# DevLoc
## Seamless device association using light bulb networks for indoor iot environments

Haus, Michael; Ott, Jorg; DIng, Aaron Yi

**DOI**
10.1109/IoTDI49375.2020.00030

**Publication date**
2020

**Document Version**
Accepted author manuscript

**Published in**
Proceedings - 5th ACM/IEEE Conference on Internet of Things Design and Implementation, IoTDI 2020

**Citation (APA)**
Haus, M., Ott, J., & DIng, A. Y. (2020). DevLoc: Seamless device association using light bulb networks for indoor iot environments. In *Proceedings - 5th ACM/IEEE Conference on Internet of Things Design and Implementation, IoTDI 2020* (pp. 231-237). Article 9097526 (Proceedings - 5th ACM/IEEE Conference on Internet of Things Design and Implementation, IoTDI 2020). IEEE. https://doi.org/10.1109/IoTDI49375.2020.00030

Accepted Author Manuscript. Link to published article (IEEE): https://doi.org/10.1109/IoTDI49375.2020.00030

# DevLoc: Seamless Device Association using Light Bulb Networks for Indoor IoT Environments

Michael Haus
*Technical University of Munich*
haus@in.tum.de

Jörg Ott
*Technical University of Munich*
ott@in.tum.de

Aaron Yi Ding
*Delft University of Technology*
aaron.ding@tudelft.nl

***Abstract*—For indoor IoT environments, spontaneous device associations are of particular interest where users establish a connection in an ad-hoc manner to enable serendipitous interaction. For instance, between a user's personal device and devices the user encounters in the surrounding environment. Our system for device grouping named DevLoc takes advantage of ubiquitous light sources around us to perform continuous device grouping based on the similarity of light signals. To control the spatial granularity of user's proximity, we provide a configuration framework to manage the lighting infrastructure through customized visible light communication. We support two modes of device associations to achieve a binding between different entities: device-to-device and device-to-area allowing either proximity-based or location-based services. Our device grouping includes several methods where in general the machine learning based signal similarity performs best compared to distance and correlation metrics.**

***Index Terms*—Mobile ad hoc networks, Network services, Ubiquitous and mobile devices, Similarity measures, Machine learning approaches**

## I. Introduction

We witness a proliferation of wireless devices, such as laptops, mobile phones, tablets, IoT boards, and more. Their wireless capabilities enable flexible formation of ad-hoc groups. Dynamic group association opens up new opportunities for users to spontaneously share resources or information. We aim to support two different types of proximity applications targeted for end users and Internet of Things (IoT). We highlight three use cases for proximity-based, user-oriented applications [1]: 1) Alice is a tourist, rides on the subway and wants to ask locals for the best way to the museum, 2) Bob, a student lands at his college airport and wants to check if anyone from his college is currently at the airport and can give him a ride to campus, and 3) Carol is a manager who wants to automatically record who is present at her daily meetings or share data during a meeting with colleagues and customers. On the other hand, we highlight two use cases for proximity-based IoT applications: 1) IoT boards upload location-tagged data that allows data filtering and data merging from multiple devices at the same area and 2) location-based access policy for consumer smart home platforms [2], e.g., Amazon Echo or Google Home. Therefore, we focus on proximity which has been identified as a group association technique where devices find one another when they are brought within a close distance or a dedicated space [3]. Proximity identifies potential group members and device association refers to the technique that connect group members.

Our system named DevLoc uses visible light signaling for continuous device grouping because light sources are ubiquitous around us ensuring practicality. DevLoc relies on Wi-Fi as the primary communication means in combination with visible light. Since visible light does not pass through opaque objects, it is a good candidate to realize distance-bounding wireless communication compared to the electromagnetic waves of Wi-Fi which easily penetrate physical barriers. Via visible light we achieve more fine-granular device associations based on light bulb coverage which are impossible to recognize with propagating Wi-Fi. On this basis, we can automatically generate meaningful data sharing policies among device groups to define with whom sharing or aggregating data. We offload the task to specify data sharing policies to lower communication layers which are typically handled as part of the application layer in wireless systems used today. Compared to Wi-Fi or other communication technologies, we justify the use of visible light by enabling or automating specific use cases based on the unique characteristic to be sensitive to spatial barriers. This compensates the downsides of visible light such as lack of hardware support at mobile devices. We adopt a master-slave mechanism for light bulbs to minimize the adaption of existing lighting infrastructure for DevLoc.

