Design and Large-Scale Evaluation of WiFi Proximity Metrics

Aymen Fakhreddine
IMDEA Networks Institute
& University Carlos III of Madrid
Madrid, Spain
aymen.fakhreddine@imdea.org

Nils Ole Tippenhauer
Singapore University of Technology and Design
Singapore
nils_tippenhauer@sutd.edu.sg

Domenico Giustiniano
IMDEA Networks Institute
Madrid, Spain
domenico.giustiniano@imdea.org

***Abstract*—We study the problem of deriving proximity metrics based on WiFi fingerprints without the need of external sensors and access to the locations of APs. Applications that benefit from proximity metrics are movement estimation of a single node over time, WiFi fingerprint matching for localization systems and attacks on privacy. Using a large-scale, real-world WiFi fingerprint data set consisting of 200,000 fingerprints resulting from a large deployment of wearable WiFi sensors, we show that metrics from related work perform poorly on real-world data. We analyze the cause for this poor performance, and show that imperfect observations of APs in the neighborhood are the root cause. We then propose improved metrics to provide such proximity estimates, without requiring knowledge of location for the observed AP. Our metrics allow to derive a relative distance estimate based on two observed WiFi fingerprints. We demonstrate that their performance is superior to the related work metrics.**

I. Introduction

In recent years, mobile smart devices have become ubiquitous, e.g., smart phones, personal health devices, smart watches, and other smart home appliances. Low cost requirements and integration with legacy networks lead to wide adoption of IEEE 802.11-based wireless communication (WiFi). Many such devices, and devices that are part of the so-called Internet-of-Things (IoT), interact with their physical environment and physically close communication partners either directly or via cloud platforms.

To determine the physical locations of such devices, WiFi-localization based approaches have become widespread. Typically, the mobile device collects a wireless fingerprint at their current location. This consists of set of MAC addresses of nearby Access Points (APs), and their received signal strength indicator called RSSI. This fingerprint data is sent to a third-party *cloud service* (such as Google, Apple, Skyhook, etc.), that estimates the device's location. The motivation for using a third-party cloud service is that fingerprint data is noisy, and it requires a large set of measurements from a large number of users and access to a large database of positions of APs in order to provide high-to-medium accuracy level.

In this work, we look at the problem of WiFi proximity metrics in positioning systems, i.e., *metrics that allow to estimate the spatial correlation or physical distance between two WiFi fingerprints*. Such metrics can be valuable in a number of scenarios:

- Measurement sanitization for user-assisted WiFi localization database building [1], [2];
- Fingerprint ambiguity estimation in an unknown environment [3];
- Detection of co-location for mobile applications on mobile devices (as attack on users' privacy, see [4] for Bluetooth).

There are three main challenges to address to effectively solve the problem of WiFi proximity metrics.

Challenge 1: Design of proximity metrics without using any external sensors. To the best of our knowledge, metrics that do not require any external sensors have received limited attention in related work. Yet, the availability of a solution in this space would allow a widespread adoption to any WiFi transceiver, from low-end IoT device to more powerful smartphone device.

Challenge 2: Metrics that can cope with probabilistic observations. Fingerprints might *not* contain information of all APs in range. There are two reasons:

- storage constraints might limit the number of APs stored for each fingerprint;
- collisions, time-out, and channel hopping during WiFi scanning can prevent deterministic observations of nearby APs.

Storage of WiFi fingerprints can require relatively large amount of memory, if the MAC address (6 Bytes) and an RSSI value (1 Byte) is stored for each AP. Given the growing density of WiFi APs, single scans for neighboring APs can easily return 30 and more results. Storing an unlimited number of APs per fingerprint can thus violate memory constraints.

Challenge 3: Metrics that do not have access to large databases of known locations of APs. In this work, we are interested in metrics that do not require prior knowledge of the locations of APs in the environment. In particular, large-scale records of AP locations are proprietary data owned by few companies, and not accessible by the public.

Finally, related work lacks of *large scale real-world data evaluation of proximity metrics*, which are needed to assess the soundness of the proposed solution.

