Tracking objects within a smart home

Article in Expert Systems with Applications · July 2018

DOI: 10.1016/j.eswa.2018.07.009

4 authors:

Frédéric Bergeron
Université de Sherbrooke

Kévin Bouchard
Université du Québec à Chicoutimi

Sebastien Gaboury
Université du Québec à Chicoutimi

Sylvain Giroux
Université de Sherbrooke

Tracking objects within a smart home

Frédéric Bergeron[a,*], Kevin Bouchard[b], Sébastien Gaboury[b], Sylvain Giroux[a]

[a] *Université de Sherbrooke, 2500, boul. de l'Université, Sherbrooke (Qubec) Canada J1K 2R1*
[b] *Université du Québec à Chicoutimi, 555, boulevard de l'Université, Chicoutimi (Qubec) Canada G7H 2B1*

Abstract

This paper presents a novel indoor tracking system built with common data mining techniques on radio frequency identification (RFID) tags readings. The system allows tracking of several objects in real-time in a smart home context and is a building block toward the deployment of an expert system to enable aging in place through technology. The indoor localization is modelled as a classification problem, instead of a regression problem as commonly seen in the literature. The paper is divided in two parts. The first one focuses on the ground truth collection that led to the model construction. The second part focuses on the filters that were designed to enable this model to be used in real-time in the smart home as a tracking software. Results from the first part show that most classifiers perform well on the static positioning of RFID tags task, with a random forest of 100 trees performing best at 97% accuracy and 0,974 F-Measure. However, collecting data to train the classifier is a long and tedious process. Results from the second part indicate that the accuracy of the random forest drops significantly when confronted with human interference. With the help of some filters, the tracking accuracy of objects can still be as high as 75%. Those results confirm that using passive RFID tags for an indoor tracking system is viable. Our system is easy to deploy and more flexible than trilateration or fingerprinting systems.

Keywords: RFID, Smart home, Data mining, Decision trees, Indoor tracking system

1. Introduction

In the recent years, technological advancements have led to the emergence of connected objects that can interact together and enable applications that would not have been possible before. Connected objects are commonly regrouped within a housing to form a smart home, which, in turn, enables many exciting possibilities Giroux et al. (2009); Chan et al. (2009). Western government are currently interested in using smart homes to enable the aging population to stay at home. For instance, smart homes can be seen as a way to secure the environment of the elders, or other persons, living with cognitive impairment to age-in-place, alone or with an informal caregiver Zulas et al. (2014). This would reduce the pressure on health systems by delaying institutionalization, while improving the quality of life of the elderly.

To build this type of smart homes, it is required to install a wide variety of sensors and actuators interacting together to provide contextualized assistance (conscious intelligence) and implement basic safety rules (reflex intelligence). While it can be argued that the technologies to achieve a complete and functional smart home to enable aging-in-place are readily available, the artificial intelligence to build an expert system providing useful services is still inadequate. There are many factors explaining the flaws of current artificial intelligence in smart homes, but one of the most important is the low-level of information that can be extracted from distributed sensors. In this context, the sensors only provide raw data which must be transformed into high-level information in order to be useful for an expert system. In many cases, this task is straightforward. For instance, an electronic contact can assert if a door is opened or closed with a simple rule (if 0, then door_open). In fact, most non-invasive sensors only provide Boolean readings with time stamps. This apparent simplicity is, however, deceitful and results in many of the challenges faced by AI researcher in smart homes. Indeed, the recognition of the resident's activities is difficult to achieve by aggregating those low-level features Cook & Krishnan (2014). This is why existing approaches can usually recognize high-level activities like *cooking dinner* or *dressing the table* but cannot tell what is being cooked or how many persons will share the meal Mehr et al. (2016); Krishnan & Cook (2014).

One major difficulty to improve the granularity of the activity recognition lies in the limitations of actual indoor tracking systems. Tracking systems are important in smart home as they facilitate activity recognition, which is in turn used by monitoring systems to predict danger or provide contextualized assistance Tesoriero et al. (2010). There are two primary types of tracking systems: those that track inhabitants Calderoni et al. (2015) and those that track objects Bouchard et al. (2013). For both types, many technologies have been proposed through the years. The most widely known is certainly the GPS and its competitors Misra & Enge (2006). They rely on a certain number of satellites to compute the position of a receptor at any point on earth using time differentiation. They are commonly used

*Corresponding author
Email addresses: frederic.bergeron2@usherbrooke.ca (Frédéric Bergeron), kevin.bouchard@uqac.ca (Kevin Bouchard), sebastien.gaboury@uqac.ca (Sébastien Gaboury), sylvain.giroux@usherbrooke.ca (Sylvain Giroux)

in car navigation and tracking. While they are great for most outdoor positioning application, they lack in precision for indoor positioning. Moreover, the GPS signal is greatly affected by structures like walls and roof. In fact, GPS signal is often not available at all inside most building.

For indoor tracking of the residents, the well-known WI-FI Crow et al. (1997) technology has been the most popular choice in recent years. It offers the advantage that most residential building nowadays are already equipped with one of more WI-FI routers. The main weakness of WI-FI is that the signal often vary due to numerous interference sources found in a house. When using the received signal strength indication (RSSI) for localization, there is also the multipath propagation problem that arises when a same signal can take different paths to reach a given tag. Another technology that is often used for indoor localization of residents in the same fashion as WI-FI is the RFID Joshi et al. (2014). Depending on the band it is transmitting on, the signal power and the type of tags, its signal has a range of a few centimeters to several kilometers. The main weaknesses are also signal interference and multipath propagation. The false positive readings can also be problematic for some applications Ma et al. (2018). An important difference between WI-FI and RFID is the receptor, where passive RFID tags are much more smaller and cheaper their WI-FI counterparts.

Other radio frequency technologies used for the task of indoor tracking include (but are not limited to) Bluetooth Rossey et al. (2016), NFC and ZigBee Ransing & Rajput (2015). They all offer similar performances and limitations. NFC is mainly used for very short distance communication and for identification cards. It lacks range for continuous tracking, but is good to detect when a tag passes near a predetermined position. Bluetooth and ZigBee are both used for medium range communication. As tracking technology, they offer similar performances and limitations as RFID and WI-FI. One limitation of the Bluetooth technology comes from the pairing mechanism, where a master node can have at most 7 slave nodes.

Table 1 resumes the main characteristics for the principal radio frequency technologies in use. The range column shows the range we can expect under normal circumstances, as advertised by their various developers (LitePoint Corporation (2013); Wikipedia contributors (2018); Martin Woolley (2018); ZigBee Alliance (2013)). The precision is left imprecise to better show the scale at which localization is performed U.S. Air Force (2017); Yim (2008); Altini et al. (2010); Ou et al. (2017). The aim of this paper is not to achieve the best precision but to show another, more stable and easier to use, approach to the problem.