In contrast to existing systems for device grouping and to facilitate applications with different spatial expansion of proximity, we provide a complete framework to manage the lighting infrastructure and control the spatial granularity of device grouping. We overcome the main disadvantage of location tags [1] that users have no control over the spatial granularity of proximity where the notion of neighborhood is entirely dependent on the type of location tag. Therefore, we enrich the lighting infrastructure by adding light signaling to the widely used Light-Emitting Diode (LED) lamps in residential and office settings. Our custom light bulbs integrate illumination with visible light signaling to automatically link physically nearby devices via the similarity of light patterns. Besides that, we preserve user privacy by comparing low-level context such as surrounding light signals instead of higher-level context like ambient sound. Users may not be comfortable with the idea of sharing context with strangers, even if doing so increases their access to timely and relevant information [4]. Our generated light patterns are ephemeral, unpredictable nonces associated with a location like a shared pool of entropy between all users at a given location at a given time [1]. The two key properties of light patterns for device grouping are: 1) reproducibility that two measurements at the same place and time yield tags matching with high probability, and 2) unpredictability that an adversary not at a specific place or time is unable to produce a tag that matches the tag measured at that location at that time.

Our key contributions are summarized as follows:

1) We design and develop DevLoc for efficient device grouping based on visible light signaling. We can adapt the lighting infrastructure based on a custom light bulb to control the spatial granularity of user's proximity.
2) We evaluate the physical channel of VLC regarding a real-world deployment of DevLoc. In addition, we perform two simulations for static device-to-device grouping and dynamic device-to-area grouping where we analyze the performance of signal comparison methods.

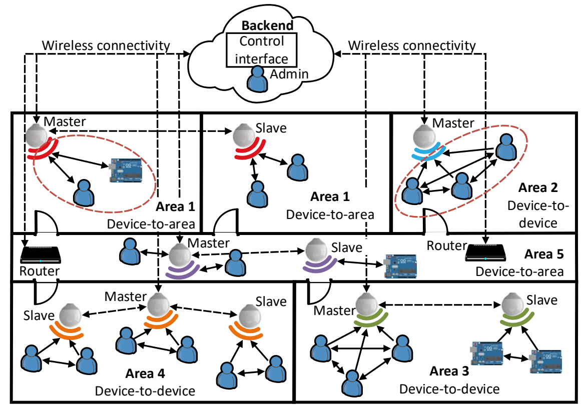
## II. Related Work

DevLoc is related to the areas of device pairing, device coupling, device association, and device grouping. Our device grouping is a guidance technique based on proximity in the real world without human interaction. For an overview, the work of [5] categorizes techniques for device associations in the following way: 1) guidance techniques where users act in the real world in order to connect devices via contact, alignment, 2) input focuses on user actions such as trigger commands, entering data, or direct manipulation, 3) enrollment is based on one-time registration of devices with an identity, and 4) matching describes approaches where users compare output of the involved devices to confirm a connection.

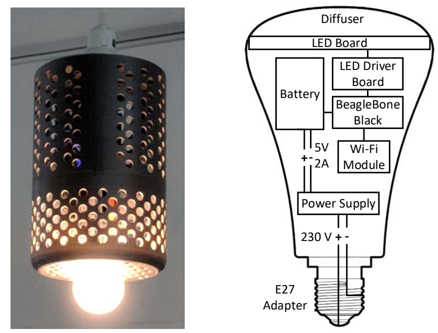
To infer close proximity of users, visible light positioning [6]–[8] is out of scope because we are not interested in the user's position to protect the user's privacy. Instead, we use context information such as ambient light patterns to recognize nearby devices due to the distance-restricted nature of light. In this context, to detect the proximity of devices, other approaches use radio signals [9], ambient audio [10], ambient noise and luminosity [11], accelerometer data caused by hand shaking [12], gait cycle detection of moving users [13], and magnetometer readings of very close devices [14]. The existing work aims to connect mainly two devices whereas DevLoc enables group associations and thereby we are able to flexibly control the granularity of device's proximity. Group association is not just an extension of pairwise association with more users [3]. Instead, many people expect that group association is a single-step procedure rather than multiple pairings. The user study of [3] reveals that close proximity is popular for groupwise associations, but also not rated highly for simplicity.

