data, while the state-of-the-art work achieves 79% F-score. Given ten seconds randomly sampled data, our model can achieve 98.8% accuracy. We also propose a novel generative model to modify the original sensor data and yield anonymized data with little fingerprint information while retain good data utility.**

I. Introduction

To provide richer services, modern mobile devices are equipped with various sensors, *e.g.*, GPS, magnetometer, accelerometers, gyroscopes. Sensor data is continuously generated and collected by service providers to support various functions like recording running traces, step count and calorie burning [1], as well as a variety of other novel uses, *e.g.*, human activities understanding and searching [2], pedestrian tracking [3] and eye gaze tracking [4], [5]. On the customer information market [6], sensor data may also be put on the shelves for further research. Despite the aforementioned attractive features, rich personal information contained in the sensor data could also pose a serious privacy threat. Traditional anonymization methods, *e.g.*, hiding user ID, querying location with encryption [7], perturbing voiceprint [8], [9] cannot effectively mitigate the risk, because each sensor has its unique physical characteristics, which can be captured as a fingerprint in its produced data. Due to strong one-to-one connection between devices and users, fingerprinting a device often implies identifying a user.

Investigating device fingerprint is important for both attack and defense. Some existing efforts have explored various methods to fingerprint different kinds of sensors for tracking users across applications [10], [11], [12]. These methods often extract dozens of pre-defined features (*e.g.*, mean, deviation, and spectral centroid) from sensor data, and use various supervised classifiers (*e.g.*, SVM, Naive-Bayes, and Multiclass Decision Tree) to fingerprint devices. Some countermeasures are also proposed, such as calibration and obfuscation, to mitigate fingerprinting. Those methods, however, have limitations in practical scenarios, and few of them achieve a systematic understanding of mobile sensor fingerprint, which makes it quite challenging to answer the following key questions.

First, what is the capacity of device fingerprint? Manufacturing imperfection makes each sensor have specific physical characteristics. In order to leverage these characteristics as fingerprint, we need to find out if the capacity of the characteristic space is sufficient to distinguish a substantially large number of devices. This cannot be answered by existing experiment results based on only dozens of devices.

Second, which features and models are better to achieve robust and efficient fingerprinting? How do the human activities affect device fingerprinting? Handcrafted features used by existing method cannot capture the essential characteristics of device fingerprint. Moreover, many of the pre-defined features, *e.g.*, *mean deviation*, and *spectral entropy*, are highly sensitive to noises like human activity, which deteriorates the robustness of fingerprint in complex real-life scenarios. For example, using 70 features, the identification F-score of [10] is about 93% when the phone has only lightly movement, however the F-score reduces to 78% when the phone moves in a moderate speed. Besides, it takes us about 5 seconds to extract those 70 features from 1 second data. To achieve robust and efficient fingerprinting, we need to extract intrinsic features of sensors when they are swallowed by user's substantial movements.

Third, how to retain utility while anonymizing sensor data? Existing countermeasures include calibration and obfuscation. Calibration can eliminate some of the errors that result from manufacturing imperfections, but many sensors, like gyroscope, are hard to calibrate manually or require specialized equipment. Obfuscation adds noises (e.g., uniform noise and Laplace noise) to the sensor readings, which reduces the fingerprinting accuracy, but also sacrifices some data utility, e.g., resulting in incorrect step count. It is challenging to design a general countermeasure to achieve a good anonymization results, as well as tradeoff between anonymization and utility for different types of sensor data.

Methodology and Contributions: To answer aforementioned challenging questions, we deeply investigate the mobile sensor fingerprint and make the following contributions.

Theoretical fingerprint capacity model: We are the first to propose a theoretical model to quantify the capacity of device fingerprint with multiple dimension features, and analyze/verify this model with a large collection of mobile device data. Our model assumes that the collection of devices' fingerprinting features follow certain distributions such as normal distribution, or uniform random distribution. We then derive the theoretical fingerprinting capacity by studying the impact of the number of features, the partition granularity of the feature space, and the number of devices to be fingerprinted.

Deep neural network based fingerprinting model: To capture the essential fingerprinting feature automatically, we design a multi-LSTM neural network to fingerprint mobile device sensors in real-life uses. This is a non-trivial task due to two reasons. First, despite the great success deep learning has achieved in computer vision, speech recognition and natural language processing, little work has applied deep learning to fingerprint sensors. It's a challenge to design a proper network structure to achieve robust fingerprinting. Second, the sensor data is sampled unevenly and extremely noisy due to arbitrary user activities. We need to carefully pre-process the raw data and pack it properly as input of the neural network. Comparing to previous work, our proposed multi-LSTM model achieves better accuracy and much stronger robustness.

Generative model based anonymizing method: We propose a novel generative model to anonymize sensor data while retain good data utility. Our method can be applied to various sensor data for real-time data release.

With users' permission, we collect motion data from 117 mobile phones, with 13 different brands devices, over more than 6 months period, and then conduct extensive evaluations. The experiments show consistent results with our capacity model. For *arbitrary* user movements, given only 1 second data, our fingerprinting model achieves 93% F-score, while the state-of-the-art work achieves 79% F-score. With only accelerometer, our model can still achieve 90.26% accuracy. If there are only 20 devices, they can be fingerprinted with 99.2% accuracy. For different devices (13 brands, 65 models in our experiment) of the same brand/model, the fingerprinting accuracy is still above 93.5%. For different devices (12 devices in our experiment) used by the same person, the accuracy reduces to 89%, due to the influence of human behavior fingerprints. Using 20s data, the top-1 accuracy is 99% and top-2 accuracy is 99.94% by voting. Using our model, we can extract fingerprint features in an unsupervised manner. It only takes 0.04ms to fingerprint 1 second data. Our anonymizing model can reduce the fingerprinting accuracy to 5% while retaining good utility with only 0.9 ms delay.

The rest of this paper is organized as follows. We review related work in Section II, and describe our methodology in Section III. In Section IV, we theoretically analyze the capacity of device fingerprint. Section V presents our neural network model for robust fingerprinting, and Section VI presents our generative model based anonymization method. Section VII reports experimental results, and Section VIII concludes the work with future work.

II. Background and Related Work

Existing efforts mainly investigated two categories of device fingerprints, software fingerprints and hardware fingerprints. Hardware fingerprints are more persistent but more challenging to characterize.