The choice of technology often depends on the tracking context and on the objectives of the tracking. User's location is often used for context aware systems that adapt and personalize services to the specific personChahuara et al. (2013). In that context, wearable technology such as a smart watch Morganti et al. (2012), a smart band Knighten et al. (2015) or a smartphone Fahim et al. (2012) are often coupled with the radio-frequency technology to gather the data. The main challenges in that case revolve around the energy consumption. Indeed, more data usually translate with more accuracy, but also into a higher transmission frequency resulting in a higher battery consumption on the wearable. It is also possible to use active tags, which are simply tags embedding their own internal battery power communicating with a fixed computer gathering the information (Bluetooth beacons Chawathe (2008), active RFID Chai et al. (2017)). In that case, the wearable's battery limitation is eliminated, but there is a need to monitor the battery level of the active tags and maintain the system over time.

Due to those constraints, non RF technologies are also in use. Systems based on technologies other than tags include ultrasonic sensors and microphones. Still, they have other weaknesses to consider. They can be mislead by pets or even wind, for instance. Perhaps the most popular tracking is still the classic motion PIR (Passive InfraRed sensor). Those sensors are cheap and simple to use, but on the other hand they tend to occasionally misfire and often lack precision for complex applications. Still, they are part of many rule-based systems due to their simplicity.

The second type of tracking, concerns the tracking of daily life objects. Object tracking is another non-intrusive source of information that can be exploited in a smart home, which provides a much richer data set with temporal continuum. It is a major improvement from the simple set of discrete Boolean information discussed before Cook & Krishnan (2014) as it embed both high-level spatial and temporal information about the tools used by the resident. While the tracking of the resident is a topic that has been vastly explored Majeed & Arif (2016); Kwok et al. (2006); Cheng et al. (2013); Hutabarat et al. (2016); He et al. (2014); Morato et al. (2014); Calderoni et al. (2015), the story unfolds very differently for the indoor tracking of the objects used in the daily activities. The widely used technologies for resident tracking (and robot tracking de Sá et al. (2016)), cannot be directly used for the tracking of daily life objects. Indeed, all these technologies require some sort of device to be installed on the objects (either a wearable, an antenna or a tag). Installing wearable on every tangible object of a smart home is obviously not economically viable, and BLE (Bluetooth low energy) beacons or active tags are too big to be installed properly on most objects. There are few object tracking systems. Existing ones commonly involve computer vision Pirsiavash & Ramanan (2012) which poses an invasiveness problem in private housing facilities like smart homes. The passive counterpart of RFID technology can offer an alternative solution. Passive tags, which do not hold any inner power supply, are much smaller than active tags (1mm thick) and can be installed on most objects. In addition, these tags are very inexpensive and, therefore, an affordable solution to the context of smart homes.

Still, object tracking with RFID offers specific challenges and applications when compared to classical resident/robot tracking. For example, it can sometimes be difficult to put a tag on some objects, either because of their form, or their usage, or their lifetime. While we can minimize the effect of those challenges, there are some others that are harder to solve due to the nature of the technology and the materials involved. In fact, most tags do not resist high temperature and cannot be used in

Table 1: Comparison of radio frequency technologies for tracking

Technology	Range	Precision	Pros	Cons
Global positioning system	Earth	3-5 meters	Always available	Cannot be used inside
Wireless Fidelity	70 meters	Few centimetres	Already in most homes	Expensive tags
Passive RFID	30 meters	Few centimetres	Cheap tags	Expensive readers
Bluetooth 5.0	240 meters	Few centimetres	Low energy	Restricting pairing mechanism
ZigBee	100 meters	Few centimetres	Up to 65000 devices	Line-of-sight constraint

the oven as inner components can melt. Microwave oven can also be problematic as it destroys the tags. Perishable objects like food can obviously not receive tags. Moreover, adding tags to several objects can increase the interference in a small room. There is also a physical limit on the number of tags a reader can process. For those reasons, object tracking is more challenging than human/robot tracking with wearable.

Despite all the challenges mentioned, object tracking is very promising for a better granularity in activity recognition in smart home and therefore it seems to be the most logical and reliable choice for the future of that field. It has been shown in previous studies that knowing the position of all objects at all time enable accurate step by step activities of daily living (ADL) recognition Bouchard et al. (2013). Since this final expert system will be used within a smart home for cognitively impaired persons, tracking objects is a more reliable way than tracking human using wearable as those persons can easily forget to wear them. This is also why we chose to use passive RFID tags. By putting them on objects, they should always be available and provide accurate data. However, it is possible to track both humans and objects at the same time using the system described in this paper.

The contributions of this paper are three-folds:

- The datasets collected for this project in a realistic smart home infrastructure are provided to the scientific community to help the advance of this discipline. Labeled RFID datasets of ADLs are very rare in the community.
- A new indoor tracking system based on the relative positions using passive RFID is introduced. This system can track several objects in real-time. Very few approaches can do this task in the literature.
- Three filters that can stabilize RFID readings and positions are introduced for this system. The filters are simple, yet very efficient according to the experiments conducted in our smart home.

This paper contributes to expert systems in the fields of healthcare and information retrieval. It uses classical data mining techniques to locate objects in a smart home and custom filters to track their positions over time. Concretely, the systems proposed in this paper have two main purposes: (1) direct tracking of objects for distracted people and (2) main building block for a non-invasive activity recognition system. As stated before, a non-invasive activity recognition system is an essential intelligent system to have in a smart home designed for impaired or semi-autonomous persons. Some examples of use of this expert system, from a medical point of view, may include:

- Detect if the person did not took his medication by looking if the pill dispenser moved (Movement is not sufficient to assert the pill was really taken).
- Fall detection by embedding tag in clothing.
- Quality, variance and complexity of the alimentation.
- Change in day to day habits following a stressful situation.

Those are all tasks that are currently performed by a caregiver and their automation would reduce pressure on healthcare systems. The final proposed system which is implemented in a full scale apartment uses simple algorithms where every decision can be explained to a human examiner. In a medical point of view, the ability to explain a decision is often mandatory.

The remainder of this paper is as follow. The next section presents the most recent approaches in the literature about indoor tracking and assesses their advantages/disadvantages in the specific case of object tracking for an expert system piloting an assistive smart home. Then we describe how we build the indoor positioning system (IPS) and how we evaluated it. Follows our indoor tracking system (ITS) with the same steps of methodology and evaluation, with a discussion.