Our main contributions are listed in what follows:

- We perform an extensive evaluation of a novel Jaccard Index-based metric and two prior work metrics over two main datasets: i) an artificial dataset based on simplified propagation models and perfect knowledge of the true

locations, and ii) a large-scale, real-world WiFi fingerprint data set consisting of $200,000$ fingerprints resulting from a large deployment of wearable WiFi sensors [5].
- We identify key drawbacks of all three metrics using our real-world dataset, analyze the causes and propose a model to explain the issue. Further analysis of our dataset confirms that our model closely matches observed data.
- Based on our error model, we propose three improved distance metric definitions. Our proposed metrics only require two WiFi fingerprints readings, and enable mobile devices to compute the results i) without continuous requests to the third-party cloud service, ii) without disclosing the location to the cloud service and to neighbor nodes, and iii) with limited requirements of local storage and low computational and implementation complexity.

This work is structured as follows. In Section II, we introduce the system model, metrics from related work and the datasets used (real and simulated data). In Section III, we introduce the detailed problem, our first attempts at addressing it, and investigations into mismatch between performance on simulated data versus real data. To address performance drop with real data, we introduce improved metrics Section V and show that they perform well in both settings. We conclude in Section VI.

II. Background

A. System Model and WiFi Fingerprints

In this work, we focus on IEEE 802.11b (which we will refer to as WiFi), but general principles hold for other wireless standards as well. We consider a node with wireless transceiver that is capable of observing the presence of WiFi APs and their RSSI values. To simplify the discussion, we assume that the infrastructure is static, and the node is moving over time. We also assume that omnidirectional antennas are used. The density of APs in range can vary from 0 to more than 40.

The node gathers measurements of WiFi fingerprints over time. Then, the node uses the proximity metric to find an estimate of the moved distance for consecutive fingerprints. We use the following notation in this work to refer to WiFi fingerprints. We define n as our node of interest, a set $\mathbb{A}$ of all APs in the target area, and $\mathbb{N} \subset \mathbb{A}$ as the subset of nearby APs within $\mathbb{A}$ that are in range of node n. We use $|\mathbb{N}|$ to indicate the cardinality of a set, i.e. the count of distinct items in the set. For integers $|x|$ is the absolute value of x.

Let $o_i = (\text{RSSI}_i, \text{MAC}_i)$ denote the AP observation of n for AP i. If i is not in $\mathbb{N}$, then $\text{RSSI}_i = 0$. Then, $F = \{o_i\}, \forall i \in \mathbb{N}$ is a WiFi fingerprint which corresponds to the list of observations of currently neighboring WiFi APs. In case of multiple fingerprints, we will use a notation of F_a with $\mathbb{N}_a$ as set of neighbors.

B. Related Work Metrics

In the following, we use $\mathbb{N}_a, \mathbb{N}_b$ as set of neighboring APs of fingerprint F_a, F_b, respectively. The observation o_{ai} refers to AP $i \in \mathbb{N}_a$ in fingerprint F_a. We consider the following two metrics, MetricE and MetricM, from related work. We present them only for the sake of comparing them to the ones we design in this work.

MetricE (Euclidean Distance): In [6], the authors discuss RSSI proximity metrics. They propose to compute the Euclidean distance between the RSSI vectors $\{\text{RSSI}_{ai}\}_{i \in \mathbb{N}_a \cap \mathbb{N}_b}$ and $\{\text{RSSI}_{bi}\}_{i \in \mathbb{N}_a \cap \mathbb{N}_b}$ from the set of APs $\mathbb{N}_a \cap \mathbb{N}_b$ present in both fingerprints F_a and F_b:

$$m(F_a, F_b) = \sqrt{\sum_{i \in \mathbb{N}_a \cap \mathbb{N}_b} \left[\text{RSSI}_{ai} - \text{RSSI}_{bi}\right]^2} \quad (1)$$

MetricM (The "Manhattan" distance): In [6], the authors also propose to use the Manhattan distance that refers to the sum of the absolute differences instead of the Euclidean one.

$$m(F_a, F_b) = \sum_{i \in \mathbb{N}_a \cap \mathbb{N}_b} |\text{RSSI}_{ai} - \text{RSSI}_{bi}| \quad (2)$$

C. NSE Dataset

Our evaluation uses a large-scale real-world dataset, collected as part of the National Science Experiment (NSE) project in Singapore. We now briefly introduce the SENSg devices that are used to gather the dataset.