A. Software Fingerprinting

Researchers have characterized different installed softwares as fingerprints to distinguish different devices, for example, the installed device drivers [13], the performance benchmarks of JavaScript engines [14], the characteristics of 802.11 traffic [15], and the timing analysis of 802.11 probe request frames [16]. A common set of approaches collect information via browsers to generate a device's software fingerprint, such as the HTML5 canvas elements [17] and user browsing history [18]. Due to the dynamic nature of installed softwares, the software-based fingerprints usually change with time.

B. Hardware Fingerprinting

Different hardware components of mobile devices have been investigated to generate fingerprints. Wireless transmitters can be fingerprinted by radio frequency (RF)[19]. Network devices have distinguishing and stable clock skews [20], [21], which can be used for fingerprinting [22]. The source network interface card (NIC) can be identified using minute imperfections in transmitter hardware [23].

Sensor Fingerprinting. Hardware characteristics can be used to identify devices, and then users. These characteristics are caused by manufacturing differences or manufacturing imperfections. In theory, most sensors have some sort of measurable bias. For example, accelerometer, gyroscope, magnetometer and ambient light sensors generate data with linear bias, and GPS sensor has clock skew imperfection. Stisen et al. [24] investigate mobile sensing heterogeneities for HAR (human activity recognition). Zhou et al. [25] extract features from audio pieces and conducted fingerprinting by feature matching. Das et al. [26] extract rich acoustic features and applied traditional classification algorithms to fingerprint. Accessing to the microphones and speakers require obvious user permission, while *motion sensors (e.g., accelerometer and gyroscope) can be accessed without requiring any user permission*, which raises potential threats to privacy. Dey et al. [12] use feature extraction and Bagged Decision Trees to generate accelerometer fingerprint. Bojinov et al. [11] have analyzed common mobile device sensors along with their imperfections. Das et al. [10] use 70 temporal and spectral features of gyroscopes and accelerometers to track mobile users through web browsers. They also propose calibration and obfuscation as two straightforward defenses. However, the calibration requires specialized equipments, while obfuscation reduces the utility of the motion sensors.

As a summary, existing works have tried around 100 handcrafted features working along with different classifiers to fingerprint sensors, and evaluated the accuracy with dozens of devices.

III. Methodology and Problem Scope

In this work, towards a systematic deep understanding of mobile sensor fingerprint, we thoroughly investigate the following challenging issues.

- **Capacity of device fingerprint.** To theoretically analyze the fingerprint capacity, we propose a multidimensional Balls-into-Bins model, taking the devices as balls and the partitions of the multidimensional hardware feature space as bins. Leveraging the statistic results of our 117 diverse mobile devices dataset, our theoretical model shows the fingerprint capacity as the devices number and sensor category grow. (See Section IV.)
- **Robust fingerprinting deep neural network.** To achieve robust device fingerprint in practical uses, where the subtle hardware characteristics are swallowed by environment noises and arbitrary user activities, we design a series of deep neural networks to automatically extract essential fingerprinting features which outperforms existing handcrafted feature based methods. Moreover, we reveal several insights about influence factors of device fingerprinting. (See Section V and Section VII.)
- **Defense model retaining utility.** To anonymize sensor data as well as retain the data utility, we propose a novel generative model consisting of an encoder and a decoder, which makes minimal modifications to the sensor data to remove fingerprint information in a real-time manner. (See Section VI.)

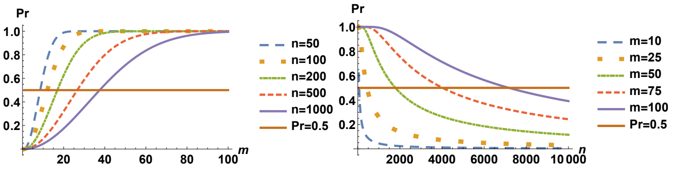
Our theoretical capacity model can be adopted for analyzing fingerprints of diverse mobile devices. The proposed fingerprinting and anonymizing models can be applied for various sensor data in time series form. In our data-driven analysis (in Section V and Section VII), we take motion sensors (namely, accelerator and gyroscope) as examples. Because motion sensors can be accessed without requiring any obvious user permission, which raise high privacy threats as well as great challenges for fingerprinting due to the rich user activity information in the sensor data. Different from the previous work, which places devices on a flat surface (i.e., a static scenario) or holds in hand with slightly movement, we consider more practical scenarios, where users can perform arbitrary activities, e.g., browsing, walking and running.

IV. Understanding Capacity of Fingerprint

In this section, we propose a theoretical capacity model to understand the capability of hardware fingerprinting and conduct rigorous analysis considering multidimensional features.

A. Brief Introduction to Onboard Sensors

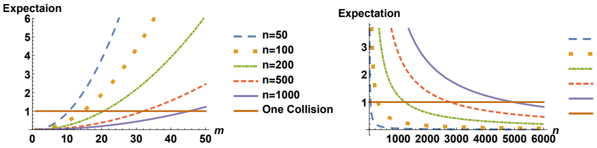
Substantial efforts have been devoted to modeling MEMS-based motion sensor noise. For the most common noises, Gabrielson [27] models the mechanical thermal noise as function of absolute temperature and the damping coefficient. Djuric [28] derives more complex noise models combining mechanical thermal noise with electrical noise sources. These noises are the characteristics (like damping coefficient) of the motion sensors. They are slightly different from each other due to the heterogeneities of the manufacturing procedure, thus, forming fingerprints for sensors.



(a) Fixed feature space size n.

(b) Fixed device number m.

Fig. 1: The probability $Pr(Col_1^m)$ vs device # m and feature space size n when the feature is uniformly distributed.



(a) Fixed feature space size n.

(b) Fixed device number m.

Fig. 2: Expected # of collisions $\mathbb{E}(C)$ vs device # m and feature space size n when the feature is uniformly distributed.

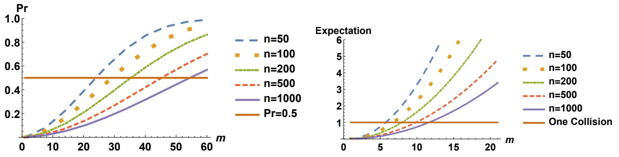
B. Capacity Model of Fingerprinting

We propose a multidimensional Balls-into-Bins model to analyze the capacity of device fingerprint. Here we take m devices as balls and n partitions of the hardware feature space as bins, and throw m balls into n bins. Intuitively, when a ball falls into a bin, it means this device possesses a specific feature. When more than one balls fall into the same bin, a collision indicates these devices have the same feature, that is they cannot be distinguished by this type of feature.

