# Localization in Ultra Narrow Band IoT Networks: Design Guidelines and Trade-Offs

Hazem Sallouha, *Graduate Student Member, IEEE,* Alessandro Chiumento, *Member, IEEE,*
Sreeraj Rajendran, *Graduate Student Member, IEEE,* and Sofie Pollin, *Senior Member, IEEE*

*Abstract*—Localization in long-range Internet of Things networks is a challenging task, mainly due to the long distances and low bandwidth used. Moreover, the cost, power, and size limitations restrict the integration of a GPS receiver in each device. In this work, we introduce a novel received signal strength indicator (RSSI) based localization solution for ultra narrow band (UNB) long-range IoT networks such as Sigfox. The essence of our approach is to leverage the existence of a few GPS-enabled sensors ($GSN$s) in the network to split the wide coverage into classes, enabling RSSI based fingerprinting of other sensors ($SN$s). By using machine learning algorithms at the network backed-end, the proposed approach does not impose extra power, payload, or hardware requirements. To comprehensively validate the performance of the proposed method, a measurement-based dataset that has been collected in the city of Antwerp is used. We show that a location classification accuracy of 80% is achieved by virtually splitting a city with a radius of 2.5 km into seven classes. Moreover, separating classes, by increasing the spacing between them, brings the classification accuracy up-to 92% based on our measurements. Furthermore, when the density of $GSN$ nodes is high enough to enable device-to-device communication, using multilateration, we improve the probability of localizing $SN$s with an error lower than 20 m by 40% in our measurement scenario.

*Index Terms*—Internet of Things, ultra narrow band, localization, RSSI, fingerprinting, machine learning.

## I. INTRODUCTION

The Internet of Things (IoT) is becoming ubiquitous, enabling everyday objects to be equipped with computation and communication capabilities [1]–[5]. To realize IoT, currently deployed networks are using long-range, low power, and low throughput communications such as Sigfox [6], Weightless-P [7], LoRa [8], and narrowband (NB)-IoT [9]. Among these technologies, ultra narrow band (UNB) based solutions, such as Sigfox, offer a compelling mix of simplicity and wide coverage to provide connectivity to millions of devices.

The demand for IoT is set to rise further with the advent of location-based services. In fact, many IoT applications fundamentally depend on the location information to meaningfully interpret any physical measurements collected [3,4]. Unfortunately, the long-range and narrow bandwidth, associated with UNB IoT networks, make localization problematic. On one hand, GPS receivers are power hungry and rather expensive to integrate with each IoT node. On the other hand, ranging-based methods lack accuracy because of the long distances and the ultra narrow bandwidth [10]. A promising alternative method to tackle these challenges is fingerprinting-based localization. Particularly, by leveraging the advances in machine learning (ML) algorithms, fingerprinting methods have expanded to include outdoor scenarios [1,2].

Conventional fingerprinting localization techniques consist of two phases: a training phase, namely an offline phase, and a location estimation phase, namely an online phase. In the training phase, received signal strength indicator (RSSI) measurements, a.k.a. fingerprints, are collected at known positions and concurrently stored in a dataset. Subsequently, in the online phase, nodes locations are estimated by comparing the real-time RSSI measurements with the entries from the dataset [11]. This comparison process is performed using ML algorithms. In general, the main disadvantage of fingerprinting localization is the necessity to keep updating the training datasets. Such a process is both time-consuming and labor-intensive. In this paper, we tackle this drawback by relying on a few GPS-enabled sensor ($GSN$) nodes to collect the training data making it, constantly, up to date. This promising potential of fingerprinting methods alongside the insisting demands for an accurate localization technique for UNB IoT networks inspired our work in this paper.

### A. Related Works

The localization problem has been extensively explored in the literature for various kind of networks in different scenarios [12]–[16]. However, the new characteristics of IoT networks, such as the long-range, limited number of messages, and the inexpensive IoT nodes, bring extra localization challenges. Therefore, the localization problem in IoT networks has recently attracted considerable research focus [4,17]. Among all localization techniques, RSSI-based methods have been proven as one of the effective solutions [18]–[24]. In particular, RSSI fingerprinting localization has been widely used in various works [24]–[28], in which both the offline and online phase challenges are addressed.

The main challenge in the offline phase is to minimize the time and effort required to construct a radio map for the area of interest [11]. To address this issue, the authors in [27] and [29] adopted probabilistic fingerprinting methods based on RSSI distribution information. Experimental results showed that probabilistic approaches provide good performance. However, the validation of these approaches is

Authors are with the Department of Electrical Engineering, KU Leuven, Belgium. e-mail: {hazem.sallouha@esat.kuleuven.be}. Alessandro Chiumento is also with CONNECT Centre, Dublin, Ireland. This work was supported in part by the FWO SBO Project SAMURAI.

limited to indoor environments [28]. For the online phase, the main challenge is mapping the new RSSI measurements to their corresponding locations using the radio map. Here, ML algorithms played a significant role in achieving better localization accuracy. A comparison between relevant ML algorithms for fingerprinting-based localization is presented in [25]. Among the presented algorithms, support vector machines (SVM) and tree-based (e.g., random forest) have shown satisfying accuracy for this range of problems [25,26]. In [30], the authors proposed SVM based device-free localization using channel state information fingerprinting. Also, in cases where the training dataset and the number of features are relatively small, SVM has shown sufficient classification accuracy [26,31]. Such characteristics are encountered in long-range networks considered in this work. However, the focus of these aforementioned works is mainly indoor environments. Hence, the proposed methods are incompetent for outdoor long-range IoT networks.

The outdoor localization accuracy of a LoRa deployment using RSSI measurements is investigated in [32] where a localization error around 30 m is achieved. However, the authors considered an optimistic scenario with a very dense gateway deployment. In [33] and [34], the localization of IoT nodes using mobile anchors is proposed. Although mobile anchors introduce swift and flexible localization solutions, they typically serve limited areas [34]. A method that uses information about nearby Wi-Fi access points to localize Sigfox nodes is proposed in [35]. This approach achieved location estimation with a median error of 39 m. However, it requires dedicated Sigfox messages, leading to inefficient use of the limited Sigfox transmissions.

The localization problem in long-range UNB IoT networks using RSSI fingerprinting has been addressed in [1,2,36]. In [2], the authors introduced datasets collected in large-scale outdoor environments. Moreover, a primary fingerprinting localization using $k$-nearest neighbors ($k$-NN) algorithm on the dataset is examined. A comparison between different RSSI based techniques for long-range IoT networks is presented in [36]. The comparison results concluded that the fingerprinting method outperforms other RSSI localization methods. However, in both [2,36], the training data has been randomly selected from the whole dataset, which implies having knowledge of RSSI fingerprints all over the coverage area. In fact, this requires either a massive number of GPS-enabled nodes to collect tanning data, which is expensive, or alternatively, spend time and effort in building a radio map that is highly vulnerable to become outdated. In [1] a novel measurement based localization approach that leverages the existence of a few GPS nodes (e.g., 1% of the nodes) is presented. Furthermore, enhancement of the localization accuracy using device-to-device short-range communication is introduced. However, the measurements were only limited to a university campus, and hence, the performance of the proposed approach in large-scale cities is an open problem.

### B. Contribution and Paper Structure

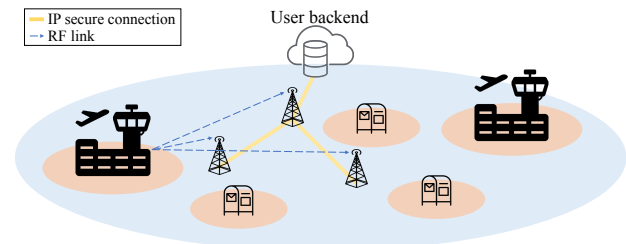In this work, we introduce a novel measurement-based localization approach that leverages the existence of a few



Fig. 1: Illustration of a long-range IoT network in which knowing at which post office your package is, provides sufficient location information.

GSN by using their fingerprints to localize other sensors. In particular, we extend the work presented in [1] by applying the localization approach on a larger scale to include an entire city. This allows us to introduce new design parameters and subsequently investigate the trade-offs between them. Our approach starts by classifying the non-GPS sensor (SN) nodes into classes and then followed by a multilateration process within classes for localization with higher precision.