2. Related work

In this section, we present the main approaches to passive RFID positioning and tracking. The decision to ignore the literature on ultrasonic sensors, passive infrared and camera based tracking is purely motivated by the inadequacy of these technologies in our context as stated in the introduction. Nevertheless, this section still includes some approaches that cannot be applied to objects tracking simply for comprehension purpose.

Many algorithms found in the literature are based on the reference tags principle (or landmarks) first exploited by the LANDMARC system Ni et al. (2004). The basic idea, shown on Figure 1, is to exploit the Received Signal Strength Indication (RSSI) of nearby tags fixed at known positions to adjust the RSSI of the tracked tags. The method can be improved with

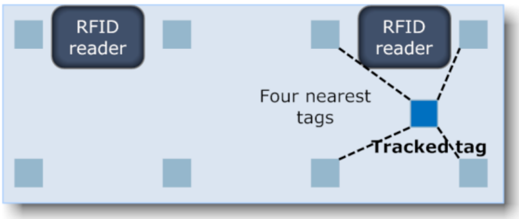

Figure 1: Reference tag method

various statistical filters Bekkali et al. (2007). They can also be used in combination with dead-reckoning, a method that infer the next position by combining speed and direction to the last known position. Kourogi et al. (2006). LANDMARC based systems work very well in general, with a good trade off between precision and accuracy. However, they require some type of landmarks on the floor which might not be very adapted to a smart home, which encompasses furniture and where most of the tracking is not performed at floor level de Sá et al. (2016). Moreover, this technique can hardly be adapted for the tracking of several objects.

The second family of algorithms is based on trilateration Fortin-Simard et al. (2012) and triangulation principles Liu et al. (2011). Trilateration uses mathematical propagation models of radio waves to convert RSSI to distance from antennas. Those distances then allow to compute isolines around antennas. The position of the tracked entity is the intersection point between three circles. Triangulation cannot be performed with all RFID systems as it requires the capability to calculate the angle of arrival. It is performed by using angles of arrival from at least three antennas in order for an intersection point to be found. The multipath propagation problem arises with those methods as the same signal can reach the reader from multiple directions with varying strengths. With this family, it is hard to track objects when they are in the range of less than three antennas. With only two of them, there is two intersections and contextual information are required to pick the right one. For instance, one intersection could be outside the room and therefore discarded. When there is only one antenna available, those systems become proximity based as there is only one signal to use. With strategically placed antennas proximity based methods can provide useful results and even full tracking system Kim et al. (2013). In addition, these techniques rarely work straightforwardly with passive RFID. They often produce several intersection points or none at all. Consequently, *ad hoc* method must be exploited to handle each of these special cases.

Finally, a last family exploits data mining and learning algorithms. However, the work on this family is scarce and mainly explores other wireless technologies. For example, Yim et al. Yim (2008) exploited wireless local area network access points to build a decision tree during an off-line phase in order to determine the user's location. They have shown that their technique is simpler to implement and perform better than the classical fingerprinting methods. Our hypothesis is that it should also perform well with RFID technology. One advantage of data mining algorithms over trilateration approaches is that they can track objects even when they are in the range of only one or two antennas as this is what the classifier learns for this position. Also, they generally requires less calculation in the on-line phase and thus, they are faster. Still, the set-up time can often be longer. One recent example of approach using learning methods is Calderoni et al. Calderoni et al. (2015). In their work, they developed a localization system based on active RFID to track the patients of an hospital. The patients wear a bracelet and the raw signals received by the antennas deployed through 48 rooms are directly used in a random forest classifier. Although their work is interesting, they used active RFID which is inadequate in our context and their experimental dataset is very small (14622 observations).

3. Static positioning inside a smart home

This paper describes a concrete implementation of an object tracking expert system within a real smart home. This system is able to track in real-time several standard objects that are used in daily life activities (cups, glasses, plates, remotes, etc.). The goal is to gather more spatial information to enable higher granularity in the activity recognition algorithms. The first component of the tracking system is a module able to perform static positioning. This module, called the Indoor Positioning System (IPS), is built to work very fast. Roughly, it takes RFID readings as inputs and it outputs a position using a classifier learned with a classic data mining algorithm, the random forest. This section first presents the smart home infrastructures used throughout this research project. Then, a special object, used to build the classifier in this experiment, is presented, followed by a formal description of the logical position concept exploited by our tracking system. The last subsections are about our methodology, from data collection to model evaluation and selection.

3.1. Smart home

This research took place at the DOMUS laboratory, at the Université de Sherbrooke. The DOMUS laboratory contains a realistic smart home infrastructure in which people could comfortably live. The smart home is composed of six rooms: a hall, a bedroom, a bathroom, a kitchen, a lounge and a lunch-room. In each of those rooms there are many sensors including passive infrared motion detectors, standard video cameras, temperature sensors, flow-meters, pressure plates and smart power analyzers. There are also twenty RFID antennas placed strategically to cover all the inner surface of the smart home. Figure 2 is a picture and Figure 3 is a map of the apartment. The DOMUS laboratory mainly focuses its researches on helping elders to stay at home. All those sensors are used as information input to build monitoring system to accompany the inhabitants in their daily lives. Thus, being able to track objects can enable better monitoring by adding additional information about the context of an activity.

The positions of the twenty RFID antennas are shown by the big crosses on Figure 3. The antennas are fixed on the wall at about one meter high. The antennas are MT-262013/TRH/A/K

Figure 2: A picture of the DOMUS smart home.

from Wireless Edge LTD and they operate at a frequency between 902-928 MHz. The only information they give is the signal strength (in Db) and the ID of the tag that responded. There are usually between two to six antennas in range at any position in the smart home. The twenty antennas are coupled with 5 readers that operate in round robin. It means that the signal of any antenna on the same reader cannot overlap, thus reducing the overall interference level in the smart home. Moreover, to reduce the overlap even more, the antennas are configured to work at a lower range than their full capabilities (around 3m). Placing all antennas in round robin on a single reader would have the major consequence of drastically increasing the delay between two readings and therefore reducing our real-time tracking speed. As the reader see farther in the paper, the exact location of the RFID antennas is not of critical importance. As a guideline, for the method to work best, they should be grouped by readers as much as possible and there should be more than one antennas capable to reach a tag at any position in the smart home.