## III. DevLoc System for Device Grouping

We introduce the DevLoc framework for device grouping in Fig. 1(a) where radio-based communication like Wi-Fi covers larger areas and penetrates spatial barriers such as walls, doors.




(a) System overview of DevLoc combining Wi-Fi routers and light bulbs for device associations among users and IoT boards.




(b) DevLoc implements the proposed framework on our customized LocalVLC system [15] to enable visible light signaling for device grouping. We show the deployed 3D-printed light bulb together with the hardware platform.

Fig. 1: DevLoc for seamless device association

We enrich those with visible light signaling to control the spatial granularity of device's proximity based on the master-slave principle of the light bulbs covering larger rooms or across multiple rooms. The different colors at the light bulbs refer to different light patterns used for device associations. The dotted red circles highlight the association among different entities: a) device-to-device using solely the device's light signals for signal comparison or b) device-to-area using the device's light signal and an area's reference light signal for signal matching. DevLoc aims to associate user's mobile devices like tablets, smartphones, laptops, and static IoT boards for data sharing and aggregation. Our custom light bulb in Fig. 1(b) inspired by [16]–[18] is a central part of DevLoc which establishes a Wi-Fi link to the lighting configuration framework and emits light patterns at a high frequency, invisible for human eyes, to seamlessly group devices. This allows us to replace existing illumination infrastructure and we are able to limit the problem of light pollution, at which different visible lights are overlapping for illumination and communication. Our light bulbs to realize device grouping are an implementation choice and not fundamental to the system design. In the following, we describe the setup and working principle of DevLoc.

### A. Control the Spatial Granularity of Device Associations

The first task of the administrator is to specify the geographic structure of the device associations by selecting proximity areas. For example, such as in Fig. 1(a) via room numbers for area one and region names like corridor for area five. Initially each light bulb and Wi-Fi router registers itself at the lighting configuration framework running at the backend. Hence, DevLoc knows the light bulbs for all defined areas and randomly chooses for each region one of the light bulbs as master light bulb and the remaining light bulbs act as slaves. We randomly generate a light pattern adapting over time for each master light bulb and the slave light bulb(s) simply broadcast the same light pattern given from the master light bulb. On this basis, we can flexibly define the spatial granularity of device proximity based on the master-slave mechanism of our light bulbs by changing the groups of light bulbs covering different areas. For instance, we cover larger regions by using the same light pattern in different rooms which are semantically the same area such as area one in Fig. 1(a) covering two rooms compared to other areas limited by the room boundaries. The achievable spatial granularity of device associations is defined by the size of rooms and regions like corridors, and the number and distribution of light bulbs. For the most fine-granular proximity, each light bulb works on its own as master light bulb without any slave light bulbs. Our custom light bulb provides a communication range of up to 10 m. The master-slave mechanism of our light bulbs ensures a minimum of technical adaptions on existing illumination because only the master light bulbs need computing power to perform the device associations. The slave light bulbs require only a Wi-Fi connection to receive the commands from the master light bulb to broadcast a specific light pattern.

### B. Triggering Device Associations

We combine Wi-Fi routers and our custom light bulbs for triggering device associations. The second task of the administrator is to specify one Wi-Fi router for each master light bulb which continuously monitors the wireless connections of the Wi-Fi router. Due to the larger Wi-Fi coverage the binding is 1:m meaning one router is linked to multiple master light bulbs. If there are no device groups yet and in case of changes on the Wi-Fi connections, each linked master light bulb requests the continuously broadcasted light pattern received from the client(s) and initiates the device association to infer which devices are in the same light communication range instead of being only in the same Wi-Fi coverage. In case of a new Wi-Fi client, the master light bulb performs the signal comparison to infer the matching device group without affecting other devices. When a Wi-Fi device disappears at the router the master light bulb removes this single client from existing device groups. Besides that, the mobility of user's also affects the triggering of device associations. In case of static users it is sufficiently to observe the Wi-Fi connections for device grouping. However, for users moving between different areas but still connected to the same Wi-Fi router we need to manually start the device grouping for current device groups via a predefined period like every few seconds. We do not use signal strength changes of the user's Wi-Fi connection to update the device association because it is sensitive to reflections and shadowing due to moving and static objects, location and distance of users to the router, and layout and material of the building. The Wi-Fi signal strength can change unexpectedly and gives excessive false positives and false negatives causing frequent device association updates which decrease the user experience.