*1) Classification based localization:* In the first step, we classify SN nodes into classes anchored by GSN nodes. For instance, in many applications, such as package localization, knowing in which airport your suitcase is, gives sufficient information. In fact, based on the application requirements, it can be assumed that classes are connected or separated, as shown in Fig. 1. To investigate the feasibility of the proposed approach, we introduce various design parameters such as the number of classes, the spacing between adjacent classes, the size of training data, and the number of features used. Each one of these parameters affects the classification accuracy and hence should be chosen carefully based on the application. For instance, the spacing between adjacent classes can be tuned to have localization service all over the coverage area (i.e., connected classes) or limited to certain locations (i.e., separated classes) based on the application requirements.

Moreover, we address the size of the training data, including the number of training examples and the number of features when using SVM and random forest (RndF) classifiers. This is particularly important given the limited number of messages in long-range IoT networks. Another essential parameter is the number of GSNs, represented by the number of classes, assuming at least one GSN per class. In fact, increasing the density of GSNs will result in smaller classes radii, lower localization errors, but more similarity in the fingerprints of nearby GSNs and hence higher classification error. Therefore, in the case of high GSNs density, we can rely on device-to-device (D2D) communication to perform multilateration.

*2) Regression-based distance estimation:* This step improves the localization accuracy by relying on device-to-device (D2D) communication between the GSN anchors and SNs, as IoT nodes, typically, by default, have modems that support short-range D2D communication [37]. Instead of decreasing the localization errors by increasing the number of classes, i.e., GSNs, we fine-tune the location within the class by performing multilateration using three GSNs. We can then localize nodes in a two-step method: first, fingerprinting classification using GSNs and then multilateration relying on D2D communica-

tion. In fact, limiting the use of RSSI distance estimation to short distances provides satisfactory performance.

In contrast to the existing literature on fingerprinting localization, our proposed two-step method saves time and effort required to build a full radio map. Such a radio map is highly vulnerable to become outdated, particularly with the long-range and outdoor deployment considered. Instead, we first rely on *GSN* anchors, with their constantly updated RSSI measurements, to classify *SN*s. However, since the locations of *GSN*s are sparse, the estimated location of a given *SN* node is represented by a zone, namely, a class. Such an estimated zone is, in fact, sufficient for many applications, such as package localization. Nevertheless, to further refine this estimation, we introduce the second step for those applications that require more localization precision. This step improves the precision, conditioned on the availability of D2D communication. In particular, the second step relies on regression-based distance estimations, delivering the exact position of the *SN* node.

The rest of the paper is organized as follows. Section II details the communication technologies used and clarifies the localization problem in long-range UNB IoT networks. In Section III, we introduce our fingerprinting base approach and detail the ML algorithms considered in this work. Localization based on distance estimation is presented in Section IV. Our experimental results using a real Sigfox deployment are presented in Section V. Finally, we conclude our work in Section VI.

## II. BACKGROUND AND PROBLEM STATEMENT

In this section we introduce the structure of UNB IoT network. Subsequently, the localization problem in such networks is thoroughly detailed.

### A. Background

Sigfox [6], is an IoT provider which uses UNB channels. The physical layer of Sigfox uses BPSK modulation technique providing a fixed bit rate of 100 bps with a payload up to 12 bytes. Moreover, the ISM 868 MHz band is used from which a bandwidth of 40kHz is split, providing orthogonal channels with a bandwidth of 100Hz each. Consequently, IoT sensor nodes randomly pick a channel for each transmission [38]. Following the regulations of the ISM band, Sigfox MAC layer is designed to limit the number of transmissions to 140 messages a day.

*1) Sigfox - Cellular IoT Network:* Sigfox is a star cellular network in which the uplink data flow from nodes to base stations (BSs) are assumed to be 97% of the overall traffic. An illustration of a Sigfox network is shown in Fig. 1. As one can see in the figure, the network consists of three parts: 1) the things, which could be any sensor or package equipped with an RF transmitter, 2) the base stations, which collect the data transmitted by the sensors and 3) the network's cloud to which all base stations are connected. Moreover, the user back-end could be added optionally, in different forms, to allow the users to access their data either via email, API, or store it to their database. In the scope of this paper, we highlight the three main elements used in our experiments.

- *Base stations (BS)s*: A single Sigfox base station can provide a coverage range up to 40 km. Hence few BSs are able to cover an entire city, both indoor and outdoor [6]. Sigfox BSs are connected to a centralized user back-end, enabling connectivity for millions of nodes.
- *Sensor Nodes (SN)*: The sensor nodes used are equipped with Telecom Design (TD) modems (TD1208). These modems support communication over Sigfox and use a transmit power up to 25 mW.
- *GPS-enabled nodes (GSN)*: They are nodes with the same functionality as *SN*. However, their modems are equipped with a GPS-receiver (TD1204) and, therefore, their coordinates are included in the messages they send.

Another possible implementation is to use hybrid networks in which *SN* nodes forward data to gateway nodes which, in turn, send it to the BSs [37]. Such a short relay connection can be made using TD-LAN. In our case, we assume that *GSN*s receive messages from *SN*s. This LAN network, with its short range communication, can be used to increase the localization accuracy of *SN* nodes within classes.

*2) Device-to-Device LAN enabled:* Nodes used in our experiments have TD Sigfox modems. These modems support communication over both Sigfox and TD LAN. Therefore, device-to-device communication can be enabled to build a TD-LAN network [37]. The TD-LAN is an energy-efficient network uses time division duplex (TDD) with narrow bandwidth channels of 25 kHz. Gaussian minimum shift keying (GMSK) is used as a modulation technique, and the frequency band is similar to Sigfox, i.e., ISM 869 MHz. In TD-LAN, the payload can go up to 17 bytes with a transmission rate of 9600 bps. A star topology can be formed by fixing a particular node in receive mode while its neighbors transmit upon request.

### B. Problem Statement

Each Sigfox message is received by all reachable BSs. However, they are typically far, in a region where the RSSI-distance curve resolution is not sufficient. Since long-range IoT networks are cellular based, one might intuitively suggest using the multiple BSs to perform multilateration ranging-based techniques for localization. In fact, unlike the typical mobile cellular networks, in long-range IoT networks, each base station covers a relatively large distance, i.e., up to 40 km [6,38], hence leading to a rather large uncertainty zone. Particularly, in multilateration process, the accuracy depends on the distance from nodes to BSs. This distance is estimated using time of arrival (ToA) or RSSI. Firstly, if ToA is used, the Cramér-Rao lower bound (CRLB) of the estimated distance $\hat{d}$ is given by [10]

$$\sqrt{\text{Var}(\hat{d})} \geqslant \frac{c}{2\sqrt{2}\pi\sqrt{SNR}\beta},\tag{1}$$

where $Var\{\cdot\}$ is the variance, $c$ is the speed of light, $SNR$ is the signal-to-noise ratio, and $\beta$ is the effective signal bandwidth. Based on (1), the accuracy of ToA localization techniques is directly proportional to the signal bandwidth. Long-range communication mainly works with UNB transmission [6]. Therefore, the UNB channels of 100Hz lead to
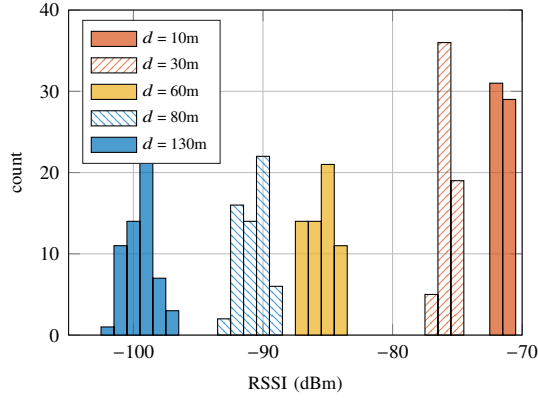
Fig. 2: A Histogram showing the variance of RSSI measurements at various distances.
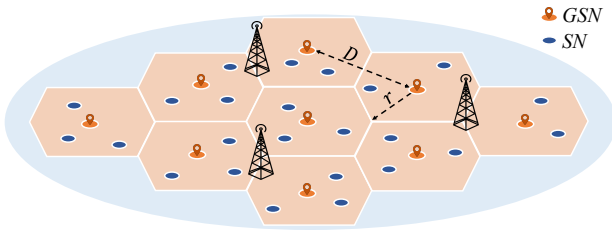


Fig. 3: Classifying $SN$s to one of the virtual $GSN$'s classes.

a significantly low time resolution in which the transmission time goes up to 1-2 seconds per message. Such a low time resolution severely degrades the estimation accuracy.