1) One-dimension Feature Space: Let's start with one-dimension feature space. The feature of each device's sensor is independent of that of other devices' sensors. Sensor features are continuously distributed, and we discretize the feature space into n partitions.

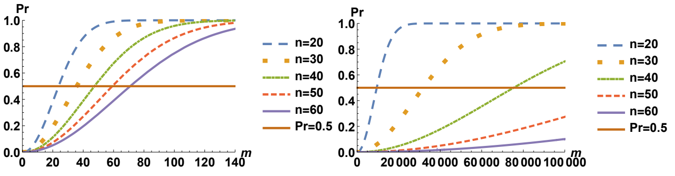
Uniformly distributed feature. First, we assume sensor features follow a uniform distribution. In this case, the model is that each ball is independently thrown into a random bin following the uniform distribution. So the probability that a ball falls into any bin is $1/n$. Let Col_i^j denote the event that there exist collisions for the balls whose indices are within the range $[i, j]$. $\mathbb{E}(C)$ denotes the expectation number of collisions. Note that we count collisions over distinct pairs, that is, if three balls fall into one bin, three collisions are counted. The probability that there exist collisions is (proof omitted due to space limitation) $Pr(Col_1^m) = 1 - Pr(\neg Col_1^m) = 1 - \frac{(n-1)!}{n^{m-1}(n-m)!}$. The expected number of collisions is $\mathbb{E}(C) = \frac{m(m-1)}{2n}$.

We now analyze how the device number m and feature space size n change the collision probability $Pr(Col_1^m)$ and collision number $\mathbb{E}(C)$. Obviously, increasing device number m brings larger collision probability (Fig. 1a) and more collision devices (Fig. 2a), while increasing feature space size n reduces collision probability (Fig. 1b) and collision number (Fig. 2b). Given one-dimension feature space with limited size, e.g., 100 bins, even a small set of devices, e.g., 20 devices, there is a more than 80% collision probability (Fig. 1).



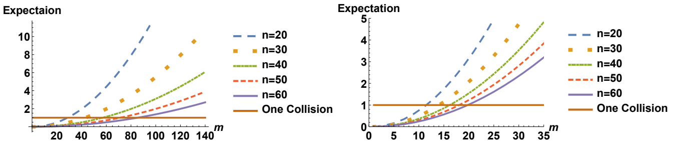
(a) The probability $Pr(Col_1^m)$ (b) The expected collision $\mathbb{E}(C)$

Fig. 3: The collision probability $Pr(Col_1^m)$ and the number of expected collision $\mathbb{E}(C)$ changes against m and n when the feature is a binomial distribution $B(n-1, 1/2)$.



(a) Uniform distribution, 2D bins (b) Uniform distribution, 6D bins

Fig. 4: The probability $Pr(I_1^m)$ versus m, n and k when features are uniformly distributed (here we suppose each dimension has the same number of bins).



(a) Uniform distribution, 2D bins (b) Normal distribution, 2D bins

Fig. 5: The expected number of indistinguishable balls $\mathbb{E}(I)$ versus m, n and k when the feature space is 2-dimension (here we suppose each dimension has the same number of bins).

Normally distributed feature. Second, we consider sensor features follow a normal distribution. To describe the feature distribution over discrete bins, we use a discrete distribution, binomial distribution, to approximate normal distribution. For m balls and n bins, each time a ball is independently thrown into a bin follow the binomial distribution $B(n-1,p)$, which means that a ball falls into the x-th bin with probability $\binom{n-1}{x-1}p^{x-1}(1-p)^{n-x}$. In this case, the probability that there exist collisions is $Pr(Col_1^m) = 1 - \sum_{\substack{\{r_1,r_2,\cdots,r_m\} \\ \subset\{0,1,\cdots,n-1\}}} \prod_{i=1}^{m}\binom{n-1}{r_i}p^{r_i}(1-p)^{n-1-r_i}$, where r_i is the index of the bin ($0 \leq r_i \leq n-1$) which the i-th ball falls into. The expected number of collisions is $\mathbb{E}(C) = \frac{1}{2}m(m-1)\sum_{i=0}^{n-1}\left(\binom{n-1}{i}p^i(1-p)^{n-1-i}\right)^2$.

Fig 3a and Fig. 3b plot the collision probability and expected collision number. Similar to the uniform distribution case, larger m and smaller n increase the collision significantly. Differently, in the normal distribution case, the collision happens with a much higher probability, which means it is much more difficult to distinguish devices in this case.

2) Multi-dimension Feature Space: In practice, multiple features can be utilized to distinguish mobile devices. Now we consider the multi-dimension Balls-into-Bins problem. Here we suppose each dimension is independent to give the upper bound of fingerprint capacity. In practice, different sensors' noise, e.g, gyroscope and accelerometer, can be considered as independent[1]. Two indistinguishable balls must collide in all dimensions, i.e. in the same high-dimensional grid. Let I_i^j denote the k-dimension collision event that there exist indistinguishable balls in the k-dimension feature space. Suppose there are k-dimension bins with bin sizes $\{n_1, n_2, \cdots, n_k\}$.

Uniform distributed feature. In uniform distribution case, the probability that indistinguishable balls exist is $Pr(I_i^m) = 1 - Pr(\neg I_i^m) = 1 - \binom{n}{m}m! * \left(\frac{1}{\prod_{i=1}^{k} n_i}\right)^m$, where $n = \prod_{i=1}^{k} n_i$. The expected number of indistinguishable balls is $\mathbb{E}(I) = \frac{m(m-1)}{2\prod_{i=1}^{k} n_i}$.

Fig. 4 plots the probability of k-dimension collision. Compared to the one-dimension case, two independent feature dimensions, e.g., two independent sensors such as accelerometer and gyroscope, significantly reduce the indistinguishable probability. With limited bin size on each dimension, e.g., 100, given 20 devices, the indistinguishable probability drops to less than 5% (compared to 80% in the one-dimension case). 50 bins on each dimension can reduce the expected collision number of 50 devices to lower than 1 (Fig. 5a).