### C. Entities for Device Association: Device and Area

As illustrated in Fig. 1(a), the third and last task of the administrator is to define the mode of device associations for each master light bulb depending on the desired application using location-based services (LBS) or proximity-based services (PBS). Device-to-device association for PBS and device-to-area association for LBS. LBS are based upon the absolute position of a user to answer the question "where we are?". In contrast, PBS are based upon context information to find co-location with other points of interest to answer the question "who are we with?". The goal of LBS and PBS is to improve the users' daily lives by providing a personalized service to enable sharing of location information and location-aware information retrieval. We identify three main differences between device-to-device and device-to-area associations: 1) trigger point in time of the device grouping, 2) required number of clients for device association, and 3) signal comparison among different entities influencing the associated binding either device-to-device or device-to-area.

In case of device-to-device associations, triggering the device grouping requires at least two connected clients at the router which is linked to one or multiple master light bulbs. To match a Wi-Fi client to a device group, the master light bulb randomly chooses one client from each existing device group for signal comparison. The established binding between nearby clients lacks the information in which specific geographic region the clients are positioned. Hence, we can only realize PBS like data sharing among close users. LBS are not feasible such as indoor localization of a single or group of users.

On the other hand for device-to-area associations, the master light bulb(s) start the device grouping immediately after the client connected to the Wi-Fi router and compare the client's signal to the area's reference signal. We establish a direct binding between the device and area, and hence we know which device is in which region and at the same time which other devices are nearby. Via LBS including PBS we are able to offer more user services. In addition, there is no restriction in terms of the number of connected clients, e.g., to have at least two connected clients for device-to-device binding. Device-to-device grouping provides less location-specific information compared to the device-to-area grouping.

### D. Generation and Detection of Light Patterns

Our custom light bulb broadcasts random light patterns for device grouping. We independently generate a random series of light on and off periods and merge them afterwards to a light pattern. The duration of each light on and off period is in the range of [1, 5] ms. The minimum duration is defined via the fastest sampling rate achievable by the hardware of our light receiver, how fast the photodiode can be sampled. We determine the maximum duration of each light on and off period by avoiding unpleasant visual experience where light flickering effects are visible. The sender periodically broadcasts the light pattern for a limited time period. The length of the light pattern must be a multiple of two to be able to distinguish different light patterns, i.e., after each light on period appears a light off period. To improve the detection rate of light patterns at the light receiver, we check for each light on and off series if the time periods are sufficiently diverse that each duration is more than 10 % different from the other periods. Our light receiver samples the raw light signal via a photodiode and receives the voltage in mV where a higher voltage indicates a light on period and a lower voltage indicates a light off period.

**How to detect cycles in the light signal?** To find repeating patterns, we apply the cycle detection algorithm from [13] on our light pattern. The algorithm achieves a reliable signal segmentation based on normalization and supports signal similarity of arbitrary co-aligned sensor data. The algorithm's input is a vector of voltage amplitudes $z = (z_1, ..., z_n)$ and the output is a sequence of consecutive light signal cycles. We utilize auto-correlation and distance calculation to find repetitive signal parts. The auto-correlation is efficiently calculated via the Wiener-Khinchin theorem [19] with complexity $n \cdot log(n)$

$$\begin{aligned} F_R(f) &= FFT[z] \\ S(f) &= F_R(f) \cdot F_R^*(f) \qquad * \mathrel{\hat{=}} \text{conjugate} \\ R(\tau) &= IFFT[S(f)] \end{aligned}$$

where $z$ are the voltage amplitudes. The resulting auto-correlation $R(\tau)$ leads to $m$ non-ambiguous local maxima $\zeta = \arg\max(R(\tau)) = \{\zeta_1, ..., \zeta_i ..., \zeta_m\}$. We calculate the distances between all local maxima and a mean distance

$$\delta_{\text{mean}} = \left\lceil \frac{\sum_{i=1}^{m-1} \zeta_{i+1} - \zeta_i}{m-1} \right\rceil$$

where $\delta_{\text{mean}}$ can be used to select minima indices from $z$ which represent signal cycles. To be specific, each local maxima defines a start point and $\delta_{\text{mean}}$ a search range to find the local minima

$$\begin{aligned} \mu &= \{\mu_1, ...\mu_i, ..., \mu_{m-1}\} \\ \mu_i &= \arg\min(z_{\zeta_i}, z_{\zeta_i+1}, ..., z_{\zeta_i+\delta_{\text{mean}}}) \end{aligned}$$