3.2. Special object

As the first module toward the deployment of a real-time tracking system, the Indoor Positioning System (IPS) aims to find the position of objects within the smart home using the RSSI of RFID tags. The first step would then be to collect data in order to do the learning. To do so, it was decided to design a special object that would maximize the signals transmission. This special object has a shape similar to a plausible object findable in a house. Yet it is built to be optimal towards RFID to make sure we get the best IPS possible. The reader should note that the special object is not a requirement, and is mostly used for the learning and the testing of our tracking system. Given those constraints, an empty reusable plastic bottle of about thirty centimetres high was selected as a starting point. The bottle is in plastic so it does not interfere too much with

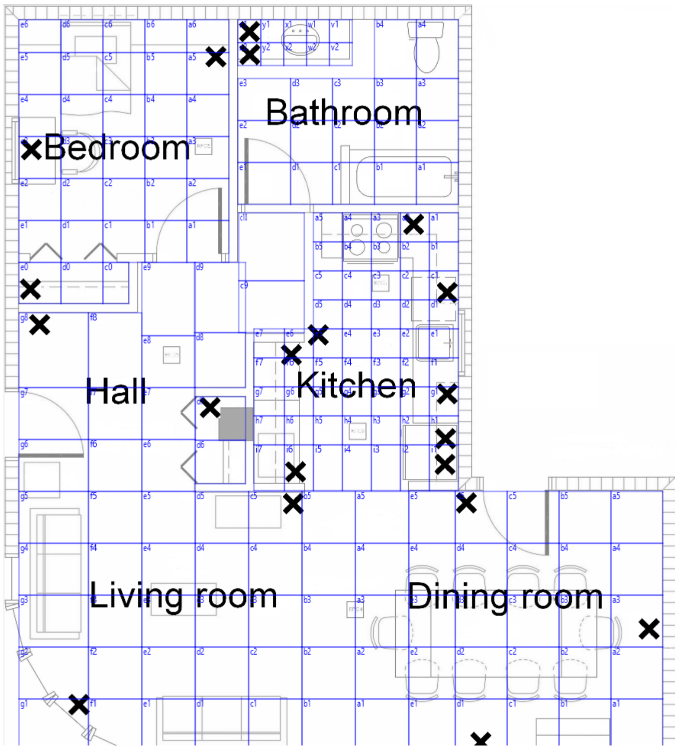

Figure 3: A map of the DOMUS smart home. The crosses mark the positions of the RFID antennas. The grid represents the zones.

the RFID signal. Four passive RFID tags are installed on it, each one facing a different direction. This way, a tag is almost always directly facing an antennas, resulting in the best signal possible under realistic conditions. Before putting the tags on the bottle, each of them was tested separately to make sure they all give similar signal strength. This is a required step to ensure that we discard bad tags that deviate too much from the mean, as suggested by Brusey Brusey et al. (2003). Programmatically, we merge the RSSIs of those tags into a single one by keeping only the highest values of the four. Figure 4 is a picture of the bottle.

3.3. Logical zones

The positioning task is generally considered as a regression problem. Therefore, most systems aim to produce precise coordinates from a given origin as a position. For instance, it is the case for the GPS system, which outputs precise latitude and longitude on Earth's surface. Still, it is not always mandatory to have the exact location of an entity. Often, it is enough to have a relative position, especially for qualitative applications. Indeed, if your goal is to tell that 2 objects are within a given range, relative positions are as good as exact positions. Moreover, in the case of an inaccurate technology, a stable qualitative information about location is more reliable and easier to take advantage off than a constantly moving precise location. With that thought in mind, in this paper, the task of positioning is considered as a classification problem where the goal is to classify RFID readings into qualitative zones. The zones are fictional divisions that can take any shapes or dimensions. Such zones are learned

Figure 4: A picture of the custom object.

through a machine learning technique. For simplicity, square zones of equal dimensions within a given room were mainly used in this project. In a previous version of the IPS, heterogeneously sized zones were tested to reflect the lower quality of the radio signals in some part of the smart home, but the research team could not confirm that such measure would significantly increase the accuracy of the IPS. There are no overlapping zones between rooms even though it would not make any difference, as rooms are not taken into account explicitly in the IPS.

For the logical zones, many dimensions were tried in various rooms to see how the accuracy of the classifiers would reflect on our choice. For instance, in the dining room, the zones were first set at 100cm x 100cm. Then, we reduced the dimensions to 75cm x 75cm, 60cm x 60cm, 45cm x 45cm, 30cm x 30cm and 20cm x 20cm. At this point, accuracy dropped by a large margin and we decided to stop. This gave us an idea of the precision the final system could achieve. Nevertheless, there seem to be a general consensus around smart home researchers that an accurate system is preferable over a precise system that fails more often. The students and researchers at the DOMUS laboratory expressed the same concern in that matter. Accordingly, bigger zones are selected for the tracking system. Thereby, the following dimensions are used throughout the smart home: 40cm x 40cm in the kitchen, 60cm x 60cm in the bathroom and the bedroom and 75cm x 75cm in the hall, the lounge and the lunchroom. There is an exception on the counter of the bathroom where zones are of 30cm x 30cm. Those dimensions reflects the kind of activities people usually do in those rooms in term of movement amplitude. Indeed, cooking requires smaller movement of objects than vacuuming for example, thus smaller zones are required to see those objects move. This is why the kitchen and the bathroom counter have an higher precision. Figure 3 presents a map of the smart home with the final zones drawn on it. In the dataset we provide, zones are named by a letter and a number (like a1). Their name could instead reflect their position, like sink1, to facilitate usage of an expert system based on logic rules on the computed positions.

3.4. Data collection

As we mentioned at the beginning of this section, the IPS uses a random forest to classify RFID readings to zones representing objects position. To construct the random forest classifier, a learning dataset was constructed. Since there are twenty RFID antennas throughout the smart home, a reading consists of a vector of twenty integers, one per antennas. The possible values range from -70 decibel to -35 decibel. For each zone, we collected fifty readings by placing the custom bottle in the middle of them, at about one meter height. This height is the one at which most people hold objects while moving them. The bottle is in the middle of each zone to get the most distinct readings possible. In no way it means that to position objects they have to be in the middle of a zone, it was simply done to maximize the readings difference from zone to zone during the learning phase. The special object is placed on a wooden stool to avoid interference from metallic surfaces or human body. Also, the antennas were set to emit a signal with a 750ms interval to avoid interference between them. Using the final dimensions, this means a total of 9550 readings in 191 zones. They are all available on the DOMUS website, `https://www.usherbrooke.ca/domus/`, along with the 25 000 more readings we collected during the precision test described in the previous section in order for researchers to reproduce our results or work toward improving the indoor positioning of objects with passive RFID.