For the case of RSSI, the CRLB of an estimated distance $\hat{d}$ is given by the following inequality [10]

$$\sqrt{\text{Var}(\hat{d})} \geqslant \frac{\ln 10}{10} \frac{\sigma_{sh}}{n_p} d, \qquad (2)$$

where $d$ is the distance between the node and the base station, $n_p$ is the path loss exponent, and $\sigma_{sh}$ represents the standard deviation of the log-normal shadowing effect. By detailing the variables in (2), we first have $\sigma_{sh}$ and $n_p$ that are channel parameters and hence, are out of control for a given environment. In (2), moreover, one observes that the estimator accuracy is inversely proportional to the distance $d$, i.e., the larger the distance, the lower the accuracy. This means that poor accuracy is achieved with long-range communication.

Interestingly, RSSI measurements present two distinctly different behaviors when the receiver is close to the transmitter, rather than further away [1]. For short distances, the RSSI-distance curve decreases rapidly with the distance. However, as the distance increases (>100 m), the RSSI-distance slope decreases asymptotically. Concurrently, the variations around the mean increases, as shown in Fig. 2. Therefore, RSSI measurements can be used for distance estimation and fingerprinting, depending on the distance. At short distances, ranging using a regression process gives an accurate estimation of the distance between the node and the base station. On the other hand, as the distance increases and hence, the variance of the RSSI measurements, the distance estimation error rises dramatically. Therefore, it is preferable to leverage such variations to distinguish between different locations and

TABLE I: Representation of RSSIs collected from one $GSN$ node

| Time index | RSSI (dBm) | | | |
|---|---|---|---|---|
| | $BS_1$ | $BS_2$ | ... | $BS_M$ |
| $t = 0$ | $RSSI_1^{(0)}$ | $RSSI_2^{(0)}$ | ... | $RSSI_M^{(0)}$ |
| $t = 1$ | $RSSI_1^{(1)}$ | $RSSI_2^{(1)}$ | ... | $RSSI_M^{(1)}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $t = T - 1$ | $RSSI_1^{(T-1)}$ | $RSSI_2^{(T-1)}$ | ... | $RSSI_M^{(T-1)}$ |

subsequently, classify nodes into classes using their RSSI measurements.

## III. Fingerprinting Based Classification

In this section, we present our novel long-range localization approach based on region partitioning using $GSN$s. In particular, $GSN$'s RSSI fingerprints are used as landmarks (i.e., training data) for our classification framework, allowing us to divide the $SN$s into classes. In the following, we introduce our system design. Subsequently, we highlight two algorithms which are widely used for such classification problems: RndF and SVM classification algorithms. Finally, we present our algorithm summary and detail the error metric of our classification problem.

### A. System Design

Consider an outdoor UNB IoT network in which sensor nodes are randomly distributed in a wide region. Assume two kinds of sensor nodes: $GSN$ and $SN$. We aim at using the nodes $GSN_1$, $GSN_2$, ..., $GSN_L$ with their know locations $c_1$, $c_2$, ..., $c_L$, respectively, to split the wide region into $L$ classes, as shown in Fig. 3. The classes are assumed to have equal radii denoted as $r$ with $GSN$s located at the centers. Moreover, we assume equal spacing $D$ between the centers of adjacent classes, i.e., adjacent $GSN$s. In order to split the region into classes based on $GSN$s, we have $D \geq \sqrt{3}r$. Accordingly, the spacing between adjacent classes, $x$, is defined as:

$$x = D - \sqrt{3}r, \qquad x \in [0, D). \qquad (3)$$

If $SN$s are uniformly distributed all over the region then, we choose classes with $D = \sqrt{3}r$, i.e., $x = 0$. However, if $SN$s are located in given parts of the region, then we choose *separated classes* with spacing $x > 0$, i.e., $D > \sqrt{3}r$.

In long-range UNB IoT networks, each message sent by $GSN$ or $SN$ is received by $M$ base stations at which RSSI is measured. This gives us a vector $\mathbf{R}^{(t)} = [RSSI_1^{(t)}, RSSI_2^{(t)}, ..., RSSI_M^{(t)}] \in \mathbb{R}^M$, with the element $RSSI_m^{(t)}$ representing the RSSI sample recorded at time index $t$ at base station $m$, where $t = 0, 1, ..., T-1$ and $m = 1, 2, ..., M$. Having $T$ messages received, it is then possible to construct a matrix $\mathbf{C} = [\mathbf{R}^{(0)}, \mathbf{R}^{(1)}, ..., \mathbf{R}^{(T-1)}]^T \in \mathbb{R}^{T \times M}$, with $[.]^T$ being the transpose operator.

Now, using $L$ $GSN$s, we assume that the region is divided into $L$ classes based on $GSN_1$, $GSN_2$, ..., $GSN_L$ with their matrices $\mathbf{C}_1$, $\mathbf{C}_2$, ..., $\mathbf{C}_L$, respectively. Table I presents the RSSI values of matrix $\mathbf{C}_l$ for a given $GSN$, namely $GSN_l$. Considering all $L$ classes, the measurements can be stored

in a three-dimensional matrix with a size of $T \times M \times L$. In classification problems, this represents a $T \times L$ set of training examples with $M$ different features. Now, given the locations and the RSSI fingerprints collected from $GSN$s, we proceed to classify all $SN$s to one of these classes using ML algorithms. In the next section, we present the relevant algorithms considered in this work.

### B. Supported Vector Machines

SVM is a supervised ML algorithm commonly used in classification problems. It is distinguished by its support vectors and kernel function [31]. The support vectors are chosen in the training phase and are defined as the critical elements of the dataset. This set of critical elements determines a *separating hyperplane* with the objective of maximizing decision margins. The learning of the hyperplane is done by mapping the dataset into a higher dimension. This is where the kernel plays a role. In this paper, we consider the RBF kernel function, a.k.a. Gaussian kernel. RBF kernel is known for its empirical effectiveness and is easy to calibrate [26,27]. Now, let vector $\mathbf{R}$ denotes a training example from a given class, i.e., $GSN$ node, and vector $\mathbf{R}'$ indicates a test example from a given $SN$ node, the corresponding RBF kernel is defined as

$$\mathcal{K}(\mathbf{R}, \mathbf{R}') \; = \; \exp\left( - \frac{\|\mathbf{R} - \mathbf{R}'\|^2}{2\sigma^2} \right), \tag{4}$$

where $\|.\|$ is the euclidean distance. $\sigma^2$ is known as the smoothness parameter of the kernel (i.e., it controls the influence of individual training examples). In other words, low $\sigma^2$ means that only *border examples* are used in determining the separating hyperplane. Whereas with high values of $\sigma^2$ *far from the border examples* are also considered in defining the separating hyperplane. It is worth noting that, in our implementation, $\mathbf{R}$ and $\mathbf{R}'$ represent vectors of RSSI measurements collected from $GSN$ and $SN$, respectively.

To gain insight into how to select the value of $\sigma^2$, consider the RSSI histogram shown in Fig. 2. For short distances (e.g., $10\,\mathrm{m}$ and $30\,\mathrm{m}$), where RSSI varies sharply, a kernel function with low $\sigma^2$ ($\approx 0.01$) is needed. On the contrary, at long distances where RSSI varies smoothly with the distance, a kernel function with high $\sigma^2$ ($\approx 2$) is preferable. In fact, high and low values of $\sigma^2$ lead to SVM with simple and sophisticated separating hyperplane, respectively. Another important parameter that should be considered when choosing the value of $\sigma^2$, is the number of features, $M$. In fact, a favorable initial value of $\sigma^2$ is $M/2$ [39]. Subsequently, we tune it, if needed, based on the way the training dataset varies.