Every $\mu_j$ represents the index of a minimum in $z$ limited to the range of $\delta_{\text{mean}}$. The indices in $\mu$ are used to split the voltage amplitude $z$ into light patterns

$$\begin{aligned} &Z = \{Z_1, ..., Z_i, ..., Z_q\} \\ &Z_i = (z_{\mu_{\frac{i}{2}}}, ..., z_{\mu_i}, ..., z_{\mu_{\frac{i+1}{2}}-1}) \text{ with } i = \{2, 4, ..., q\} \end{aligned}$$

This method works reliably only for simple light patterns with a maximum length of six on and off periods. Hence, we introduce our own method to detect light patterns taking into account the period of each light on and off phase. We summarize the light signal into a list of periods

$$\hat{z} = \{(s_1, d_1), ..., (s_n, d_n)\}$$

where $s_n \in \{0, 1\}$ describes whether the light is on or off and $d_i \in \mathbb{Z}$ specifies the duration of each period. We merge similar signal parts with a difference smaller than 10 % because the light sender introduced a 10 % dissimilarity among the light on and off periods to enhance the robustness of signal cycle detection. The remaining unique signal parts define the parts of the signal pattern together with the length. We overlay the light signal with a time window defined by the pattern length to identify the light pattern. We don't know the start position of the light pattern resulting in $m = {}^{\text{len}(\hat{z})}/_2 + 1$ possible candidates as light pattern.

### E. Implementation Details of DevLoc

We use the small, low-cost, single-board computer BeagleBone Black as system and development platform. We have implemented two Linux kernel modules to broadcast light patterns at the light bulb and receive light patterns at the user's mobile device. We use MQTT for the communication among light bulbs. Each master light bulb subscribes to the central backend to receive the configured light pattern which is further published to the slave light bulb(s). In addition, we take advantage of Python twisted, an event-driven network programming framework where we provide short callbacks to receive and send data between light bulbs and user clients.

## IV. Evaluation of Device Associations via DevLoc

We evaluate the physical channel of VLC with respect to a real-world deployment of DevLoc including the impact of ambient light and the field of view (FoV) at the receiver's photodiode. In addition, to analyze the performance of DevLoc in varying environments, we simulate two different environments with static and moving users to identify for each case the best working device grouping in terms of high detection accuracy and low runtime.

### A. Properties of VLC Physical Channel

Regarding a real-world deployment of DevLoc, we have evaluated in [15] the impact of ambient light at VLC by using two different LEDs as VLC transmitter, an omnidirectional LED with a weak light signal and a directional LED with a strong, beaming light signal. The results have shown that the directional LED is less influenced by the ambient light compared to the omnidirectional LED. Nevertheless, with a stronger ambient light similar to an active light source or direct sunlight the performance to detect signal patterns using the directional LED drops significantly. In contrast, the omnidirectional LED only works reliably at a low ambient light intensity. In addition, we measure the FoV at the photodiode of the receiver. The omnidirectional LED obtains a range of 165°–50° and the directional LED achieves a FoV of 175°–5°. As future work, we plan to enhance the robustness of

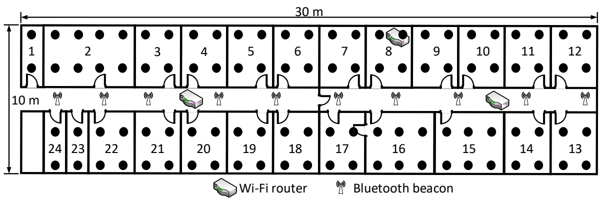



Fig. 2: We use our university lab as simulation environment for device associations which consists of 24 rooms with different sizes. We take real traces of the Wi-Fi and Bluetooth environment at different positions over the university lab for comparison with device localization.

DevLoc by analyzing the effect of overlapping light patterns from different light bulbs whether we are able to separate and identify the different signals. We will adopt the algorithm in [20] which uses orthogonal codes to detect and isolate adjacent light sources, e.g., light markers for object identification.