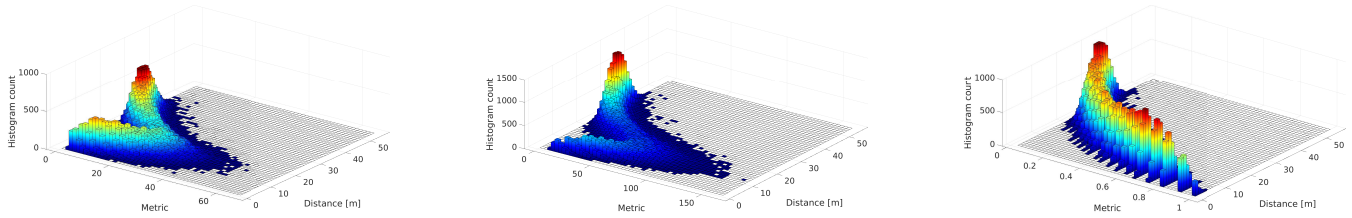
SENSg sensors: A total of 50,000 devices are produced and used by students at schools. The students wear the devices for a week or longer, and collect data about their daily life. The data is automatically uploaded to a cloud platform, and made available to the students to analyze. The devices are called SENSg [5], and they record WiFi fingerprints (as defined in this work) every 12 seconds. Using a third party API, the WiFi fingerprints are mapped to location estimates after the data is uploaded to the cloud. The SENSg devices store only the 20 APs with highest RSSI per fingerprint. Storing the MAC address and received signal strength for each AP requires 7 Bytes, so a fingerprint with 20 observed APs is 140 Byte large.

Measurements and data gathering: The real world datasets used in this work are gathered by students during the NSE project. In particular, the dataset used in this work is a subset of all fingerprints taken. For all fingerprints used, we also have a location estimate by the third party API. Location provided by the third party API is subject to accuracy errors, as in typical cloud-based location based-systems.

D. Simulation setup

In addition to the real world dataset, we generate an artificial dataset with $200,000$ fingerprints as in the real measurement data. The fingerprints are randomly distributed in an area which is roughly equivalent to the area covered in the real-world dataset. We generate this second dataset to have a better control over noise and other factors that lead to unexpected behavior in the observed APs, and their RSSI values. In addition, this allows us to know the true location of fingerprints, which will be exploited in the evaluation.

To simulate path loss $L(d)$ and the resulting RSSI, we use the 802.11 propagation model E presented in [7] that takes into account LOS (Line-of-Sight) and NLOS (Non-Line-of-Sight) channels, breaking point distance in which the attenuation



(a) MetricE (b) MetricM (c) MetricJ

Fig. 1: Simulation-based evaluation of initially proposed vs prior work metrics.

slope drastically changes because of the scattering effect, log-normal shadowing, fading and other physical channel phenomena. $L(d)$ consists in the sum of a freespace component $L_{FS}(d)$ whose slope depends on the distance d in comparison to the aforementioned breakpoint distance d_{BP} and a shadow fading component $S_F(d)$ that accounts for the large scale scattering.

$$L(d) = \begin{cases} L_{FS}(d) + S_F(d) & d \leq d_{BP} \\ L_{FS}(d_{BP}) + 10\alpha_2 \log_{10}\left(\frac{d}{d_{BP}}\right) + S_F(d) & d > d_{BP} \end{cases} \tag{3}$$

α_2 is the attenuation slope after d_{BP} while $L_{FS}(d)$ is computed with a lower attenuation slope α_1 before d_{BP}.

$$L_{FS}(d) = 10\alpha_1 \log_{10}\left(\frac{4\pi f d}{\mathfrak{c}}\right) \tag{4}$$

f denotes the frequency and $\mathfrak{c}$ the speed of light in vacuum. The shadow fading component is modeled by a log-normal distribution centered in zero with a standard deviation σ_{SF}. The constants for model E as introduced in [8] are the following: $d_{BP} = 20$ meters, $\alpha_1 = 2$, $\alpha_2 = 3.5$ and $\sigma_{SF} = \{3, 6\}$.