### C. Random Forest

RndF is ensemble classifier based on creating various tree predictors in the training phase and then output the class labels which have the majority vote among the trees. Each tree is grown by randomly drawing samples, with replacement, from the training set. In our localization approach messages collected from $GSN$ are used to construct trees. These trees are then combined by averaging their probabilistic prediction. RndF attains high classification accuracy and can handle

---

**Algorithm 1** Model training and classification of $SN$s

1: **Input:** $GSN_1$, $GSN_2$ ..., $GSN_L$
 ML $\in$ {SVM, RndF} and the required number of rows in the training data $T$
2: **Output:** Estimated class of the $SN$: $SN_{\mathrm{class}}$

3: **while** True **do**
4:  newMessage $\leftarrow$ receiveMessage ( )
5:  $\mathbf{R}^{(t)} \leftarrow$ newMessage.measureRSSI ( )
6:  **if** newMessage.sender $\in GSN_1$, $GSN_2$ ..., $GSN_L$ **then**
7:   $\mathbf{C}_l$ = newMessage.getC ( )
8:   $\mathbf{C}_l[\, t_l \% T, : ] \leftarrow \mathbf{R}^{(t)}$    $\# t_l$ is the time index for $\mathbf{C}_l$
9:   $t_l \leftarrow t_l + 1$
10:  **else**
11:   $SN_{\mathrm{class}}$ = ML.predict (newMessage)  $\#$ classification
12:  **end if**
  $\#$ training phase
13:  **if** time.seconds ( ) % trainPeriod == 0 **then**
14:   ML.train ($\mathbf{C}_1$, ..., $\mathbf{C}_l$ ..., $\mathbf{C}_L$)
15:  **end if**
16: **end while**

---

outliers and noise in the data [40] particularly, in cases with a rather high number of features. Moreover, it is known for being less susceptible to over-fitting.

### D. Algorithm Summary and Error Metric

The steps for updating the training data and estimating the class of any given $SN$ are summarized in Algorithm 1. The algorithm works as follows. When a new message arrives at the network back-end, it first asks for the RSSI measurements from all BSs, i.e., $\mathbf{R}^{(t)}$. Subsequently, the algorithm checks the message sender. If the sender is one of the $GSN$s, it uses $\mathbf{R}^{(t)}$ to update the corresponding class matrix, namely, $\mathbf{C}_l$ (lines 7-9). Otherwise, the message is assumed to be from an $SN$. Hence, the algorithm proceeds to estimate its class denoted as $SN_{\mathrm{class}}$. The ML model is trained periodically, as described in Algorithm 1 (lines 13-15). In particular, the ML model's retraining interval is defined as *trainPeriod*. This period depends on the message transmission rate and the environment. A reasonable empirical value for a Sigfox network in an urban area is one hour. Note that, in Algorithm 1, the number of rows in the training data, $T$, is fixed. This means that whenever a new training message is recorded, it overwrites its oldest counterpart, thus keeping the training dataset updated.

As shown in Algorithm 1, the $SN$ in an unknown class centered at $c$ is classified into one of the $GSN$ known classes. Therefore, one can define the classification error as

$$\xi \; = \; \|\hat{c} - c\|^2, \tag{5}$$

where $\hat{c}$ is the center of the estimated class. Consequently, in case a classification error occurs, the minimum error is defined by the distance $D$ between adjacent classes (i.e., the separation between adjacent $GSN$s). Consequently, one can write

$$\min \xi \; = \; \min \|\hat{c} - c\|^2 \; = \; D. \tag{6}$$

Now, define the radius $r$ as the distance between $GSN$ and the furthest $SN$ in a given class. The classification error decreases when $x$ increases, i.e., when $D \gg \sqrt{3}r$. The influence of $x$ on the classification accuracy, for a given number of classes, is presented in the results section using measurements.

## IV. Regression-based Distance Estimation

Exploiting the benefits of having $GSN$s can be extended by using short-range device-to-device (D2D) communication between $GSN$ and $SN$ nodes. If three $GSN$s are available within the communication range, RSSI values can be used for distance estimation [11,27]. Thus, in addition to classifying $SN$s into classes, it is possible to further estimate their location within the class. This can be achieved by estimating distances between $SN$ and $GSN$s. Even though this step requires at least 3 $GSN$s, it can substantially increase the localization accuracy.

### A. Regression

Instead of building a full radio map, the regression process is a sufficient alternative to estimate distances as a function of RSSI. Basically, a regression process interpolates discrete measurements and generates a continuous output function, which subsequently used to map any RSSI value to its corresponding distance [41,42]. Assume RSSI samples are collected at $J$ different distances from a given $GSN$ node, the regression problem can be written as

$$\operatorname*{argmin}_{a} \left\{ \left( \sum_{j=1}^{J} \Psi(RSSI_j) - d_j \right)^2 \right\}, \tag{7}$$

where $d_j$ is the distance of the $j$-th location from the $GSN$. Assuming an $n$-th degree polynomial regression function, $\Psi(RSSI_j)$ is expressed by

$$\begin{aligned} \Psi(RSSI_j) &= a_0 + a_1 RSSI_j + a_2 RSSI_j^2 + \\ &\quad \cdots + a_n RSSI_j^n, \end{aligned} \tag{8}$$

where $a_i$ (with $i = 0, 1, 2, ..., n$) is the $i$-th coefficient and $j = 1, 2, ..., J$. The motivation behind choosing the polynomial regression is its evaluation simplicity and the fact that polynomials dominate the interpolation theory [43]. Once $\Psi(RSSI_j)$ is defined, the distance estimated from a new $RSSI_t$ measurement is written as

$$\hat{d} = \operatorname*{argmin}_{d_j} \left\{ \left( \Psi^{-1}(d_j) - RSSI_t \right)^2 \right\}, \tag{9}$$

Here, it is worth noting that $\Psi^{-1}(d_j)$ is a continuous function. Hence, $\hat{d}$ can take any non-negative value.

### B. Algorithm Summary and Error Metric

Algorithm 2 presents the details of estimating the location of a given $SN$ when D2D communication is enabled. Once $SN_{\text{class}}$ is estimated using Algorithm 1, the position within the class, $SN_{\text{location}}$, can be estimated using Algorithm 2. Let $K$ denotes the number of $GSN$ nodes which are in *range* of the target $SN$. The function *inRange*$(SN, GSN_i)$ simply returns one if $GSN_i$ is able to receive messages from $SN$

---

**Algorithm 2** Localizing $SN$s with D2D communication

1: **Input:** $GSN_1$, $GSN_2$ ..., $GSN_L$ with their locations $c_1$, $c_2$, ..., $c_L$, respectively
   The $SN$ to be localized
2: **Output:** Estimated location of the $SN$: $SN_{\text{location}}$

3: $K \leftarrow \sum_{i=1}^{L}$ inRange $(SN, GSN_i)$
4: **if** $K \geq 3$ **then**
5:      Use (9) to compute $\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_K$
6:      $SN_{\text{location}} = \text{multilateration}(\hat{d}_1, \hat{d}_2, ..., \hat{d}_K, c_1, c_2, ..., c_K)$
7: **else**
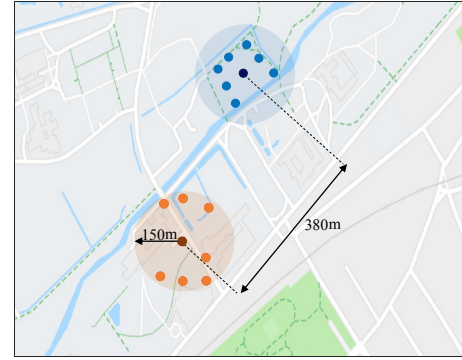8:      $SN_{\text{location}} = SN_{\text{class}}$      #from Algorithm 1
9: **end if**

---



Fig. 4: A map showing the nodes' positions in the university campus with $D \approx \frac{3\sqrt{3}}{2}r$. Dark orange and dark blue are $GSN$s.

and zero otherwise. If $K \geq 3$, the algorithm uses (9) to find $\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_K$ and subsequently performs the multilateration process. However, if $K < 3$, the multilateration process is not applicable. Hence, it returns $SN_{\text{class}}$ estimated in Algorithm 1.

In practice, the function $\Psi(RSSI_j)$ does not fit the data perfectly. Therefore, an estimation error is expected. The average localization error is given by

$$\xi = \sqrt{\frac{1}{KT_s} \sum_{t=1}^{T_s} \sum_{k=1}^{K} |\hat{d}_{tk} - d_{tk}|^2}, \tag{10}$$

where $T_s$ is the number of RSSI measurements used to estimate the distance between one $GSN$ and the given $SN$ node. Moreover, $\hat{d}_{tk}$ in (10), represents the estimated distance from the $k$-th $GSN$ using the $t$-th RSSI measurement.

## V. Experimental Results

In this section, we present our experimental setup. Subsequently, results using IoT nodes that support communication over both Sigfox and TD-LAN are detailed.