### B. Simulation Settings for Device Association

We evaluate DevLoc via a dedicated simulator running two different simulations with static and dynamic users. To support this, we have previously identified the best working parameters for device grouping which are summarized and italicized in Table I. We perform a trace-driven simulation with persisted environment data from our university lab as shown in Fig. 2. Thereby, we compute statistical features (min, max, median, var, std, mean, sum, length) and time-series tailored features via tsfresh [21] for light patterns. To enable the simulations, each grouping client uses three different real traces: Wi-Fi and Bluetooth scans, and random light patterns with varying length. We achieve a realistic simulation by imitating the network latency between the grouping server and the clients by a random waiting time before each client sends the requested environment data to the grouping server. Thereby, we choose a random start within the sensing range for light patterns, Wi-Fi and Bluetooth scans, and we randomly select a sampling period within the identified best working sampling ranges.

### C. Static Device-to-Device Simulation of Device Grouping

**Simulation settings** In the static simulation no user is moving and each user remains in the same room. The grouping server waits until all devices are connected and starts the device grouping. Table I shows the parameters for the static simulation. We perform the device grouping using random light patterns with a varying length $\in \{2, 4, 6, 8, 10\}$ and for at least two users up to ten users.

**Simulation results** By using 10-fold cross validation, Table II presents the best working device grouping technique (highlighted in bold) in terms of a fast reasoning and a reasonable total result, meaning the average over accuracy, precision, recall, and F1-score. The ML-based device grouping performs similar or slightly better than the signal similarity metrics like Spearman and Pearson. In contrast, the device localization using Bluetooth and Wi-Fi features works far

TABLE I: Settings from parameter estimation (italic) and simulation parameters (bold) for device grouping

| Simulation | Parameters | |
|---|---|---|
| Static & Dynamic | **Similarity metrics** | Pearson, Spearman |
| | *Similarity threshold* | 0.7 |
| | *Similarity equalize method* | DTW |
| | **Localization classifiers** | Content-based filtering, random forest, SVM |
| | *Sampling period localization* | 5 s |
| | **Similarity classifiers** | Random forest, extra trees, gradient boosting |
| | *Sampling period to train similarity classifiers* | 50 ms |
| Static | **Light patterns** | [2, 4, 6, 8, 10] |
| | **Users** | [2, 3, 4, 5, 6, 7, 8, 9, 10] |
| Dynamic | **Grouping frequency** | [10, 20, 30] s |
| | **Users** | [3, 5, 10] |
| | **Rooms** | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] |

worse. Regarding the runtime of our device grouping, the median time to receive data for device grouping is about 1.41 s which stands for 83.93 % of total time in comparison to the device grouping with 0.27 s and 16.07 % of the total time.

For a thorough evaluation we further analyze the performance using different number of grouping users. We reach the highest total result of 0.97 with six grouping users because with less users the grouping signals lack significant patterns and with more users the noise over the grouping signals increases leading to a higher error rate for device grouping. In the following, we show the number of grouping users with descending total result in brackets, meaning the average over accuracy, precision, recall, and F1-score: 6 (0.97), 4 (0.94), 3 (0.93), 8 (0.93), 5 (0.92), 9 (0.92), 10 (0.92), 7 (0.92), 2 (0.87). Moreover, we evaluate the performance of light patterns with different lengths for device grouping. The light pattern with four random on and off periods works best compared to a decrease of 9 % using the worst light pattern with ten random periods. In detail, we present the average performance of light patterns with different lengths sorted by descending total result in brackets: 4 (0.97), 2 (0.95), 6 (0.91), 8 (0.9), 10 (0.88).

### D. Dynamic Device-to-Area Simulation of Device Grouping

**Simulation settings** In comparison to the static simulation, the users are moving between different rooms in the dynamic simulation. Fig. 2 shows our simulation environment for device associations where the rooms are positioned in a rectangular grid with an intra room distance of 2 m and inter room distance of 3 m and we calculate the distances among all room combinations. For each user we calculate a random path between the rooms using the duration of one simulation iteration of 20 min and distribute the time as duration of stay over the rooms using a multinomial distribution. As a result, the user's random path is a list of tuples with duration of stay