Normally distributed feature. Similarly, we use binomial distribution $B(n-1,p)$ as the approximation of normal distribution, and the probability for a ball falls into the k-th bin in the i-th dimension is $\binom{n_i-1}{x-1}p_i^{x-1}(1-p_i)^{n_i-x}$, where p_i is the binomial distribution parameter for the i-th dimension. So the expected number of indistinguishable balls is $\mathbb{E}(I) = \frac{1}{2}m(m-1)\prod_{i=1}^{k}\sum_{j=0}^{n_i-1}\left(\binom{n_i-1}{j}p_i^j(1-p_i)^{n_i-1-j}\right)^2$. Fig. 5b plots the expected number of indistinguishable devices for 2-dimension feature space. Compared to the uniform distribution case, the fingerprint capacity decreases, while increasing feature dimensionality still significantly enlarges the capacity.

V. Robust Device Fingerprinting

In this section, we explore features for robust device fingerprinting. Existing works have proposed dozens of handcrafted features. The state-of-the-art work [10] uses 70 temporal and spectral features of sensor data as device fingerprints. Most of these features are highly sensitive to large noises like human activities. In practical sensor data, subtle hardware fingerprints are usually swallowed by substantial movement signals, e.g., walking and running. To investigate the impact of activities on fingerprinting, we implement the proposed method in [10], and test the accuracy and efficiency on our data collected from 117 different devices. Our sensor data are collected in both static scenario, where the phone is placed flat on a surface, and highly dynamic scenario, where the user holds the phone and performs arbitrary activities such as walking and gaming. It takes us about 5 seconds to extract 70 features from 1 second data. As shown in Fig. 6, in the static scenario, using a set of handcrafted features can achieve 93% F-score for 97 devices given 1 second data[2]. When it comes to the highly dynamic

[1] Note that three axes of one sensor are physically separated, but the feature dependency among them is till unexplored.

[2] As reported in [10], it can achieve 96% F-score given 5 seconds data of 96 devices. With our dataset, [10] achieves 97.5% given 5 seconds data for 97 devices, which is consistent with the reported result.

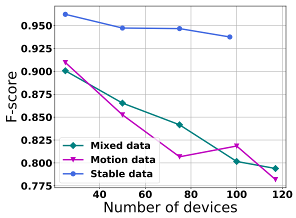


Fig. 6: Fingerprinting F-score of the state-of-the-art method [10] in both static and dynamic scenario.

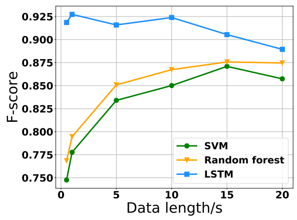


Fig. 7: Fingerprinting F-score of different models for different input data lengths in dynamic scenario.

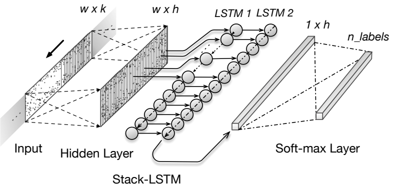


Fig. 8: The network structure of LSTM fingerprinting model.

scenario, the F-score reduces significantly to 77% for 117 devices. And the F-score declines significantly as the device number increases.

A. Neural Network based Fingerprinting

Our results reveal that, it is quite challenging to achieve robust device fingerprinting using real-life sensor data. In order to capture the inherent hardware features automatically, we propose to use deep neural network for fingerprints extraction. Facing various challenges, we explore a variety of deep neural networks and design a Long Short Term Memory network (LSTM) model which is suitable for sensor feature extraction. Raw data are carefully processed before being fed into the model. Our proposed model achieves high accuracy in highly dynamic scenarios.

1) Multi-LSTM fingerprinting model.: LSTM network, as a variant of RNN introduced by Hochreiter and Schmidhuber [29], is capable of learning long-term dependencies like the fingerprinting information in sensor data. We design a Multi-LSTM structure as shown in Fig. 8. More specifically, the input are $w \times k$ sequences, where w is the data length (e.g., $w = 100$ when input 1-second data with 100Hz sample rate), k is the channel number (e.g., $k = 3$ for data from three axes of the accelerometer). w is correlated to the fingerprinting delay in a realtime system. k is the dimension number in our theoretical capacity model in Section IV, if k input channels are independent. Through one hidden layer, data are fed into a multi-LSTM structure in order to extract persistent features. Then it outputs a size h vector as the input of the second hidden layer, which is then connected with the soft-max layer.

As a comparison, we also design and build a CNN (Conventional Neural Network) model taking raw sensor data as input, a CNN model taking the data processed by Short-Time Fourier Transform (STFT) as input. Based on our extensive evaluation, we find that the multi-LSTM model significantly outperforms the other two models.

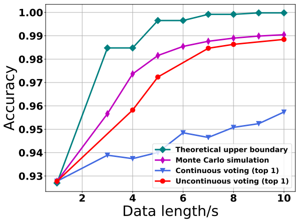


Fig. 9: Fingerprinting accuracy with Majority Voting.

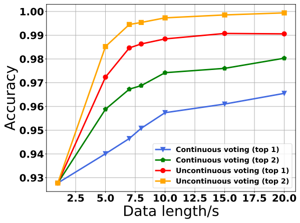


Fig. 10: Fingerprinting accuracy with Majority Voting.

Unsupervised fingerprinting. Based on our LSTM model, unsupervised fingerprinting can also be conducted by treating the penultimate layer (the size h vector) of our network as a device's fingerprint feature. Given unlabelled data from K devices, we can extract the fingerprint feature of each piece of data, and apply unsupervised learning, e.g., k-means clustering, on all data pieces to cluster data from the same device together. Furthermore, given only a few labelled data for each device, we can identify a large-scale of unlabelled data, which enables more stronger and more practical unsupervised attack without requiring many labelled training data.

2) Data Pre-processing: For each motion sensor, i.e., accelerometer or gyroscope, three data sequences are simultaneously generated with with timestamps by three axes. So, in our experiments, we obtain 6 data sequences from two motion sensors. Each sequence can be a channel of the neural network input. However, they are generated with unstable time intervals, which depends on the schedule of the mobile operating system according to the real-time system load. Hence, we conduct piece-wise cubic Hermite interpolation to obtain equally spaced data sequences as the inputs of neural networks. We also divide the continuous sensor data into small sequences of the same length w.