### A. Experiments setup

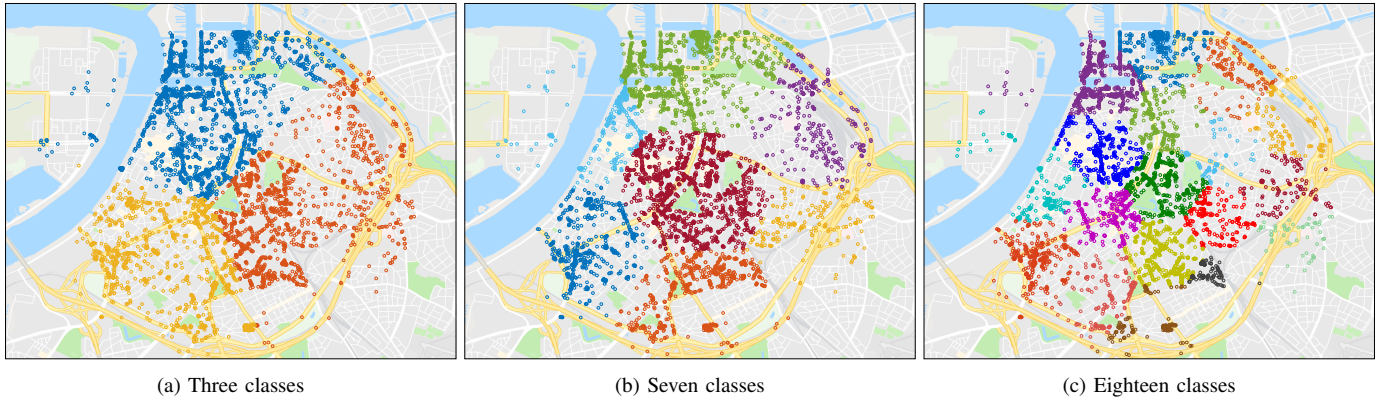The experiments have been conducted in two different scenarios, a university campus, and a city center.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2931628, IEEE Internet of Things Journal

7



(a) Three classes        (b) Seven classes        (c) Eighteen classes

Fig. 5: Maps showing the position of nodes divided into three, seven, and eighteen classes.

*1) University campus:* In the second scenario, 16 nodes equipped with Sigfox modems are positioned in KU Leuven's Arenberg campus with $D \approx 2.5r$, as shown in Fig. 4. Each node sent 100 messages from which the RSSI values are measured at the 3 BSs that have received the messages. In Fig. 4 the dark orange and dark blue nodes are $GSN$s. Moreover, D2D communication is enabled over TD-LAN network. Using D2D communication, we aim to further increase the localization accuracy within classes using the RSSI measurement collected when nodes communicate with each other.

*2) City center:* Fig. 5 presents the positions, in the city of Antwerp, from which Sigfox messages are collected [2]. In particular, over 14000 Sigfox messages were collected using 20 cars of the Belgian postal service [2] for two months and a half. Messages are sent from a region with a radius of 2.5 km and are received by 58 Sigfox BSs, in total. Figures 5a, 5b, and 5c show the virtual splitting of the city into 3, 7, and 18 classes with $D$ equals 1830 m, 1600 m, and 1300 m, respectively. The dataset contains the GPS coordinates, base station ID, RSSI of each Sigfox message. In order to resemble $GSN$ in this dataset, we assume that all measurements collected within a 100 m radius from the center of each class represent one $GSN$. This provides 65, 45, and 25 messages per class to be used as training data in the case of three, seven, and eighteen classes, respectively. Subsequently, the rest of the measurements are used as testing data to be classified. It is worth noting that, in our analysis, uniform test sets have been used by making sure that all training and test sets are of the same size for each class.

### B. Sigfox Communication - classification

In this scenario, we investigate the localization accuracy using Sigfox network. In particular, we focus on a star IoT network scenario in which D2D is not available. In the following, we thoroughly investigate different design parameters that affect the classification accuracy. In this subsection, we mainly use the dataset collected in the city of Antwerp [2], unless mentioned otherwise.

*1) Number of features:* Fig. 6 presents the performance of RndF, SVM, and $k$-NN with a different number of features. As shown in the figure, using the proposed algorithm, RndF outperforms $k$-NN presented in [2] with $k = 11$. Moreover, it
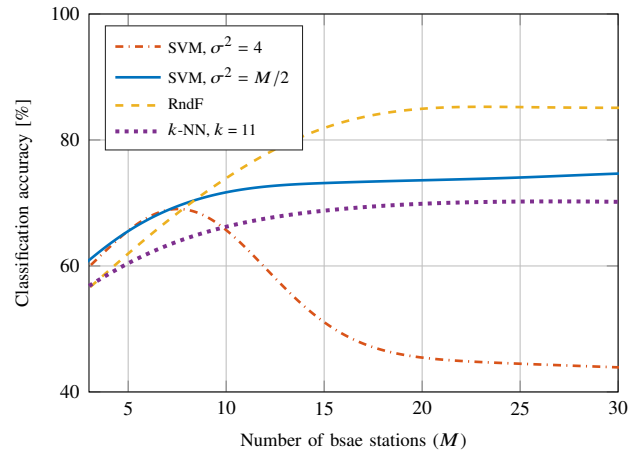


Fig. 6: Classification accuracy versus the number of BSs (features), 7 classes, $D = 1.6$ km, $r = 600$ m, and 40 training messages from each $GSN$ node.
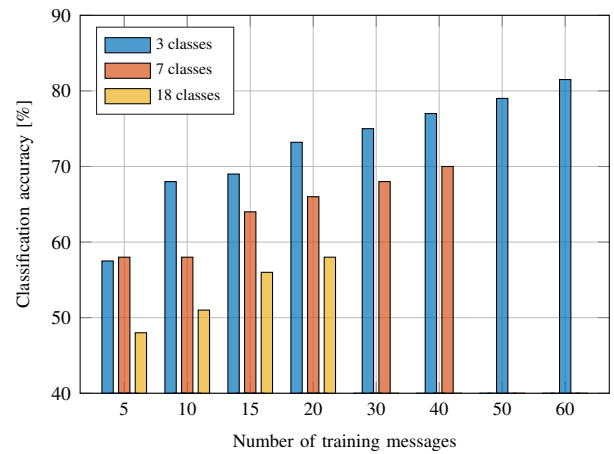


Fig. 7: Classification accuracy versus the number of training messages from each $GSN$ node for three, seven, and eighteen classes with $D = \sqrt{3}r$.

provides better performance than SVM when the number of features, namely, BSs, is above 8. The reason behind this trend is twofold; first, it is well-known that SVM works better with separable datasets [39,44]. However, in long-range UNB IoT network, increasing the number of features means adding BSs that are more likely to be far away (> 5 km). Hence, having

**(a) Three classes**

| True label \ Predicted label | C1 | C2 | C3 |
|---|---|---|---|
| C1 | 0.68 | 0.24 | 0.08 |
| C2 | 0.07 | 0.84 | 0.09 |
| C3 | 0.11 | 0.12 | 0.77 |

**(b) Seven classes**

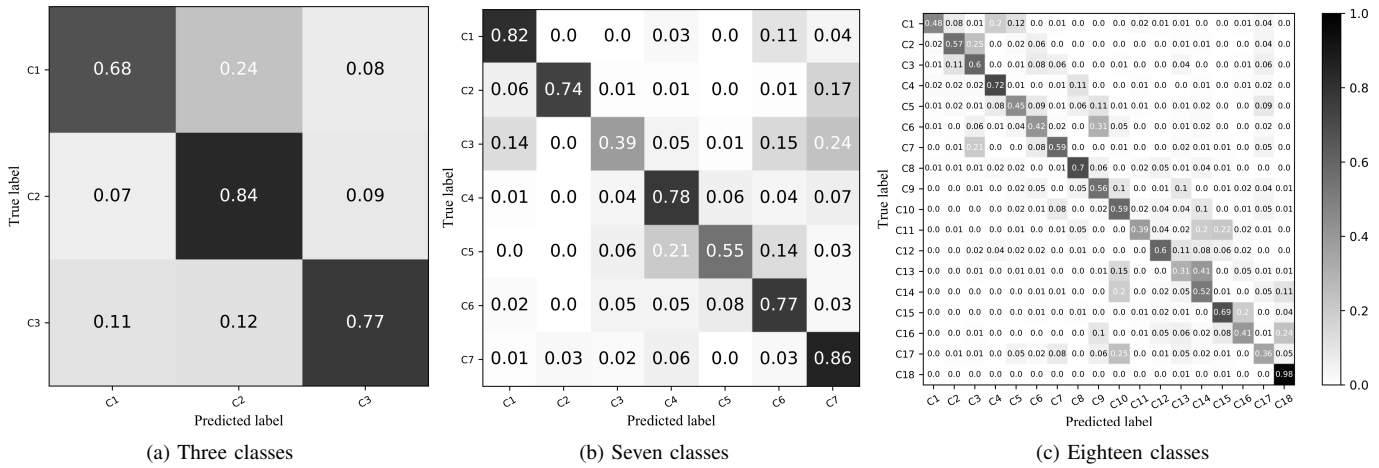| True label \ Predicted label | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
|---|---|---|---|---|---|---|---|
| C1 | 0.82 | 0.0 | 0.0 | 0.03 | 0.0 | 0.11 | 0.04 |
| C2 | 0.06 | 0.74 | 0.01 | 0.01 | 0.0 | 0.01 | 0.17 |
| C3 | 0.14 | 0.0 | 0.39 | 0.05 | 0.01 | 0.15 | 0.24 |
| C4 | 0.01 | 0.0 | 0.04 | 0.78 | 0.06 | 0.04 | 0.07 |
| C5 | 0.0 | 0.0 | 0.06 | 0.21 | 0.55 | 0.14 | 0.03 |
| C6 | 0.02 | 0.0 | 0.05 | 0.05 | 0.08 | 0.77 | 0.03 |
| C7 | 0.01 | 0.03 | 0.02 | 0.06 | 0.0 | 0.03 | 0.86 |