III. Proximity Metrics for WiFi Fingerprints

In this section, we summarize our problem statement, and then present a number of candidate proximity metrics that will be evaluated later.

A. Problem Statement

Our goal is to provide a metric $m(F_a, F_b)$ able to estimate the expected spatial correlation between two fingerprints. The metric will be optimized for accuracy in the estimate, and low computational cost. Intuitively, that metric should be 1 if two fingerprints are taken at the exact same location, and 0 if they are completely uncorrelated (e.g., no single access point was observed by both fingerprints).

Consequently, we define the proximity metric as a function $m(F_a, F_b) = y$ between the fingerprints F_a and F_b, with $0 \leq y \leq 1$.

B. Problems with Metrics from Related Work

We evaluate MetricE and MetricM from related work over our simulated data set. We summarize the results in Figure 1 (a) & (b). From our study, we find that both have several issues: i) they are not normalized, but return a value ≥ 0, with smaller values indicating proximity; ii) they do not work well for distances larger than 15 meters, for which only few mutual APs are observed. We note that the original work used the metrics to find closest matching fingerprint pairs, and not to estimate exact distances. As it can be seen in Figure 1, for distances greater than 15 meters, the metric score is decreasing on average, leading to values indicating closer proximity. One explanation for that behavior could be that with increasing distance between the fingerprint locations, fewer mutual APs are observed and the sum of RSSI differences will decrease with increasing distance. This is a practical issue, as low metric score could either indicate close proximity, or distances of more than 15 meters. To the best of our knowledge, such issues were not discussed before in related work.

C. Using Jaccard-Index to Reward Mutual Observations

Based on the above findings, intuitively, a metric should score high if a large fraction of the APs observed in two fingerprints are shared, i.e. mutually observed. Based on that, our initial proposal is to use a metric that contains a factor relating to the Jaccard Index [9], defined as:

$$m(F_a, F_b) = \frac{|\mathbb{N}_a \cap \mathbb{N}_b|}{|\mathbb{N}_a \cup \mathbb{N}_b|} \tag{5}$$

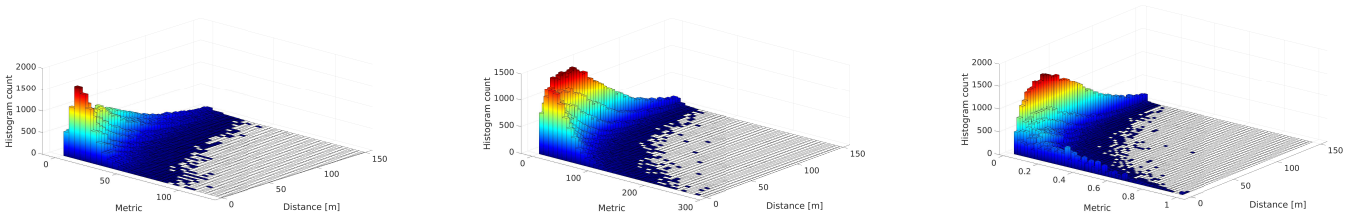
In other terms, for two fingerprints, the Jaccard Index is a ratio of number of mutually observed APs, divided by the total number of observed APs. To simplify the discussion, our first proposed metric MetricJ is exactly the Jaccard Index, see Eq. 5.

We note that MetricJ is range-free, i.e. RSSI values are not directly used, while MetricE and MetricM measure the similarity of RSSI values of APs that are observed in both fingerprints. Nevertheless, we evaluate the use of MetricJ as distance metric, and discuss the results. We later discuss the use of the Jaccard Index as factor to scale other range-based metrics.