TABLE II: Best working classifiers and features for device grouping with static and moving users

| Simulation | Grouping technique | Feature type | Result | Runtime | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|---|
| Static | Gradient boosting | Selected tsfresh | .96 | 4.02 s (8) | 1 | .94 | .94 | .94 |
| | Gradient boosting | Selected statistical | .83 | 0.75 s (4) | .89 | .81 | .81 | .8 |
| | **Gradient boosting** | **Statistical** | **.81** | **0.45 s (2)** | **1** | **.75** | **.75** | **.75** |
| | Spearman | Light pattern | .81 | 0.49 s (3) | 1 | .75 | .75 | .75 |
| | Spearman | Duration of light pattern | .81 | 0.42 s (1) | 1 | .75 | .75 | .75 |
| | Pearson | Light signal | .64 | 2.06 s (5) | .25 | .77 | .79 | .76 |
| | Random forest | Bluetooth | .43 | 2.64 s (7) | .34 | .48 | .52 | .38 |
| | SVM | Wi-Fi | .31 | 2.61 s (6) | .32 | .2 | .44 | .26 |
| Dynamic | Pearson | Duration of light pattern | .95 | 0.46 s (3) | 1 | .93 | .93 | .93 |
| | Pearson | Light pattern | .95 | 0.47 s (4) | 1 | .93 | .93 | .93 |
| | **Pearson** | **Light signal** | **.95** | **0.28 s (2)** | **.95** | **.95** | **.95** | **.95** |
| | Gradient boosting | Selected tsfresh | .93 | 1.58 s (8) | .9 | 1 | .9 | .93 |
| | Gradient boosting | Selected statistical | .93 | 0.53 s (5) | .9 | 1 | .9 | .93 |
| | Content-based filtering | Bluetooth | .84 | 0.59 s (6) | .95 | .81 | .81 | .81 |
| | Content-based filtering | Wi-Fi | .84 | 0.61 s (7) | 1 | .78 | .78 | .78 |
| | Extra trees | Statistical | .83 | 0.26 s (1) | .75 | 1 | .75 | .83 |

for each room where the user stays and moves to the next room if the duration of stay expired. For example, user A has the random path: [(120, 1), (300, 3), ...] which specifies that the start position is in room 1 and after 120 s the user moves to room 3 and stays there for 5 min, and so forth. Thereby, we randomly create user groups for each room, i.e., at which simulation time how many users are in the same room. During the simulation each user chooses a random movement speed in the range of 1.25 to 1.53 m/s (4.5–5.5 km/h) [22] for each movement between rooms. If the users are in motion they are in the corridor and not associated with any room. For device grouping, each room is associated with unique location-dependent environment data including Wi-Fi and Bluetooth scans, and light patterns and acts independently of other rooms. Table I shows the parameters for dynamic device-to-area simulation covering grouping frequency, number of users, and number of rooms.

**Simulation results** Via 10-fold cross validation, Table II shows the best working device grouping (highlighted in bold) with respect to a fast runtime and a reasonable total result, meaning the average across accuracy, precision, recall, and F1-score. In contrast to the static simulation, the device grouping based on similarity metrics works slightly better compared to ML-based device grouping. The device localization using Wi-Fi and Bluetooth features achieves a similar result. With respect to the runtime of our device grouping, the median time to receive data for device grouping is about 0.43 s (71.67 % of the total time) in comparison to the device grouping with 0.17 s (28.33 % of the total time). Moreover, we analyze the performance of device grouping with a varying number of rooms, sorted after decreasing total result in brackets: 1 (0.99), 2 (0.96), 3 (0.92), 5 (0.9), 6 (0.89), 4 (0.87), 8 (0.84), 7 (0.82), 9 (0.79), 10 (0.76). The device grouping works best with less rooms because the more rooms the higher the chance that the user lacks the up-to-date light pattern of the designated room due to movement or decoding issues. Besides that, the frequency of device grouping with 20 s works best, the accuracy of device grouping decreases by 16 % with 30 s and with a 10 s frequency the accuracy decreases another 8 %.

To sum up, Table II shows two different best working classifiers and features for device grouping depending on the use case either for static or moving users. The scenario with several rooms and moving users is more realistic in practice and we favor this approach for device association.

## V. Conclusion

DevLoc is a ready-to-deploy system solution to enable seamless device grouping based on visible light signaling for data sharing and aggregation. Our custom light bulb broadcasts light patterns so that clients detect cycles in the light patterns for device grouping. Our evaluation of DevLoc via two simulations with a single room and static users and multiple rooms with moving users reveals that in general the machine learning based signal similarity performs best compared to distance and correlation metrics.