**(c) Eighteen classes**

Fig. 8: The confusion matrices for the case of three, seven, and eighteen adjacent classes with training data of 60, 40, and 20 messages from each $GSN$, respectively, and $D = \sqrt{3}r$.

similar RSSI measurements makes the classes inseparable. RndF, on the other hand, does not have such constraint, since similar features and samples are not used to extend trees. Secondly, SVM with a Gaussian kernel requires enough examples in order to find the optimal separating hyperplane in a higher dimensional space [31,44]. However, in Fig. 6, the training messages are fixed to 40, from each $GSN$ node. The performance of SVM and RndF with a different number of training messages and a fixed number of features is presented in Fig. 11.

Furthermore, Fig. 6 illustrates the effect of $\sigma^2$ on the performance of the SVM classifier. As shown in the figure, selecting a fixed $\sigma^2 = 4$ results in lower classification accuracy with $M$, for $M > 8$. This is because SVM over-fits on the training data when the number of features is larger than 8. However, this over-fitting is avoided by using an adaptive $\sigma^2$ equals to $M/2$. In fact, the result presented in Fig. 6 extends the one reported in [1], by giving deep insights on when and which ML algorithm to use.

*2) Number of training messages:* In long-range IoT networks, the number of messages a node can send is limited. For instance, in Sigfox network each node can send 140 messages a day, with an average of one message every 10 minutes. Therefore, it is crucial to estimate the $SN$ class using as minimum training messages as possible. In Fig. 7, we present the performance of RndF with three, seven, and eighteen classes using a different number of messages. As shown in the figure, having more training messages is beneficial for the classification accuracy. However, the trade-off here is the long time required to collect training messages. Nonetheless, it is worth noting that, collecting training messages is a cumulative process, therefore, the waiting time will be overcome in the long-term.

*3) Number of classes:* The number of classes, i.e., partitions, is a crucial parameter as it affects both the classification accuracy and localization accuracy. As shown in Figs. 8a, 8b, and 8c, using RndF with 100 trees, for the case when the city is partitioned into three regions, the classification accuracy is much higher in comparison with the cases with seven and
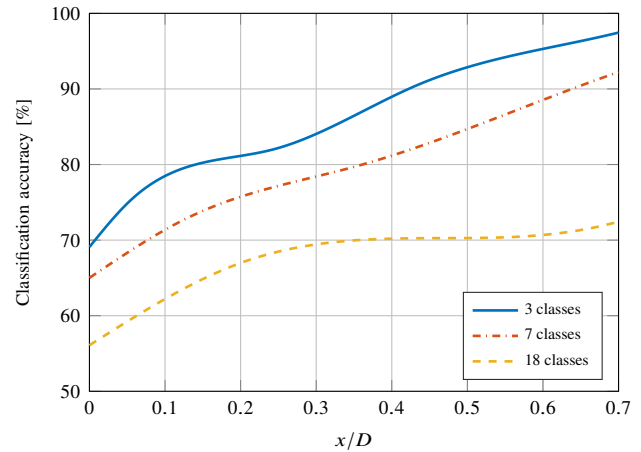
Fig. 9: Classification accuracy using RndF with different $x/D$ and with 20 tanning messages from each $GSN$ node.

eighteen classes. However, the case with three classes provides less location information when compared with the case of eighteen classes. Accordingly, the trade-off is the classification accuracy versus the localization accuracy. In fact, this design parameter is mainly application dependent. Therefore, for applications such as packages tracking, it's sufficient enough to place $GSN$ on post offices checkpoints rather than placing extra $GSN$ all over the city. Consequently, better classification accuracy can be achieved.

*4) Spacing between adjacent classes:* In order to quantify the impact of the spacing between adjacent classes, $x$, on the performance of the proposed algorithm, the classification accuracy as a function of $x/D$ is presented in Fig. 9. Notice that $x \in [0, D)$, hence, dividing it by $D$ gives us the $x$-axis $\in [0, 1)$. As shown in Fig. 9, increasing the spacing between adjacent classes improves the classification accuracy, since the classes borders, which mostly cause classification errors, are eliminated. Hence, it is easier to distinguish messages. However, the cost here is the localization coverage region. In fact, since $D$ is fixed, increasing $x$ results in a region that is not fully included in the localization problem. Moreover, Fig.

TABLE II: Training time of ML algorithms

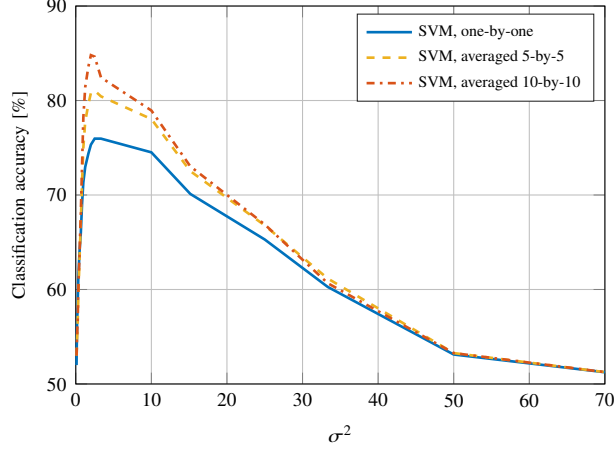| Classifier | # Train. messages | Train. time [ms] |
|---|---|---|
| RndF, 100 trees | 40 | 200 |
|  | 400 | 600 |
| RndF, 200 trees | 40 | 400 |
|  | 400 | 1200 |
| SVM | 40 | 36 |
|  | 400 | 460 |
| $k$-NN | 40 | 10 |
|  | 400 | 30 |



Fig. 10: The classification accuracy as a function of $\sigma^2$ for the measurements conducted on the university campus.

9 illustrates the point at which the classes become connected, i.e., $D = \sqrt{3}r$. At this point, $x$ approaches zero and the classification accuracy is minimal as the number of *SN*s at the classes' borders is maximal. Therefore, the minimum number of messages required to achieve a certain accuracy also increases. This is due to the fact that more training data is needed to learn the class of each node, particularly for *SN*s at the border of the classes.

*5) Training time:* Algorithm 1 is set to retrain the model periodically with a retrain interval of one hour. Therefore, it is vital to investigate the training time needed. Considering the case of 7 classes, Table II presents an estimate of the training time with a different number of training messages, from each *GSN* node. In particular, Table II shows that $k$-NN requires less time for training compared to both SVM and RndF. However, it is shown that the training time for both SVM and RndF is in the range of hundreds of milliseconds (ms). Such a training time is rather short in comparison with the message transmission rate in UNB IoT networks, which is one message per few minutes.