4. Building models for the IPS

The next step to build the Indoor Positioning System (IPS) is the learning phase itself. The positioning is considered here as a classification problem. To do so, thorough testing with the most popular classification algorithms were performed. Our goal here was not to make a strong theoretical contribution, so building a specific classification algorithm was ruled out. Moreover, it turned out, as expected, that most models perform quite well on this type of classification problem. Even though the bulk of our tests was done on decision trees, other families of algorithms, such as the multilayer perceptron and the Bayesian network, were also explored. The well-known Weka data mining library Hall et al. (2009) was exploited as well as some of its extensions in order to achieve this learning phase. The classification problem consists of finding to what class a vector of twenty values given by the twenty RFID reader belongs. The classes are the different zones, expressed in the generic format *room_number*, like *lounge_a1*. This generic format then allows to convert zone names into Cartesian coordinates if needed. Our problem is, therefore, a low dimension combined to a high number of classes. There are actually a countable finite number of possible different vector of value in this specific problem. We argue, however, that an exact model cannot be learned since two identical vector instances could belong to different classes given the precision of our RFID readers. Moreover, obtaining all different instance of vector could take a significant effort, if even possible.

Table 2: Accuracy and distances with non tree models.

Non Tree	Error dist	Mean dist	Wrong zone	Acc(%)
NBayes	3.1058	0.4885	1480	83.2738
NNET	2.7501	0.2128	728	90.5337
1NN	2.5408	0.1268	473	94.7573
BayesNet	**2.0543**	**0.0684**	**316**	**96.6064**

Four metrics were used to evaluate the performance of each classifier. The first one, *Error dist*, is the mean distance of the wrongly classified readings from the target zone only. The second one, *Mean dist* is the global error margin on the dataset. Distances are given in zones and not in centimetres as zones do not need to have all the same dimensions. The third one, *Wrong zone* is the total number of misclassified zone. It is in direct relation with the last one as we can express the *Accuracy (Acc)* as $1 - WrongZone/NbTotal$. The remainder of the section describe the algorithms tested to build the IPS and present a comparison of their performance.

4.1. Multilayer perceptron

The multilayer perceptron Arora (2012), also known as a neural network, is a powerful model that is state of the art for many machine learning tasks. In Weka's implementation, all nodes are sigmoid units and learning is done through back-propagation. The loss function is a simple boolean. It has a single hidden layer of size given by: $(NbAttributes{+}NbClasses)/2$. The learning rate is 0.3 and the momentum is 0.2. The network is trained for 500 epochs, unless a validation set is provided. If so, early stopping is done after 20 epochs with decreasing performance. Because of all those hyper-parameters, training a neural network can often be a long and complex process. Neural networks are good at generalizing, especially from high dimensional datasets, but this is far from being the case in our context. Finding the bests hyper-parameter is also a tedious process that can take a long time depending on the computing power available. The results are shown in Table 2 under the label NNET.

4.2. K-Nearest-Neighbours

The k-nearest neighbours Aha et al. (1991), called Ibk in Weka is a simple algorithm that consists, for each instance to classify, to look for the k-closest ones in the training set. The Euclidean distance is used to measure the similarity. The default value for **K** is 1. We also tried with **K** varying from 1 to 5. Figure 5 shows the results of those tests. As we can see, accuracy decreases as **K** increases. The two last tests in Figure 5 demonstrate that not all fifty readings from the same zone are the same and that some other zones have very similar readings. If all readings were the same for a zone and never found in any other zone, results would have been similar to the first ones. However, it requires the entire dataset to be contained in memory, which might cause problems on some architecture with limited memory.

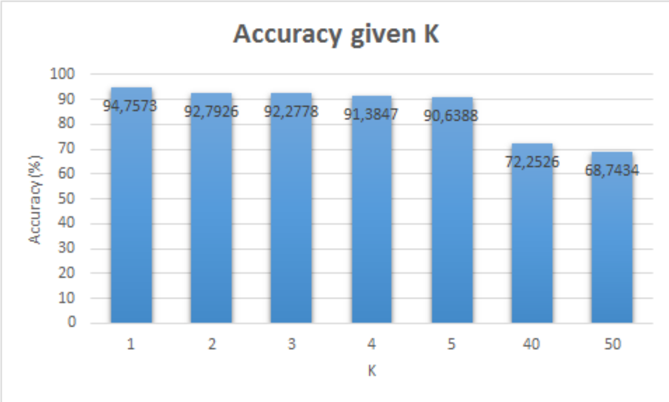

Figure 5: Accuracy given **K** in K-Nearest Neighbours

4.3. Bayesian Network

The Bayesian network Bouckaert (2008) is a probabilistic model presenting itself as a directed acyclic graph that we can use to represent a probability distribution of classes over attributes. Training a Bayesian network can be seen as two separate steps: learning the network structure and learning the probability tables. The one present in Weka offers numerous possibilities in the choice of algorithm for each of those steps. We used the default K2 algorithm to learn the network structure. K2 is a hill climbing method that uses a fixed ordering of variables to maximize quality measure of the network structure. In our case, the quality measure was the Bayesian metric from Cooper & al. Cooper & Herskovits (1992) (see equation (1)), a measure that tends to approximate the likelihood of the graph. The graph was initialized as a Naive Bayes Network. This means that the classifier node is connected to all other nodes. In equation 1, B_S represents the network structure of the database D. $P(B_S)$ is then the prior network structure and r_i is the cardinality of the data. N_{ijk} is the number of cases in D where the variable x_i as the value v_{ik}.

$$Q_{K2}(B_S, D) = P(B_S) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(r_i - 1 + N_{ij})!} \prod_{k=1}^{r_i} N_{ijk}! \quad (1)$$

To learn the probability tables, the SimpleEstimator is used. It estimates the conditional probabilities directly using the given data. A smoothing constant of 0.5 is used by default when computing the probability tables. This model proved to be one of the best tested in our experiments, only matched by the Random Forest. However, it requires a very long training. Indeed, the complexity of complete inference is NP Wu & Butz (2005). Classification could also be longer as well because it requires a lot more computation than decision trees which are usually under $O(\log n)$.

4.4. Naive Bayes

The Naïve Bayes (NBayes) model does a simple data analysis to later perform classification by a probability calculus over all the attributes. If C is the class and x a data vector from the set X, then the probability of a class c given the example

x is $P(C = c|X = x) = \prod_i p(X_i = x_i|C = c)$. If needed, the data is discretized. This model makes the assumption that the attributes are independent from the class. Once the probability for all classes is computed, the max is selected as the class. This model did not perform well for our IPS.

4.5. Classic Trees

The next category of algorithms tested are the decision trees. Our hypothesis was that these algorithms would be the most adapted to the properties of our dataset. Therefore, the most popular algorithms were all tested. The the results are summarized in Table 3. The same metrics were used for the trees based models than for the previously tested learning methods.