3) Evaluation Metric and Model Comparison: To evaluate the effectiveness of fingerprinting models, we use two metrics: accuracy and F-score. As a multi-class classification task, the model accuracy is defined as the proportion of correction predictions, F-score gives a tradeoff between precision and recall, which is defined as F-score $= \frac{2 * Presicion * Recall}{Precison + Recall}$.

Using our large dataset collected from 117 diverse devices in real-life scenarios, we conduct comprehensive evaluation on our neural networks as well as the state-of-the-art methods [10]. We report detailed evaluation results and analysis in Section VII. As a summary, Fig. 7 shows the fingerprinting accuracy of our model compared with models in [10] for highly dynamic scenarios given different input data lengths w. It reveals that handcraft features based models (SVM and Random Forest) can only achieve $74\% \sim 87\%$ F-score due to the large noise caused by human activities. Our multi-LSTM approach achieves 93% F-score with only 1 second sensor data.

4) Majority Voting Strategy: To further increase the accuracy, we apply the Majority Voting Strategy: suppose a fingerprinting model for t-second input data has been trained to achieve an accuracy p. Then theoretically, given a piece of $s \times t$-second data, where $s \in \mathbb{N}$ and $s > 1$, by majority voting we can achieve the following accuracy:

$$Accuracy(s) = \begin{cases} \sum_{i=m}^{s} C_s^i p^i (1-p)^{s-i} - \frac{1}{2} C_s^m p^m (1-p)^{s-m}, \\ \qquad\qquad \text{if } s \text{ is even and } m = s/2 \\ \sum_{i=m+1}^{s} C_s^i p^i (1-p)^{s-i}, \\ \qquad\qquad \text{if } s \text{ is odd and } m = (s-1)/2 \end{cases}$$

Using our dataset we conduct Monte Carlo simulation to simulate the performance of majority voting strategy on real data[3]. Fig. 9 shows the theoretical and real data based Monte Carlo simulation results. With majority voting, given 10 seconds data, the theoretical accuracy can achieve 99.9%, and the simulation results can exceed 99%. Then we evaluate the voting strategy for two different real-life scenarios: (1)*continuous voting*: an attacker obtains a piece of continuous data of a device and divided them into small pieces for voting; (2)*uncontinuous voting*: an attacker obtains pieces of uncontinuous data collected at different time, and uses them for voting. The uncontinuous voting attack can compromise privacy-preserving methods which release scattered data pieces to protect user privacy. As shown in Fig. 9 and Fig. 10, the continuous voting can achieve 96% top-1 accuracy and 97.4% top-2 accuracy given 10 seconds data, while the uncontinuous voting can achieve 98.8% top-1 accuracy and 99.7% top-2 accuracy given 10 pieces of 1 second data.

VI. Anonymizing Sensor Data

Facing fingerprinting attack with high accuracy, there is an urgent demand for effective countermeasures to protect user privacy. To anonymize sensor data, we propose a novel generative model, which eliminates the fingerprints effectively while minimizing data utility loss.

A. Anonymization Model

The model is composed of a generator and a discriminator. The basic idea is to train a generator, which takes the original data as input and outputs de-fingerprinted data to fool a well-trained discriminator, e.g., our multi-LSTM fingerprinting model. That is, the discriminator takes the de-fingerprinted data as input and outputs incorrect labels. The goal is to maximize the randomness of the discriminator's output labels with minimal data modification. In this way, our model anonymizes the input data while retains the data utility. Our model structure is demonstrated in Fig. 11. The generator is an auto-encoder containing two convolutional layers. For the discriminator, we use our multi-LSTM model since it outperforms other fingerprinting models.

When training the generator, instead of using correct labels, we match each piece of sensor data with a random label to force the generator to modify the original data to anonymized data. Simultaneously, the generator also tries to minimize differences between anonymized data and original data. We define the loss function as $loss_g = cross\_entropy(y, y') + max\{0, \|x - x'\|_{L2} - \epsilon\}$, where y denotes the random labels we match to the data, y' denotes the outputs of discriminator during training, x and x' denote the original data and de-fingerprinted data, and ϵ denotes an acceptable error of the generator. By reducing the loss during training, we can control similarity level between original and anonymized data, as well as the anonymization level. Our model takes less than 1 ms to anonymize 1 second data. So it can serve as a new feature in the future mobile operating systems.

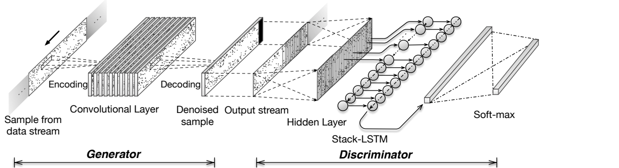


Fig. 11: The network structure of data anonymization model.

B. Evaluation Metrics

Anonymization Effect. It is hard to tell whether a countermeasure can defend against all fingerprinting methods. A fair metric is to use the state-of-the-art fingerprinting methods [10] and our multi-LSTM based fingerprinting as attack models, and measure the fingerprinting accuracy of these models on the anonymized data. The lower the accuracy, the better the anonymization effect.

Utility. It is essential that the anonymized data should not scarify the data utility. We measure data utility from two perspectives: (1) *Modification distance*: We use the L_2 distance between the original data sequence and anonymized data sequence to measure the extent of the modification, which should be as small as possible. (2) *Data usage*: We use the output of motion sensor based applications to test the anonymized data, e.g., we use the output of a pedometer application to check if the step count of the anonymized data is correct.

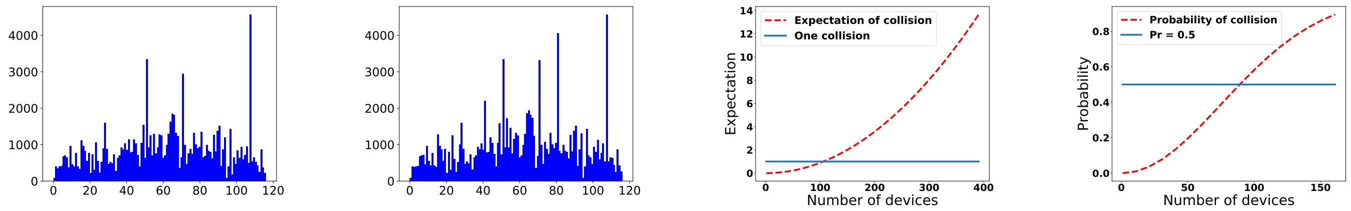
VII. Data-driven Analysis

To deeply investigate, we collect a large highly diverse real-life dataset over 6 months, and conduct a series of data analysis and evaluation based on the dataset. Based on extensive experimental results, we further explore the multi-dimension device features and the fingerprint capacity, and reveal more insights about different influence factors (e.g., human and hardware model) on fingerprinting. We also prove the effectiveness and efficiency of our fingerprinting and anonymizing models.