*6) RSSI averaging:* As shown in Fig. 6, in case of a low number of features, SVM outperforms RndF. Here, we focus on this particular case using the measurements conducted on the university campus. Recall that in university campus we have 3 BSs and 100 messages. When SVM is used, we first need to choose the value of $\sigma^2$ for the RBF kernel. In Fig. 10, we present the classification accuracy of 100 messages with $\sigma^2$. As shown in the figure, the best classification accuracy is achieved at $\sigma^2 \approx 4$. Accordingly, we choose a kernel function with $\sigma^2 = 4$ for our SVM. Fig. 11 presents the
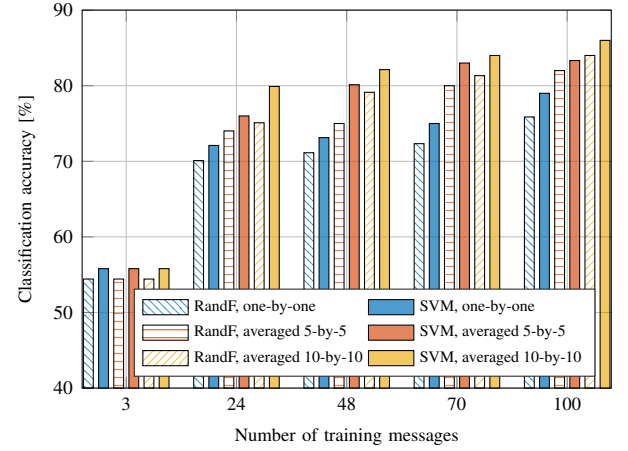


Fig. 11: Classification accuracy of both SVM and RndF algorithms using the university campus dataset. The number of features is fixed and equals to 3.
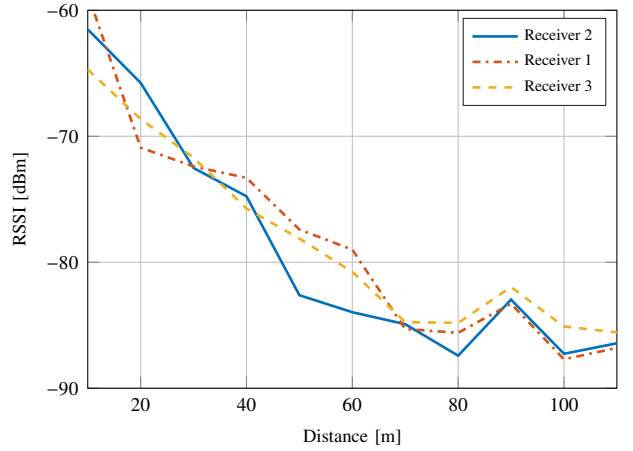


Fig. 12: Average RSSI values at three different receivers with a separation distance of 10 m.

effect of the size of the training data on the classification accuracy for both SVM and RndF. As shown in the figure, SVM outperforms RndF since enough training examples are available for the dataset with three features used. Maximum classification accuracy of 78% is obtained when classifying messages one-by-one. However, by averaging the RSSI values, for both *GSN* and *SN* nodes, we overcome the effect of small scale temporal fading. Hence, better accuracy is achieved. In particular, averaging the messages' RSSI 10 by 10 brings the classification accuracy to 87% using our SVM classifier, as illustrated in Fig. 11. The cost of this improvement is the delay required to receive ten messages before making an estimation.

A wrongly classified message implies that the estimated location of the *SN* is in another class. Consequently, the corresponding distance error is the distance from the estimated class's center to the correct one. To further improve the accuracy, the impact of enabling D2D communication between nodes is presented in the following subsection.

### C. Device-to-Device enabled - Regression

When the density of *GSN* nodes is high enough, and in case TD-LAN is enabled, RSSI can be measured in the near
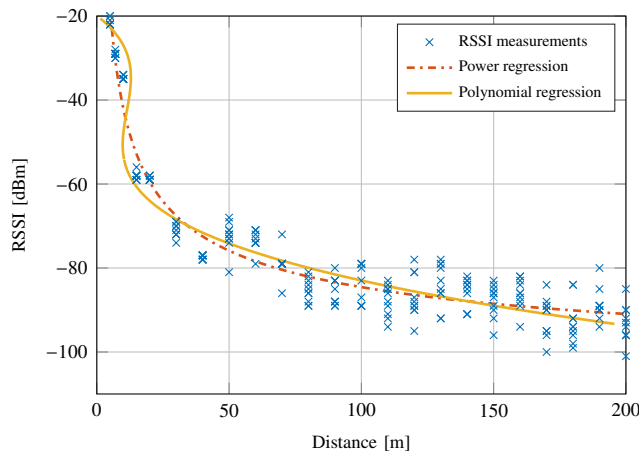
Fig. 13: Power regression and polynomial regression of RSSI measurements collected in steps of 10 m.
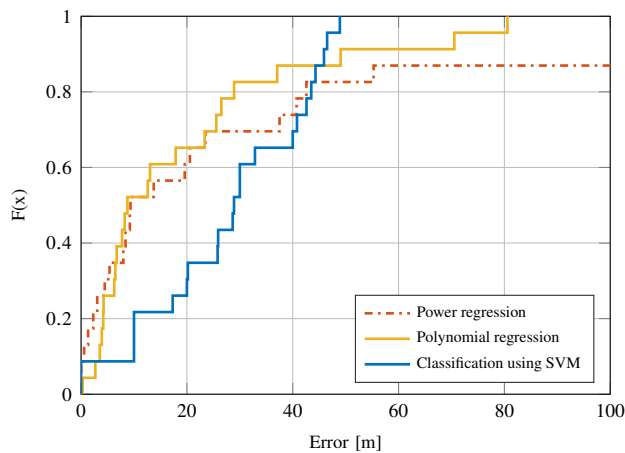


Fig. 14: Empirical CDF: performance of regression vs. fingerprinting for the measurements conducted on the university campus.

zone where the RSSI-distance curve has sufficient resolution. To estimate the distance using RSSI, we first collect discrete measurements and subsequently, use a regression process to fit RSSI-distance curve. To this end, we used three nodes in receiving mode, placed on a line with 10 m inter-distance. Then, we placed transmitters on a line, perpendicular to the line made by the receivers, at distances from 10 m to 200 m with a spacing of 10 m. It is worth mentioning that, this setup is not optimal. An optimal setup requires receiving nodes to be placed on the corners of the considered area to have positions diversity. Hence, the obtained accuracy can be considered as a lower bound. The measurements of this setup are illustrated in Fig. 12. The interpolation of RSSI measurements using power series and polynomial regression is shown in Fig. 13. In particular, we used a two-term power series regression and a third-degree polynomial regression. This results in the curves shown in Fig. 13, which are used for estimating the distances in the test phase.

Finally, Fig. 14 presents the cumulative distribution function (CDF) of the localization error for both power series and polynomial regression. It can be concluded that similar performance is obtained from both methods. Furthermore, for the same setup, the performance of fingerprinting localization is presented. While distance estimation outperforms fingerprinting localization for errors lower than 40 m, fingerprinting localization error is always bounded by the distances between *GSN*s. Distance estimation at distances further than 200 m from the receiver results in high errors (>60 m). Therefore, it is recommended to limit the use of RSSI ranging-based localization to classes with a maximum radius of 200 m.

## VI. CONCLUSION

This paper presents localization methods for UNB IoT networks by exploiting the available *GSN*s to estimate the location of other *SN*s. Firstly, RSSI measurements from *GSN*s are used as a source of training data to classify other *SN* into location-based classes. By using only communication over Sigfox, this solution offers the localization service for free, since *GSN*s have to send their own messages anyway. Secondly, it has been shown that using D2D communication, regression-based distance estimation improves the localization accuracy of the classified *SN*s. However, this requires extra *GSN* nodes. Using measurement based dataset, we introduced the design parameters which affect the performance of the proposed approach, such as the number of classes, the spacing between adjacent classes and the number of features used. Furthermore, the effect of each of these parameters, along with the trade-offs introduced with them, has been thoroughly investigated.

## ACKNOWLEDGMENT

## REFERENCES

[1] H. Sallouha, A. Chiumento, and S. Pollin, "Localization in Long-range Ultra Narrow Band IoT Networks using RSSI," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[2] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, "Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, p. 13, 2018.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[4] R. C. Shit, S. Sharma, D. Puthal, and A. Y. Zomaya, "Location of things (lot): A review and taxonomy of sensors localization in iot infrastructure," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2028–2061, 2018.

[5] H. Sallouha, B. Van den Bergh, Q. Wang, and S. Pollin, "ulora: Ultra low-power, low-cost and open platform for the lora networks," in *4th ACM Workshop HotWireless '17*, pp. 43–47, 2017.

[6] Sigfox: *http://www.sigfox.com*, [online] Accessed on: Jan. 2019.

[7] Weightless-P: *http://www.weightless.org/*, [online] Accessed on: Jan. 2019.