First, Weka offers two implementations of the decision tree, one using the Cart algorithm Breiman et al. (1984) and one using the C4.5 algorithm Quinlan (2014). C4.5 is called J48 in Weka. It uses the information gain metric to find the best attribute to split on in each node until there is only one class left or a minimum number of examples is reached. Then a pruning step is done trying to remove leaves that do not bring any accuracy gain. There is also a slightly different version of C4.5, called J48Grafted Webb (1999). It essentially is J48 with a post training algorithm that adds nodes to an existing tree to reduce training error. However, this grafting step was not very effective on our dataset, yielding in an increase in accuracy of less than 0.1 percent. Cart is called SimpleCart in Weka. It uses the Gini impurity as the splitting criterion. It can be expressed as in equation (2) where i and k are classes from the item set in the node on which I_G is computed and f represents the fraction of the item in that set that have the indexed class.

$$I_G(f) = \sum_{i \neq k} (f_i f_k) \tag{2}$$

It also differs from C4.5 at the pruning step where it uses the cost complexity pruning. It performs worst by 1 percent than C4.5, which suggests that pruning does not yield significant differences with our dataset.

Another tree that we tried is the Random Tree (RTree) Breiman (2001). In it, only a subset of attributes is considered for each split. It is therefore well adapted to datasets consisting of a large number of attributes. The size of this subset is $\log_2(D)+1$, where D is the number of attributes. There is no pruning. It performed slightly worse than the other trees. Indeed, our datasets are composed of only 20 numerical attributes and Random Tree is better suited for very high dimensionality problems.

4.6. Trees mixed with other models

The trees algorithm presented in section 4.5 all worked under the same basic principles. This sub-section presents three other trees that have a model on their leaves or nodes.

4.6.1. Functional Tree

Functional Tree Gama (2004), FT in Table 3, is like a normal tree with the difference that it has logistic regression functions at inner nodes or leaves or both, as in our experiment. Weka uses fifteen iterations of LogitBoost Friedman et al. (2000), a variant of AdaBoost where logistic regression techniques are applied on it. If AdaBoost is a generalized additive model, LogitBoost is a convex optimization of its logistic loss. The Functional Tree was not very effective on our data, with a lost in accuracy over C4.5.

Table 3: Accuracy and distances with the trees.

Tree	Error dist	Mean dist	Wrong zone	Acc(%)
LADTree	2.5308	0.2683	1001	88.5690
Reptree	2.6336	0.2571	916	89.0523
FT	2.6242	0.1870	674	91.2797
LMT	3.7239	0.2711	673	91.5213
RTree	2.9782	0.2264	718	92.0782
Cart	2.3776	0.1645	652	92.4564
J48	**2.2521**	0.1376	577	93.4860
J48graft	**2.2254**	0.1322	561	93.5701
NBTree	2.7027	0.1486	520	93.9588
RForest	2.4331	**0.0622**	**243**	**97.2158**

4.6.2. NBTree

NBTree Kohavi (1996) particularity is that its leaves are composed of naive Bayes classifiers. The author of this model affirms that it can outperform both the normal tree and the naive Bayes classifier, especially on large datasets. With a gain of 0.5 percent over C4.5 and of 10 percent over the naive Bayes, this affirmation seemed to be confirmed.

4.6.3. Logistic Model Tree

The Logistic Model TreeLandwehr et al. (2005), LMT in Table 3, looks like the FT in the way that there is logistic regression functions at the leaves. It produces bigger trees than FT, thus being longer to train for similar accuracy. Indeed, a bigger tree implies more leaves and more logistic functions to learn. It also means that prediction time is slower since there are more nodes to examine.

4.6.4. Alternating tree

The last single tree we tried was the LADTree Holmes et al. (2002), an alternating tree capable of multi class classification. Like the Functional Tree, LADTree uses LogitBoost as the base of the algorithm. The tree presents itself as a series of AND/OR rules, one rule per boosting iteration. A score vector is associated to each answer to those rules (true or false), one score for each possible class. Classification is made by adding all confronted scores at each node reached. Then the class associated with the highest score is chosen.

By default, Weka uses only ten boosting iterations, but it resulted in an accuracy of only 31.8554% with our dataset. In their original paper, the authors show that accuracy gets significantly better with more iterations. Table 3 shows the results with 50 boosting iteration. With 50 iterations it gives an accuracy of 88,5690%, a number more consistent with the results from other trees. The downside of this algorithm is that adding more iterations also greatly increases the training and the prediction time. While it took only minutes to train and test with the 10-fold cross-validation and ten boosting iterations, it took several hours with fifty.

4.7. Trees forest

The last tree algorithm that was tested is the Random Forest (RForest) Breiman (2001). Random Forest is simply 100 Random Trees trained separately, exactly the same way as described before. The predicted class is then the modal class between all 100 predicted classes. This forest surpassed the single random tree, thus being by far the most accurate model on our datasets. It even outperformed the Bayesian Network while being shorter to train and faster to use. Its mean distance is more than half the nearest tree which is means even when a tag is positioned wrongly it has more chance to be close enough for the activity recognition algorithm. Nevertheless, it is slower than basic classifiers such as C4.5 and SimpleCART, which was to be expected since it is like searching in a C4.5 tree 100 times in a row.

4.8. Discussion

In the previous sections, the algorithms tested to build the IPS were presented. Most of them produce a similarly good accuracy and a low mean distance from target. Still, some of them are considerably faster to train and to use. Training time is not really a big factor since it is part of the off-line configuration phase. Prediction time, however, is important to consider for the final goal of real-time activity recognition. This final system will have to locate many objects multiple times per second for the expert system to take quick decision and act upon it. For this reason, we chose the use the random forest as the classifier. Classification in a random tree is really fast as it requires few numerical comparisons. The maximal depth of a tree is 17 nodes in our final random forest for a size of about 2900 nodes and leaves. The model built can be used straightforwardly to classify any objects equipped with passive RFID tags into one of the logical zones. The RSSI values of that object simply have to be passed as a parameter to the model into a single 20 attributes vector. If the object is equipped with more than one tags, the signals are merged using the method described in the previous section. For the final IPS, we chose to use a random forest of 50 trees (97.2999% accuracy). By using only 50 trees, the classifier is faster for an accuracy similar to the 100 trees. Both training and classification are faster, which is the principal criterion for this part, given similar accuracy.

5. Dynamic tracking system

The next part of this paper focuses on transforming the accurate indoor positioning system into a dynamic system able to track several objects in real-time. It first present the software architecture then the experiment we did with it.

The indoor tracking system (ITS) is built around the IPS. It handles all readings from the collection to the final zone and gives it to the sequence analysis module. The complete flow is shown in Figure 6. First, there is a module responsible to acquire readings from the readers and aggregate them by object, if necessary, as explained previously. From this point, a data object consists of the reading vector, the object's name and a timestamp. The object's name is retrieved from a database using the tags ids. The ITS uses this name to keep track of objects over time as each inner module works asynchronously.