A. Data Collection

With users' permission, we collect motion sensor (accelerometer and gyroscope) data from total 117 mobile phones with 13 different brands (Tab. I) when the users performed arbitrary movements. The sensor data is sampled at $60Hz \sim 200Hz$, and each data record is annotated with the device id and user id. We divide continuous sensor data (more than 150 hours data) into $1 \sim 20$-second segments as our dataset. For training and testing, we randomly split our dataset into two parts: 80% for training and the other 20% for testing. To

[3] First, we use 1-second pieces of training data to train our multi-LSTM model. Then, during each step in the simulation, we randomly select a label category of test data and randomly choose a piece of s-second data from this category. Next, we split the data into s 1-second pieces, and use the trained model to predict the label of each piece. Using the s labels for majority voting, we obtain the final predicted label for this s-second data.



(a) Distribution in accelerometer dimension

(b) Distribution in gyroscope dimension

(c) Expectation number of indistinguishable devices

(d) The probability for indistinguishable devices

Fig. 12: Fingerprinting capacity in two dimension

prevent bias of evaluation results caused training/testing data selection, for every experiment we repeat random training data selection ten times and report the average results.

Brand	Proportion	Brand	Proportion
Apple iPhone	30.77%	Nexus	3.42%
Apple iPad	16.24%	Vivo	2.56%
Xiaomi	9.40%	Nubia	1.71%
Huawei	12.82%	LeShi	1.71%
Samsung	7.69%	LG	0.85%
OnePlus	5.98%	Lenovo	0.85%
MeiZu	5.13%	OPPO	0.85%

TABLE I: Details of the device models.

B. Explore Multi-dimension Features

We firstly explore the multi-dimension feature space of the device fingerprint, since feature dimensionality significantly affects the capacity (see Section IV-B).

Two sensors: accelerometer and gyroscope. Here, two sensors (accelerometer and gyroscope) can be treated as independent dimensions for device fingerprint. First, we conduct LSTM-based fingerprinting on each sensor's data separately. As shown in Tab. II, given only 1 second data, to identify 117 devices we can achieve 90.26% accuracy for accelerometer and 68.62% accuracy for gyroscope. With majority voting of 20 seconds data, the accuracy can be improved to 93.99% and 80.03% respectively. By fusing predict results of two sensors using confidence boosting strategy[4], our model achieves 91.41% accuracy, which is close to the accuracy of fusing two sensors' data together as input into one network.

	Accuracy	Majority voting (20s)
Accelerometer	90.26%	93.99%
Gyroscope	68.62%	80.03%
Fusing results of 2 sensors	91.41%	

TABLE II: Fingerprinting accuracy for 117 devices of a single sensor and fusing 2 sensors given 1 second data.

Six axes of accelerometer and gyroscope. Each sensor has three axes, which are supposed to be independent because they are physically separated in the MEMS-based model. We fingerprint three axes of each sensor separately. Given 1 second data of one axis of accelerometer and gyroscope, our model can respectively achieve 72% and 53% accuracy for 117 devices (Tab. III). Fusing the results of 6 axes together by confidence boosting strategy produces 87.01% accuracy.

Senor	Axes	Accuracy
Accelerometer	ax	72.56%
	ay	72.96%
	az	43.57%
Gyroscope	gx	52.92%
	gy	54.02%
	gz	53.85%
Fusing results of 6 axes		87.01%

TABLE III: Fingerprinting accuracy for 117 devices of a single axis and fusing 6 axes given 1 second data.

Note that, though different sensors and different axes are physically independent, we cannot claim their fingerprint data are absolutely independent, because there may be some correlation among their fingerprints due to the environment influence like temperature and humidity. This may explain that, prediction based on fused data achieves better accuracy (92%) than fusing prediction results of different sensors/axes.

C. Fingerprint Capacity Analysis

After exploring multi-dimension features, we are ready to map the Balls-into-Bins model with the fingerprinting scenario, and to estimate the capacity bound based on the investigation of real-life data. In our model, the "balls" are devices. "bins" are partitions of a feature space with distinguishable resolution. Since our experiment shows 92% accuracy for 117 devices, we can claim with confidence that there are at least 117 distinguishable partitions in the feature space. For the different feature dimensions, they are required to be independent to each other. Based on the analysis in Section VII-B, here we model feature dimensions as two sensors, which can be assumed independent. This two dimension model with 117 "bins" for each dimension gives the lower bound of fingerprint capacity. For the ball's distribution probability on each dimension, we use the frequency that devices are classified into one "bin" (i.e., label). Fig. 12a and Fig. 12b illustrate the distribution probability of 117 bins for accelerometer and gyroscope respectively. Here, we assume all balls have the same distribution in each dimension. Now we have obtained all parameters of our capacity model based on real data. Fig. 12 shows the estimated device fingerprint capacity. For less than 90 devices, the expected collision is less than 1, and the probability of collision is less than 50%. For more than 110 devices, there is expected to be at least one collision. This is consistent with our experiment result that for 117 devices given 20 seconds data of each devices the top-2 fingerprinting accuracy is 98%.

[4] The confidence boosting strategy works as follows. For a prediction, a single predictor has different confidence scores for each label, which is the last layer's values of the network. We add up the confidence scores of multiple predictors and take the maximum of them as the final predicted label.