We evaluate the performance of MetricJ compared to MetricE and MetricM with two sets of data: simulated fingerprints, and a large set of real-world data (see Section II).

D. Simulation-based Evaluation

We start with a simulation-based evaluation, using the dataset described in Section II-D. To evaluate the quality of the metric, we compare the metric score to the true distance between the locations at which the fingerprints are taken. We compute the Spearman correlation [10] between the two values to obtain a quantitative result. Unlike the widely used Pearson correlation, the Spearman correlation evaluates the monotonic



(a) MetricE (b) MetricM (c) MetricJ

Fig. 2: Real data-based evaluation of initially proposed vs prior work metrics.

Metric	Dataset	Correlation	Improv. Correlation
MetricJ	Artificial	0.91	0.91
MetricE	Artificial	0.59	0.94
MetricM	Artificial	0.72	0.94
MetricJ	Real-world	0.46	0.49
MetricE	Real-world	0.31	0.53
MetricM	Real-world	0.41	0.54

TABLE I: Summary of Spearman correlation values of original and improved metrics on our datasets.

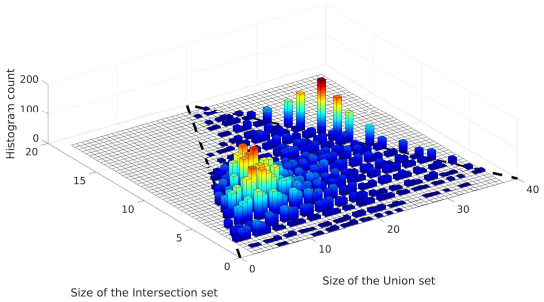


Fig. 3: Jaccard index components: the number of mutually observed APs versus the total number of APs observed in two fingerprints that are at most 1 meter distance apart.

relationship and it is then a better fit for non-linear correlation studies. In addition, the Spearman correlation is proven to be robust in the sense of being resistant to outliers [10].

From visual inspection of our simulation results presented in Figure 1, MetricJ is superior to the two related work metrics. This is confirmed by the correlation scores of 0.59 (MetricE), 0.72 (MetricM) and 0.91 (MetricJ) (see Table I).

E. Real-World Data-based Evaluation

For this dataset (introduced in Section II-C), we do not have accurate ground truth locations available. Instead, we will use the distance between the fingerprints based on the third party's localization result (which itself is a noisy estimate) for the performance evaluation. We evaluate around $200,000$ fingerprints located all around Singapore.

As it can be observed in Figure 2, all three metrics perform much worse than expected on the real dataset. We note that for MetricJ, a value of 1 should correspond to a small distance, while for the other two metrics, smaller values mean shorter distances. For all three metrics, distances of $<$10m generally do have expected metric scores, but the Spearman correlation between distance and score is not very strong: 0.46 (MetricJ), 0.31 (MetricE), 0.41 (MetricM). While MetricJ's correlation score is still better than the other two metrics, MetricJ has very low values even for short distances, and high scores are almost never reached. Clearly, the performance predicted based on simulations is not achieved when the evaluation is done using our real-world dataset. Surprised by those results, we set out to investigate the cause, and possible mitigations.

IV. Analysis of MetricJ

We now present our analysis of the causes for the bad performance of MetricJ on the real data set. We start by analyzing metric scores for fingerprints that are estimated to be taken at the same locations. We find that MetricJ scores are low because the actual number of mutually observed APs is much smaller than expected compared to the total number of APs observed in both fingerprints.

To confirm that finding, we select $11,433$ fingerprint tuples that are within respective estimated distance according to the third-party cloud service of 1 meter or less. We then compute the numerator and divisor of the Jaccard index and show the results in Figure 3. Points on the dashed diagonal lines indicate that both fingerprints contain the same set of observed APs. We expected to see the same or similar set of APs in both fingerprints, with diagonal up to a divisor score of 20 (c.f. Section II-C), and only few cases of divisor scores higher than 20. Instead, the results show that there are *two distinct clusters*: around (7,12) and (14,28). Given the construction of MetricJ, those clusters would lead to metric scores of around 0.5, although scores of 1 would be expected given the short distance between the fingerprints. The question is now: *why there are so few APs mutually observed for fingerprints taken at the same location?*

A. Probabilistic Observations of APs

Our hypothesis is that the fingerprint collection process on the devices suffers from a probability e to miss a nearby AP completely (in addition to expected RSSI variations). More formally, e would lead to the following expected ratio of mutually observed APs versus total APs in the shared neighborhood:

(a) Probability of mutual observation: $P_m = (1-e)^2$.
(b) Probability of having at least one observation: $P_o = 1-e^2$.