To ensure that DevLoc protects the user's privacy during device grouping at the light bulb, we will analyze how to apply fully homomorphic encryption for time-series data where multiple parties compute whether they are nearby without learning each other's inputs. Moreover, by using our Morse-code inspired modulation scheme from LocalVLC [15], we plan to encode a location identifier emitted by our custom light bulb(s) for device grouping and compare the results with the device grouping via light signal patterns in terms of robust device grouping in presence of light interference.

## REFERENCES


[1] A. Narayanan, N. Thiagarajan, and M. Lakhani, "Location Privacy via Private Proximity Testing," in *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS)*, 2011.

[2] S. Mare, L. Girvin, F. Roesner, and T. Kohno, "Consumer Smart Homes: Where We Are and Where We Need to Go," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2019, pp. 117–122.

[3] M. K. Chong and H. W. Gellersen, "How Groups of Users Associate Wireless Devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2013, pp. 1559–1568.

[4] A. A. de Freitas and A. K. Dey, "Using Multiple Contexts to Detect and Form Opportunistic Groups," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 2015, pp. 1612–1621.

[5] M. K. Chong, R. Mayrhofer, and H. Gellersen, "A Survey of User Interaction for Spontaneous Device Association," *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–40, 2014.

[6] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light," in *Proceedings of the 20th International Conference on Mobile Computing and Networking (MobiCom)*, 2014, pp. 447–458.

[7] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang, "Lightweight Visible Light Positioning for Wearables," *GetMobile: Mobile Computing and Communications*, vol. 19, no. 3, pp. 18–21, 2015.

[8] ——, "Wearables Can Afford: Light-Weight Indoor Positioning with Visible Light," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2015, pp. 317–330.

[9] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, "Amigo: Proximity-Based Authentication of Mobile Devices," in *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp)*, 2007, pp. 253–270.

[10] D. Schürmann and S. Sigg, "Secure Communication Based on Ambient Audio," *Secure Communication Based on Ambient Audio*, vol. 12, no. 2, pp. 358–370, 2013.

[11] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani, "Context-Based Zero-Interaction Pairing and Key Evolution for Advanced Personal Devices," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2014, pp. 880–891.

[12] R. Mayrhofer and H. Gellersen, "Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 792–806, 2009.

[13] D. Schürmann, A. Brüsch, N. Nguyen, S. Sigg, and L. Wolf, "Moves like Jagger: Exploiting variations in instantaneous gait for spontaneous device pairing," *Pervasive and Mobile Computing*, vol. 47, pp. 1–12, 2018.

[14] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "MagPairing: Pairing Smartphones in Close Proximity Using Magnetometers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2016.

[15] M. Haus, A. Y. Ding, and J. Ott, "LocalVLC: Augmenting Smart IoT Services with Practical Visible Light Communication," in *Proceedings of the 20th IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–9.

[16] S. Schmid, J. Ziegler, G. Corbellini, T. R. Gross, and S. Mangold, "Using Consumer LED Light Bulbs for Low-Cost Visible Light Communication Systems," in *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems (VLCS)*, 2014, pp. 9–14.

[17] Gummeson, Jeremy, J. McCann, C. Yang, D. Ranasinghe, S. Hudson, and A. Sample, "RFID Light Bulb: Enabling Ubiquitous Deployment of Interactive RFID Systems," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 1, no. 2, 2017.

[18] S. Schmid, T. Bourchas, S. Mangold, and T. R. Gross, "Linux Light Bulbs: Enabling Internet Protocol Connectivity for Light Bulb Networks," in *Proceedings of the 2nd International Workshop on Visible Light Communications Systems (VLCS)*, 2015, pp. 3–8.

[19] L. Cohen, "Generalization of the Wiener-Khinchin Theorem," *IEEE Signal Processing Letters*, vol. 5, no. 11, pp. 292–294, 1998.

[20] A. U. Guler, T. Braud, and P. Hui, "Spatial Interference Detection for Mobile Visible Light Communication," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018, pp. 1–10.

[21] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.

[22] N. Carey, "Establishing Pedestrian Walking Speeds," *Portland State University*, 2005.