[8] L. alliance: *http://lora-alliance.org*, [online] Accessed on: Jan. 2019.

[9] NB-IoT: *https://goo.gl/DaUHkV*, [online] Accessed on: Jan. 2019.

[10] S. Gezici *et al.*, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE signal processing magazine*, vol. 22, no. 4, pp. 70–84, 2005.

[11] S. Yiu *et al.*, "Wireless rssi fingerprinting localization," *Signal Processing*, vol. 131, pp. 235–244, 2017.

[12] H. Elsawy, W. Dai, M.-S. Alouini, and M. Z. Win, "Base station ordering for emergency call localization in ultra-dense cellular networks," *IEEE Access*, vol. 6, pp. 301–315, 2018.

[13] M. Z. Win, Y. Shen, and W. Dai, "A theoretical foundation of network localization and navigation," *Proceedings of the IEEE*, vol. 106, no. 7, pp. 1136–1165, 2018.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2019.2931628, IEEE Internet of Things Journal

11

[14] Z. Liu, W. Dai, and M. Z. Win, "Mercury: An infrastructure-free system for network localization and navigation," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1119–1133, 2018.

[15] M. Z. Win, A. Conti, S. Mazuelas, Y. Shen, W. M. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 56–62, 2011.

[16] M. Z. Win, W. Dai, Y. Shen, G. Chrisikos, and H. V. Poor, "Network operation strategies for efficient localization and navigation," *Proceedings of the IEEE*, vol. 106, no. 7, pp. 1224–1254, 2018.

[17] M. Z. Win, F. Meyer, Z. Liu, W. Dai, S. Bartoletti, and A. Conti, "Efficient multisensor localization for the internet of things: Exploring a new class of scalable localization algorithms," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 153–167, 2018.

[18] D. Zhang, S. Zhao, L. T. Yang, M. Chen, Y. Wang, and H. Liu, "Nextme: Localization using cellular traces in internet of things.," *IEEE Trans. Industrial Informatics*, vol. 11, no. 2, pp. 302–312, 2015.

[19] D. Macagnano, G. Destino, and G. Abreu, "Indoor positioning: A key enabling technology for iot applications," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pp. 117–118, IEEE, 2014.

[20] R. C. Shit, S. Sharma, D. Puthal, and A. Y. Zomaya, "Location of things (lot): A review and taxonomy of sensors localization in iot infrastructure," *IEEE Communications Surveys & Tutorials*, 2018.

[21] M. Kwak *et al.*, "An energy-efficient and lightweight indoor localization system for internet-of-things (iot) environments," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, p. 17, 2018.

[22] J. Zhang, G. Han, N. Sun, and L. Shu, "Path-loss-based fingerprint localization approach for location-based services in indoor environments," *IEEE Access*, vol. 5, pp. 13756–13769, 2017.

[23] P. Fonseka and K. Sandrasegaran, "Indoor localization for iot applications using fingerprinting," in *IEEE World Forum on Internet of Things, WF-IoT 2018*, 2018.

[24] Q. Song *et al.*, "Csi amplitude fingerprinting-based nb-iot indoor localization," *IEEE Internet of Things Journal*, vol. 5, no. 3, 2018.

[25] S. Bozkurt *et al.*, "A comparative study on machine learning algorithms for indoor positioning," in *International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–8, IEEE, 2015.

[26] W. Farjow, A. Chehri, M. Hussein, and X. Fernando, "Support vector machines for indoor sensor localization," in *2011 IEEE Wireless Communications and Networking Conference*, pp. 779–783, IEEE, 2011.

[27] S. Kumar *et al.*, "Gaussian process regression for fingerprinting based localization," *Ad Hoc Networks*, vol. 51, pp. 1–10, 2016.

[28] R. C. Shit *et al.*, "Ubiquitous localization (ubiloc): A survey and taxonomy on device free localization for smart world," *IEEE Communications Surveys & Tutorials*, 2019.

[29] R. C. Shit, S. Sharma, D. Puthal, and S. S. Tripathi, "Probabilistic rss fingerprinting for localization in smart platforms," in *2018 International Conference on Information Technology (ICIT)*, pp. 254–259, IEEE, 2018.

[30] R. Zhou, X. Lu, P. Zhao, and J. Chen, "Device-free presence detection and localization with svm and csi fingerprinting," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7990–7999, 2017.

[31] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 981–994, 2008.

[32] Y. Liy, Z. He, Y. Li, H. Xu, L. Pei, and Y. Zhang, "Towards location enhanced iot: Characterization of lora signal for wide area localization," in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, pp. 1–7, IEEE, 2018.

[33] X. Wang, Z. Qian, X. Wang, and L. Huang, "Robust localization for cognitive iot via the mobile anchor node based on the diameter-varying spiral line," *IEEE Access*, vol. 7, pp. 28487–28497, 2019.

[34] H. Sallouha, M. M. Azari, and S. Pollin, "Energy-constrained uav trajectory design for ground node localization," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, 2018.

[35] T. Janssen, M. Weyn, and R. Berkvens, "Localization in low power wide area networks using wi-fi fingerprints," *Applied Sciences*, vol. 7, no. 9, p. 936, 2017.

[36] M. Aernouts *et al.*, "A comparison of signal strength localization methods with sigfox," in *15th WPNC*, pp. 1–6, IEEE, 2018.

[37] TD: *https://github.com/Telecom-Design/Documentation_TD*, [online] Accessed on: Jan. 2019.

[38] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *Telecommunications (ICT), 2016 23rd International Conference on*, pp. 1–6, IEEE, 2016.

[39] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[40] I. Ahmad, M. Basheri, M. Iqbal, and A. Raheem, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, 2018.

[41] A. Chiumento *et al.*, "Gaussian process regression for csi and feedback estimation in lte," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, pp. 1440–1445, June 2015.

[42] A. Chiumento *et al.*, "Adaptive CSI and feedback estimation in LTE and beyond: a Gaussian process regression approach," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, p. 168, Jun 201.

[43] J. Yang and Y. Chen, "Indoor localization using improved rss-based lateration methods," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6, IEEE, 2009.

[44] G. C. Cawley and N. L. Talbot, "Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters," *Journal of Machine Learning Research*, vol. 8, no. Apr, pp. 841–861, 2007.

**Hazem Sallouha** (GS'16) received the B.Sc. degree in electrical engineering from IU Gaza, Palestine, in 2011, the M.Sc. degree in electrical engineering from Jordan University of Science and Technology (JUST), Jordan, in 2013. Currently, he is a Ph.D. candidate at the Department of Electrical Engineering, KU Leuven, Belgium. His main research interests are localization techniques with unmanned aerial vehicles, Internet of things networks, and machine learning algorithms for localization.

**Alessandro Chiumento** (S'12–M'16) received his Ph.D. in cellular network management from Imec, Leuven, Belgium, in 2015. He subsequently worked as a postdoctoral researcher at KU Leuven on massive machine-to-machine communication, channel prediction, very dense networks, and the application of machine learning to theoretical problems in telecommunication and information management. He is currently an MSCA EDGE fellow in CONNECT, Dublin, Ireland and is working on the project "Internet of Complex Things", which aims to develop a formal framework for causality in distributed wireless networks and to provide adaptive run-time control methods for vast and dense IoT sensor networks.

**Sreeraj Rajendran** (GS'16) is a post-doctoral researcher in the Networked Systems group of KU Leuven. He received his Ph.D. from KU Leuven, Belgium in 2019 and masters in communication and signal processing from IIT, Bombay, India in 2013. He worked as a senior design engineer in the baseband team of Cadence (Tensilica) after completing his masters. He also worked as an ASIC verification engineer in Wipro Technologies during 2007-2010. His main research interests include machine learning algorithms for wireless spectrum awareness and low power wireless sensor networks.

**Sofie Pollin** (S'03–M'06–SM'14) received the Ph.D. degree (Hons.) from KU Leuven, Leuven, Belgium, in 2006. From 2006 to 2008, she was with the University of California at Berkeley, Berkeley, CA, USA, where she was involved in the research on wireless communication, energy-efficient networks, cross-layer design, coexistence, and cognitive radio. In 2008, she joined the Green Radio Team, Imec, as a Principal Scientist. She is currently an Associate Professor with the Electrical Engineering Department, KU Leuven. Her current research interests include networked systems that require networks that are ever denser, heterogeneous, battery-powered, and spectrum constrained. Dr. Pollin is a BAEF and Marie Curie Fellow.