This results in the following maximal Jaccard Index MJI for n APs in the shared neighborhood:

$$\text{MJI} = \frac{P_m * n}{P_o * n} = \frac{P_m}{P_o} = \frac{1-e}{1+e}$$

In other words, an effective upper bound for our metric score depends on e, even if fingerprints are taken from the same location.

B. Probability e of missing a nearby AP

First cluster. As expressed above, we have identified two clusters. Our guess is that the cluster around (14,28) is likely due to the limited cache of 20 best APs per fingerprint.

Second cluster. For the cluster around (7,12), we speculate that this is due to the limited scanning capabilities of WiFi chipsets. WiFi nodes scan each WiFi channel for APs for a limited time (e.g. 120-180 ms). As WiFi APs typically transmit beacons every 100 ms, channel congestions may cause that beacons transmitted during the time spent on that channel are lost due to the likelihood of beacons collisions and hidden nodes. We leave a more detailed investigation for future work.

Based on the clusterization above, we analyze the real-world dataset to attempt to give an estimated value of e due to caching and channel congestion. We compare the number of APs within a range of 20 meters from two fingerprints separated by a maximum distance of 1 meter to the total number of APs present in both fingerprints. The 20 meters radius we choose corresponds to the breakpoint distance after which the signal is more likely to suffer from an attenuation of a higher slope factor due to obstacles.

$$e = \frac{\mathbb{E}(\text{\# of APs in FPs of d} \leq \text{1m})}{\mathbb{E}(\text{\# of APs within 20m})} = \frac{17.16}{67.62} = 0.25 \quad (6)$$

V. Improved Metrics

A. Candidate Metrics

We now present a set of candidate metrics to improve on MetricJ with regards to imperfect observations of APs, and to leverage RSSI values. In general, the metric should incorporate two properties: a) a factor reflecting on similarity in observed APs in both fingerprints, and b) counterbalance as much as possible the effect of the previously computed probability e.

We now explore options for both a) and b), leveraging our insights as presented in Section V-B. In particular, we will convert the a)-related components of MetricE and MetricM into factors that range from 0 (for no similarity) to 1 (for maximal similarity) by imposing a limit on the RSSI difference for mutually observed APs, and a scaling factor s. In addition, we will introduce the same b)-related factor for all three metrics discussed.

Similarity score for mutually observed APs: In MetricE and MetricM, differences of RSSI values for mutually observed APs are computed, of which the squared or absolute value is then summed up. The resulting values are in practice between 0 and 250 (see Figure 2). We now replace that with a construction that returns values between 0 and 1 for each mutually observed AP. Together with the normalization factor, that will lead to an overall metric score between 0 and 1 (where larger scores indicate higher spatial correlation). We define that similarity score for two fingerprints F_a, F_b as follows:

$$\delta(o_{ai}, o_{bi}) = 1 - \frac{|\text{RSSI}_{ai} - \text{RSSI}_{bi}|}{max(|\text{RSSI}_{ai}|, |\text{RSSI}_{bi}|)} \quad (7)$$