Device placement scenario	Method	# of device	Metrics	Remarks
On flat surface	[10]	93	96% F-score	[10]'s result
On flat surface	Our LSTM model	97	97% Accuracy, 97% F-score	1 second, Our dataset
Arbitrary human motion	[10]	117	77% Accuracy, 78% F-score	[10]'s method and our 1 second dataset
	Our LSTM model	117	91% Accuracy, 91% F-score	1 second, Our dataset
Mixed data	[10]	117	80% Accuracy, 79% F-score	[10]'s method and our 1 second dataset
	Our LSTM model	117	92% Accuracy, 93% F-score	1 second, Our dataset
	LSTM model	117	96% Accuracy (top-1), 97.4% Accuracy (top-2)	10 seconds, Our dataset
	with continuous voting	117	96.5% Accuracy (top-1), 98% Accuracy (top-2)	20 seconds, Our dataset
	LSTM model	117	98.8% Accuracy (top-1), 99.7% Accuracy (top-2)	10 1-second pieces, Our dataset
	with un-continuous voting	117	99% Accuracy (top-1), 99.9% Accuracy (top-2)	20 1-second pieces, Our dataset

TABLE IV: Results in different scenarios and comparison with the state-of-the-art work [10].

D. *Fingerprinting Accuracy in Different Scenarios*

Now we conduct extensive experiments using our dataset to evaluate the performance of our LSTM model in different scenarios and compare it with the state-of-the-art work [10].

Static vs. Dynamic. First, we consider the static scenario versus dynamic scenario. As summarized in Tab. IV, in the static scenario, our model and [10] both achieve a high accuracy, which are 97% and 96% F-score respectively. But when it comes to the highly dynamic scenario, given 1 second data, the F-score of [10] drops to 78% while the F-score of our model is 91%. Mixing the static and dynamic data together to recover the real-life scenario, our model achieves 93%, while F-score of [10] is only 79%. The results show that our model is more robust to large noises like human activity.

Majority voting. As presented in Tab. IV, leveraging continuous majority voting, given 10 seconds data, we can achieve 96% top-1 accuracy and 97.4% top-2 accuracy. For our uncontinuous majority voting, given 10 pieces of 1 second data, we can achieve 98.8% top-1 accuracy and 99.7% top-2 accuracy. Fig. 9 and Fig. 10 plot the voting accuracy increasing with the length of data. Given 20 pieces of 1 second data, we can achieve 99.9% top-2 accuracy.

Influence of brands and models. Here, we are interested in the question "is it more difficult to distinguish devices of the same brand/model? " The devices in our experiment have 13 brands and 65 models. As reported in Tab. V, the device distinguishability is slightly different across different brands or models. For the same brand or model, devices can still be identified with a high accuracy, e.g., 93% for 36 iPhones, 93% for 12 iPhone6 and 92% for 9 iPhone7.

Influence of human. The collected data carry both hardware information and human behavior information. To answer how the human behavior fingerprints affect device fingerprinting, we asked one volunteer to use 12 different devices freely. As presented in Tab. V, in this case, our model can identify the 12 devices with 89% accuracy, which is slightly worse than the overall accuracy, while using random forest in [10] achieves 71% accuracy. So, the human factor indeed increases the difficulty of motion sensor based device fingerprinting. How to separate the human fingerprints and device fingerprints remains a challenging question.

Influence of device number and training data size. Given the fingerprint capacity, more devices imply more collisions in the feature space. Our evaluation results show that as the device number increases from 20 to 117, the accuracy declines from 99.20% to 92%. Obviously, more training data samples produce stronger fingerprinting models.

Granularity	device #	user #	Accuracy	Remarks
All devices, all users	117	77	92%	Whole dataset
One brand, different devices	55	36	94%	Apple products
	36	29	93%	iPhone
	19	17	96%	iPad
	11	10	94%	Xiaomi
	15	14	91%	Huawei
One model, different devices	12	10	93%	iPhone6
	9	8	92%	iPhone7
	8	7	95%	iPad Air2
One user, different devices	12	1	89%	Our LSTM model
	12	1	71%	Random Forest

TABLE V: Fingerprinting accuracy in different granularity given 1 second data.

Unsupervised fingerprinting. Furthermore, we treat the penultimate layer of our LSTM network as a device fingerprint feature, and apply k-means clustering ($k = 117$) on features of 100,730 1 second data pieces of 117 devices. The Adjust Rand Index and the Adjust Mutual Information between the clustering result and the ground truth are 81% and 88% respectively. This result also proves the effectiveness of our fingerprinting model, more importantly, presents the chance to conduct unsupervised fingerprinting using the extracted features.

E. *Defenses Performance*

We evaluate our anonymization model, and compare it with methods in [10], which directly add noises to the data. Since in dynamic scenario the sensor data has large variance, adding Laplace noise and white noise according to the methods in [10] will cause large L_2 distance. So, here we compare the uniform noise in [10] with our model. We use the fingerprinting accuracy of our LSTM model, SVM and Random Forest models in [10] as the anonymization effect metric, and L_2 distance and step count results as the data usage metric. There is a tradeoff between anonymization effect and data utility.

Tab. VI presents step count results[5] on sensor data anonymized by our model and uniform noise. The results in Tab. VI and Fig. 14 show that when L_2 distance < 6, our model reduces the fingerprinting accuracy to 19% and cause no error to the step count; while uniform noise causes 0.51 deviation and only reduces the fingerprinting accuracy to 32%. When L_2 distance < 30, our model causes less than 0.89 deviation and reduces the accuracy to 5%; while uniform noise causes 1.78 deviations and reduces the accuracy to 10%. Thus, our model achieves both better anonymization effect and

[5] In our experiment, each of volunteers takes 30 steps with the mobile phones and repeats it for 20 times.

Our model	$L_2(\approx)$	0	6	9	15	20	30
	Mean	30	30	29.97	29.97	29.68	29.38
	Stdev	0	0	0.21	0.34	0.63	0.89
Uniform noise	$L_2(\approx)$	0	6	12	18	24	30
	Mean	30	29.86	29.52	28.77	29.02	29.27
	Stdev	0	0.51	0.76	1.31	1.62	1.78

TABLE VI: Step count results of anonymized data by our model and existing method [10].

data utility. Fig. 14 confirms that our model works effectively against different fingerprinting models.

F. Efficiency

We run our experiments on a server with 12 Intel Core i7-5930K 3.50GHz CPUs and 1 Titan X (Pascal) GPU. It takes us around 2 hours to train the LSTM model. For the testing, it takes 0.04 ms and around 0.9 ms to fingerprint and anonymize the fingerprint respectively for 1 second sensor data. As a comparison, the average time of extracting the 70 features proposed in [10] from 1 second data is 5.16s. The results show that it is very costly to extract a large set of temporal and spectral features from the data. Instead, our models can provide both realtime fingerprinting and anonymizing.