At the beginning, the ITS only served to record the objects' positions over time. However, first results quickly showed the need for some filter. For this reason, we added a first filter (prefilter) on the raw readings to reduce the impact of a misreading. Then, a second filter (post-filter) is applied to the zone found by the IPS the ensure regularity over time. As those filter need context to operate, they are included in the ITS. This way, they have access to the full sequence of positions, or the required window.

The ITS is currently implemented in Java. Each module works in his own thread and they exchange data using a custom message queue with the publish/subscribe paradigm. Execution speed is an important factor to maintain the real-time aspect of the tracking. Accordingly, each module is optimized to do its task under 20 milliseconds, which is the speed of the RFID readers. In fact, on average, they operate for about 4 milliseconds per object and sleep until a new reading awakes them.

The sequence analysis module is not properly part of the tracking system. Its only purpose is to extract statistics on the system. Since the ITS is always running, it is in the sequence analysis module that we can record the paths taken by the objects as determined by the ITS and compare them to a theoretical path.

In Figure 6, there is no mention of any user. This is because the whole system is always working, always analyzing the position of every tag. If an object starts moving, it should be because someone moved it, but it still does not make this person a user, only a source a variation in the system, much like the sun is not a user from the perspective of a thermometer.

5.1. Filters

As stated before, the first results indicated a need to filter the data received from the RFID readers to make the predicted zones and sequence more compliant with the reality. Indeed, even if the accuracy of the random forest is around 97% at 750 milliseconds interval, there are still some interference that occur occasionally. When the antennas emit at 20 milliseconds, those interference happen more often, leading to more positioning errors.

5.1.1. Filter on RFID readings

The first filter, noted pre-filter in Figure 6, exists to regulate the raw data. It is needed because sometimes a reader might give a value of 0 Db when it should not because of interference.

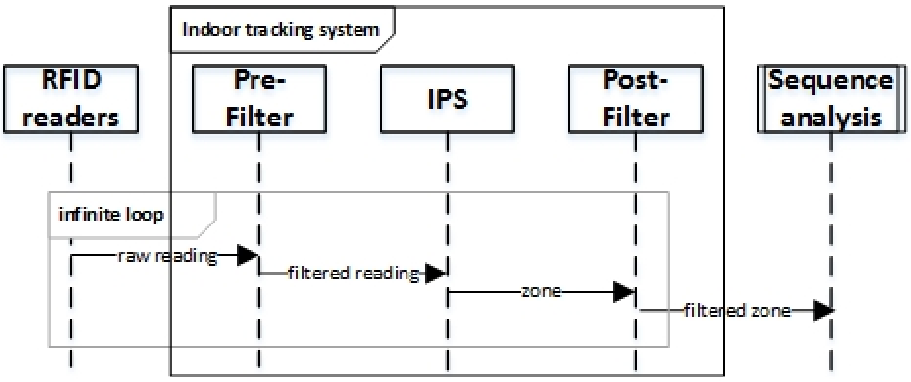

Figure 6: Data flow inside the ITS

This wrong value could then conduct the classifier to predict a wrong zone. This kind of error made objects move by themselves when they were not supposed to be moving at all. Instead of just ignoring the whole reading, we choose to compensate for the missing values by using a moving average on each reader. It is expressed by the following general expression:

$$\bar{R} = \frac{\sum_{n=0}^{N-1} C_n * R_{N-n}}{\sum_{n=0}^{N-1} C_n} \tag{3}$$

Where R is the weighted average, R_{N-n} is the Nth reading and C_n is the Cth weight coefficient. Note that we consider N_0 as being the current reading. We also have to divide by the sum of all coefficients to make sure they sum to one. We tried four different weighting function. The weights are given by the following functions: the logarithm function, the linear function, the exponential function with base 2, and finally, the factorial function. We also tried the simple moving average, or the constant function. They are explained in Table 4. The algorithm for the filter is simple:

1. Receive a vector from the previous module;
2. Add the vector to a circular buffer of size n;
3. Apply equation 3;
4. Send the result to the next module (the IPS).

Table 4: Weighting function for the pre-filter.

Distribution	Expression	Remarks
Logarithm	$C_n = \frac{1}{\ln(n+2)}$	We use $n + 2$ to avoid the part of the logarithm function that is negative or rapidly growing.
Linear	$C_n = 1 - 0.1n$	
Exponential	$C_n = \frac{1}{2^{a\,n}}$	
Factorial	$C_n = \frac{1}{a\,n!}$	We only use $N = 10$ and tweak the modifier a as $n!$ grows too fast.
Constant	$\bar{R} = \frac{\sum_{n=0}^{N-1} R_{N-n}}{N}$	

5.1.2. Filters on predicted zones

The second filter, noted post-filter, applies directly on the predicted zones. While the first filter can reduce the impact of some misreading, there can be long interference in the signal that extends the moving average window. There is also some situations where the objects seem to teleport themselves, also because of interference. This second filter directly tackle this issue. It comes in to version: a blocking filter that only allow a predicted zone to be a direct neighbour from the previous one and a limiting filter that only allow moving one zone at the time.

The blocking filter is the first one we tried with this approach. Let us say the object is thought to be in zone A2 and the next predicted zone is D5. Of course, one of the two positions is false as this would mean the object teleported, assuming it moved at normal speed. In that case, the blocking filter would reject D5 and say the object is still in A2. Illustration of this example is shown in Figure 8. Obviously, this mean that if the filter misses the first move for some reason, the object would get stuck. The resolve this issue, we added a time-out on the position where after X rejected zones the current position becomes the predicted one, no matter what the filter says. In our experiment, we used a $X = 20$. This represents about half a second which is the time needed to cross most zones at average speed. The algorithm for this filter is:

1. Receive a zone from the previous module;
2. Retrieve the previous zone;
3. If the received zone neighbours the previous zone, return the zone;
4. Else if the time-out is expired, return the zone and reset the time-out
5. Else return the previous zone and increase the time-out by one.

The limiting filter is a more permissive version of the first one where moving is always allowed, but one zone at the time. This means that with the previous situation where the object is in A2 and the newly predicted zone is D5 the filter would output the zone B3, as illustrated in Figure 8. This filter has eight degrees of liberty that allow moving in each direction of our two dimensions grid. Since moving is always allowed, there is no need for a time-out. This means that every outlier will produce a move. In other words, this filter does not remove any outlier, it only reduces their impact. On the other hand, it increases the speed at which the system can recover after missing several zones when compared to the blocking filter. Moreover, the fake zone visitation it creates have a high probability of not being fake at all since objects tend to move in straight ways, either when rolling, falling or when held by humans. Figure 7 shows this filter in action. The algorithm is:

1. Receive a zone from the previous module;
2. Retrieve the previous zone;
3. Compute a combined zone:
   (a) Move the x axis by one unit in the direction of the received zone;
   (b) Move the y axis by one unit in the direction of the received zone.
4. Return the combined zone.