Counterbalance the effect of the probability e: In MetricJ, normalization (through the divisor) was based on the size of the union set of observed APs. As discussed in Section IV, the observed value of that size (and size of the intersection set) is biased by e. The idea we adopt limits the number of considered mutually observed APs for metrics computations. As the main peak in Fig. 3 is around (7,12), we propose to set a maximum number of APs, $\#^{\text{APs}}_{\text{max}} = 7$ over which the similarity score defined in Eq. 7 is computed, rather than the whole set $\mathbb{N}_a \cap \mathbb{N}_b$. These $\#^{\text{APs}}_{\text{max}}$ APs are chosen as those in $\mathbb{N}_a \cap \mathbb{N}_b$ with the lowest absolute difference of corresponding RSSIs between both FPs (Fingerprints) $|\text{RSSI}_{ai} - \text{RSSI}_{bi}|$, we denote this set as: $\mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})$

MetricJ+i (improved MetricJ): In this metric, we extend MetricJ with the upper bound on the intersection size to compensate e as discussed above. The resulting metric is:

$$m(F_a, F_b) = \frac{\sum_{i \in \mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})} 1}{\#^{\text{APs}}_{\text{max}}} \quad (8)$$

Here, we choose to represent $|\mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})|$ as $\sum_{i \in \mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})} 1$ to highlight similarities to the other two improved metrics.

MetricE+i (improved MetricM): The next metric is the Manhattan distance-based MetricM, improved with our new similarity function $\delta(F_a, F_b)$ and the compensation for the effect of the probability e.

$$m(F_a, F_b) = \frac{\sum_{i \in \mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})} \delta(o_{ai}, o_{bi})}{\#^{\text{APs}}_{\text{max}}} \quad (9)$$

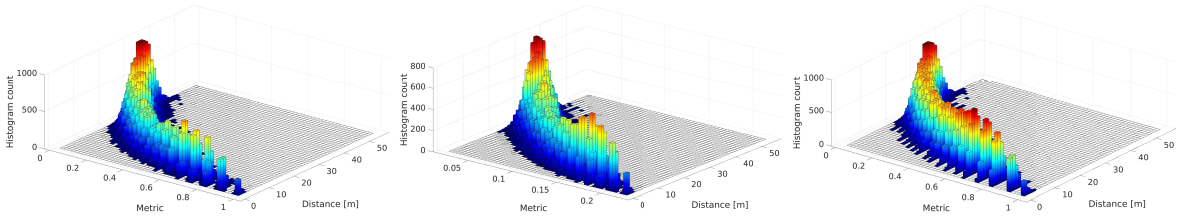
MetricM+i (improved MetricE): The last metric is MetricE with our new similarity function $\delta(F_a, F_b)$ and the compensation for the effect of the probability e.

$$m(F_a, F_b) = \frac{\sqrt{\sum_{i \in \mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})} \left[\delta(o_{ai}, o_{bi})\right]^2}}{\#^{\text{APs}}_{\text{max}}} \quad (10)$$

B. Evaluation

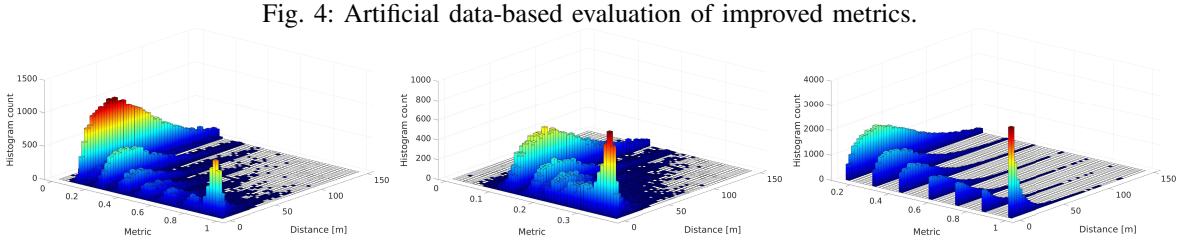
We evaluate the metrics with simulated and real data, with results shown in Figure 5. The Spearman correlation scores are as follows: 0.49 (MetricJ+i), 0.54 (MetricM+i), and 0.53 (MetricE+i). All correlation results are summarized in Table I. We conclude that both proposed improvements (similarity score and e-effect compensation) together improve the previously discussed metrics, with all three metrics performing similarly in terms of Spearman correlation.