VIII. Conclusion

The raw sensor data, containing device-dependent noises, has been demonstrated to be an effective fingerprint of mobile devices. In this work, we showed that a few (less than 100) data samples collected from the motion sensors are enough to uniquely identify the source mobile device. We designed a multi-LSTM neural network framework to fingerprint mobile device sensor in real-life uses. Our system achieves 93% fingerprinting F-score given only one second data even users are doing arbitrary movement, and the accuracy improves to 98.8% when we have 10 seconds data. We then proposed a novel generative model to yield anonymized data with little fingerprint information while retain good data utility. Several interesting directions are left for future investigation: integrating other sensor data to define better device/human ID for identifying users, improving accuracy and reducing requested training for better forensics usage.

Acknowledgement

Xiang-Yang Li and Lan Zhang are the contact authors. The research is partially supported by the National Key R&D Program of China 2017YFB1003000, China National Funds for Distinguished Young Scientists with No. 61625205, NSFC No. 61572281, No. 61472218, No. 61520106007, Key Research Program of Frontier Sciences, CAS, No. QYZDY-SSW-JSC002, NSF ECCS-1247944, and NSF CNS 1526638.

References

[1] Guo Xiaonan, Liu Jian, and Yingying Chen. Fitcoach: Virtual fitness coach empowered by wearable mobile devices. In *INFOCOM*. IEEE, 2017.

[2] C. Liu, L. Zhang, Z. Liu, K. Liu, X.-Y. Li, and Y. Liu. Lasagna: towards deep hierarchical understanding and searching over mobile sensing data. In *MobiCom*. ACM, 2016.

[3] Y. Jiang, Z. Li, and J. Wang. Ptrack: Enhancing the applicability of pedestrian tracking with wearables. In *ICDCS*. IEEE, 2017.

[4] L. Zhang, X.-Y. Li, W. Huang, K. Liu, S. Zong, X. Jian, P. Feng, T. Jung, and Y. Liu. It starts with igaze: Visual attention driven networking with smart glasses. In *MobiCom*. ACM, 2014.

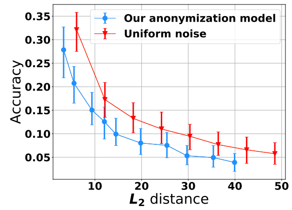


Fig. 13: L_2 distance and anonymization effect (measured by LSTM model).

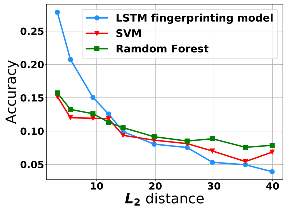


Fig. 14: Anonymization effect measured by different fingerprinting models.

[5] Z. Li, M. Li, P. Mohapatra, J. Han, and S. Chen. itype: Using eye gaze to enhance typing privacy. In *INFOCOM*. IEEE, 2017.

[6] Catherine Dwyer. Privacy in the age of google and facebook. *IEEE Technology and Society Magazine*, 30(3):58–63, 2011.

[7] X.-Y. Li and T. Jung. Search me if you can: privacy-preserving location query service. In *INFOCOM*. IEEE, 2013.

[8] J. Qian, F. Han, J. Hou, C. Zhang, Y. Wang, and X.-Y. Li. Towards privacy-preserving speech data publishing. In *INFOCOM*. IEEE, 2018.

[9] J. Qian, H. Du, J. Hou, L. Chen, T. Jung, X.-Y. Li, Y. Wang, and Y. Deng. Voicemask: Anonymize and sanitize voice input on mobile devices. *arXiv preprint arXiv:1711.11460*, 2017.

[10] A. Das, Nikita Borisov, and Matthew Caesar. Tracking mobile web users through motion sensors: Attacks and defenses. In *NDSS*, 2016.

[11] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh. Mobile device identification via sensor fingerprinting. *arXiv preprint arXiv:1408.1416*, 2014.

[12] S. Dey, N. Roy, W. Xu, R. Roy Choudhury, and S. Nelakuditi. Accelprint: Imperfections of accelerometers make smartphones trackable. In *NDSS*, 2014.

[13] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *Usenix Security*, 2006.

[14] Keaton Mowery, Dillon Bogenreif, Scott Yilek, and Hovav Shacham. Fingerprinting information in javascript implementations. *W2SP*, 2011.

[15] Jeffrey Pang, Ben Greenstein, R. Gummadi, Srinivasan Seshan, and David Wetherall. 802.11 user fingerprinting. In *MobiCom*. ACM, 2007.

[16] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee. Identifying unique devices through wireless fingerprinting. In *WiSec*. ACM, 2008.

[17] Keaton Mowery and Hovav Shacham. Pixel perfect: Fingerprinting canvas in html5. *W2SP*, 2012.

[18] Peter Eckersley. How unique is your web browser? In *PETS*, pages 1–18. Springer, 2010.

[19] Neal Patwari and Sneha K Kasera. Robust location distinction using temporal link signatures. In *MobiCom*. ACM, 2007.

[20] Vern Paxson. On calibrating measurements of packet transit times. In *ACM SIGMETRICS*, volume 26, pages 11–21. ACM, 1998.

[21] S. Moon, P. Skelly, and Don Towsley. Estimation and removal of clock skew from network delay measurements. In *INFOCOM*. IEEE, 1999.

[22] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy. Remote physical device fingerprinting. *TDSC*, 2(2):93–108, 2005.

[23] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *MobiCom*. ACM, 2008.

[24] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *SenSys*. ACM, 2015.

[25] Zhe Zhou, Wenrui Diao, Xiangyu Liu, and Kehuan Zhang. Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound. In *CCS*. ACM, 2014.

[26] Anupam Das, Nikita Borisov, and Matthew Caesar. Do you hear what i hear?: fingerprinting smart devices through embedded acoustic components. In *CCS*. ACM, 2014.

[27] Thomas B Gabrielson. Mechanical-thermal noise in micromachined acoustic and vibration sensors. *IEEE transactions on Electron Devices*, 40(5):903–909, 1993.

[28] Zoran Djurić. Mechanisms of noise sources in microelectromechanical systems. *Microelectronics Reliability*, 40(6):919–932, 2000.

[29] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.