The Spearman correlation is a good indicator of metrics performance. However, it does not capture other aspects necessary to compare the proposed metrics, such as the *robustness*. We introduce robustness as a performance measure in



(a) MetricM+i (b) MetricE+i (c) MetricJ

Fig. 4: Artificial data-based evaluation of improved metrics.



(a) MetricM+i (b) MetricE+i (c) MetricJ+i

Fig. 5: Real data-based evaluation of improved metrics

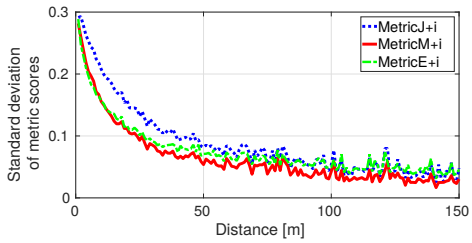


Fig. 6: The standard deviation of metrics scores computed for FPs separated by distances ranging from 1 to 150 m (real measurement dataset).

the sense that for a specific physical distance between FPs, the corresponding metric score should be as consistent as possible across different conditions and scenarios. For the real measurement dataset, we show in Fig. 6 the standard deviation of metrics scores vectors of size 3000 for each distance from 1 to 150 meters with a ± 0.1 m margin. Naturally, the lower this standard deviation is, the more robust is the corresponding metric. Although the 3 metrics compared in this Section show more or less similar Spearman correlation coefficients, Fig. 6 demonstrates that MetricM+i is the most robust one compared to MetricE+i and MetricJ+i.

VI. Conclusion

In this work, we discussed proximity metrics for WiFi fingerprints that do not need external sensors and do not have access to the locations of APs. Using real data from a large dataset as well as simulated data, we have shown that metrics proposed in related work do not perform as expected in noisy real datasets, and we have proposed a range of alternatives. We have also shown that access points might be missed in real environments with some probability e and proposed an upper bound for metric scores as a function of e. Based on those insights, we have improved our proposed metrics. The best performing metric (MetricM+i) has resulted in a Spearman correlation score of 0.54 with the real dataset and 0.94 with the artificial one.

Acknowledgments

The authors would like to thank Ryan Ong and Sandra Siby for their work on metric calculations for the NSE dataset in 2015. The authors thank the National Research Foundation (NRF) of Singapore for providing access to the National Science Experiment data. This work has been funded in part by the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919).

References

[1] A. Goswami, L. E. Ortiz, and S. R. Das, "Wigem: A learning-based approach for indoor localization," in *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '11. New York, NY, USA: ACM, 2011, pp. 3:1–3:12. [Online]. Available: http://doi.acm.org/10.1145/2079296.2079299

[2] S. He and S.-H. G. Chan, "Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490.

[3] W. Sun, J. Liu, C. Wu, Z. Yang, X. Zhang, and Y. Liu, "Moloc: On distinguishing fingerprint twins," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*. IEEE, 2013, pp. 226–235.

[4] A. Korolova and V. Sharma, "Cross-app tracking via nearby bluetooth devices," in *Proceedings of PrivacyCon*, 2017.

[5] E. Wilhelm, S. Siby, Y. Zhou, X. J. S. Ashok, M. Jayasuriya, S. Foong, J. Kee, K. Wood, and N. O. Tippenhauer, "Wearable environmental sensors and infrastructure for mobile large-scale urban deployment," *Sensors*, vol. 16, no. 22, pp. 8111–8123, November 2016.

[6] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2. IEEE, 2000, pp. 775–784.

[7] V. Erceg, L. Schumacher, P. Kyritsi, A. Molisch, D. Baum, A. Gorokhov, C. Oestges, Q. Li, K. Yu, N. Tal *et al.*, "TGn channel models," May 2004.

[8] E. Perahia and R. Stacey, *Next generation wireless LANs: 802.11 n and 802.11 ac*. Cambridge university press, 2013.

[9] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, no. 5323, pp. 34–35, 1971.

[10] C. Croux and C. Dehon, "Influence functions of the spearman and kendall correlation measures," *Statistical methods & applications*, vol. 19, no. 4, pp. 497–515, 2010.