6. Experiments

In this section, we present the experiments we conducted using the ITS described above. The results were all collected by the sequence analysis module.

6.1. Tracking data

As with the IPS, the first step for the indoor tracking system, ITS, was to collect meaningful data we can compute the accuracy on. Since the final use of the ITS will be as a key component of an activity recognition expert system, meaningful data is when you mimic a real life activity. So, for each room, we designed four paths representing activities like taking something from the buffet to the kitchen counter or entering the apartment, going to the closet before exiting. We draw those 24 paths on a map of the zones and listed crossed zones. This forms the theoretical dataset. The practical dataset was collected by making the same plastic bottle of water we used for the IPS follow the various paths. To do so, we placed black tape on the floor and we walked the paths holding the bottle in our hands. At all time, we made sure the bottle was right over the line. To add some realism, we varied the position of the bottle by sometime holding it upside down or on the side. We repeated each path ten times to get more data. For this experiment, RFID antennas were set to emit at 20ms intervals to better catch the movement. However, RFID is not a real time technology. This means the system will do its best to emit at 20ms interval but it may take longer. There is no guarantee on the real time between two RFID signals. The practical dataset then consists of several thousand RFID readings for each of the 24 paths. Figure 9 shows our experimental set-up, with the tape on the floor and some of us holding the bottle. This dataset is also freely available on the DOMUS website. It is important to understand that these path where used in order to accurately evaluate the system. The ITS is in no way limited to tracking these examples.

6.2. Evaluation metrics

Once we had collected the dataset, we wanted to evaluate how the ITS would perform on it. Literature on how to evaluate a tracking system is scarce. Most of the time, it is done manually, or by comparing to an highly accurate tracking system such as ActiveBatHarter et al. (2002). A system like that is expensive to buy, and in our case, since we are tracking objects and not human, there is no guaranty it would have worked properly. In the end, we used two metrics called Targeted Zones Found (TZF) and Sequential Targeted Zones Found (STZF) to evaluate it.

Let us say that T is the set of all theoretical zones for a given path and that P is the set of the practical zones found for this path. Then, TZF corresponds to $T \cap P$, or, to put in simple words, TZF is the proportion of the theoretical zones the classifier correctly identified. Logically, if the classifier was perfect this score should always be perfect since this is essentially the metric it was trained on.

The STZF metric is an extension on the previous one that takes into account the temporal factor of a tracking sequence. For this metric, a predicted zone will only be counted if the previous theoretical zone was previously identified, as in a finite state machine. For example, imagine there is a theoretical sequence formed by the zone B-A-C, in that strict order. Imagine also a practical sequence of A-A-A-C-A. In that case, the TZF would be 2/3 and the STZF 0/3 since B was never predicted. If the practical sequence is A-A-A-C-B-A, then TZF is 3/3 and STZF 2/3.

Another metric commonly used while comparing sequence is the edit distance, also known as the Levenshtein distance, which reflect the number of operation needed to convert one sequence into another. This metric is not really significant in that case since the practical sequence is much longer than the theoretical one giving a poor score due to the number of *delete* operations to perform.

6.3. Results

The previous section presented the two metrics we use in this section to characterize the ITS. We first presents the raw results and discuss on them. Then, we explain the filters we added in order to increase the accuracy of the system.

6.3.1. Raw results

In the beginning of this section, we presented the tracking dataset we collected, then the metric to evaluate it. Figure 10 and Figure 11 show the raw results of the ITS, given per room. Given the high accuracy of the IPS, those results are poor and clearly indicates that the human body greatly interferes with the RFID signal. With an average of 36% accuracy on the STZF metric, the system could not provide much useful information about objects' ongoing movement, only the position before and after.

6.3.2. Results with filters

The first filter we added to the ITS is the moving average discussed before. Results for all weighting function are shown in Table 5.

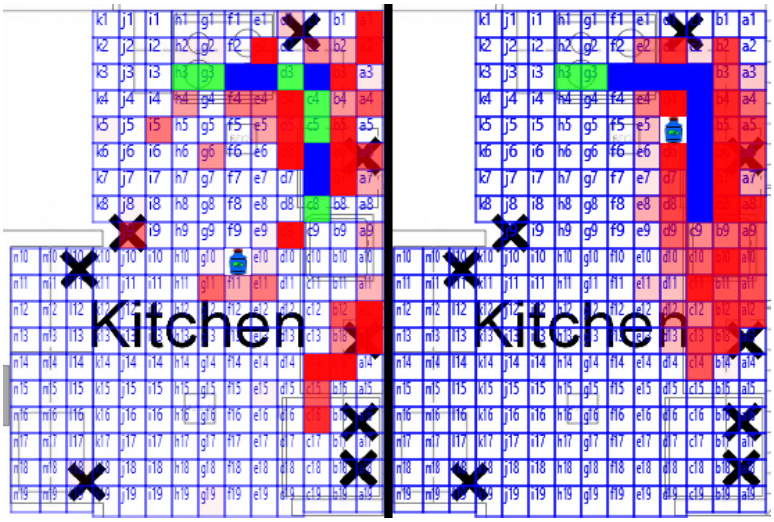

Figure 7: Map of the kitchen with raw data versus the limiting filter. The green squares show the expected path and they turn blue when correctly identified by the algorithm. Red squares are wrongly identified zones.

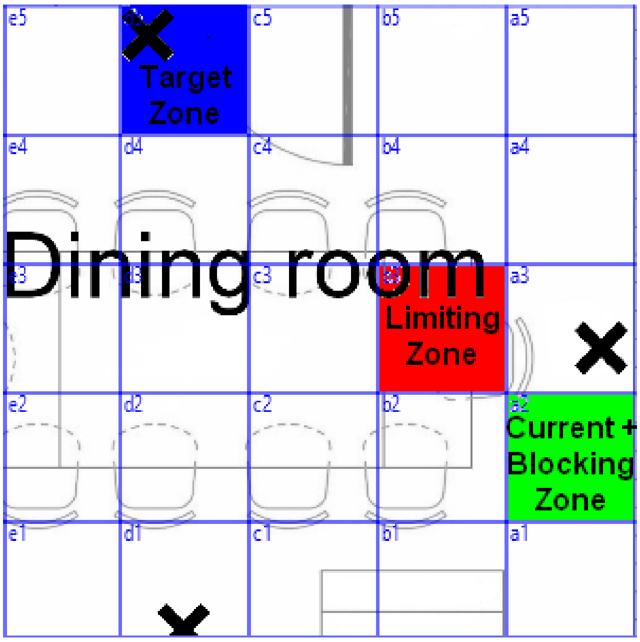

Figure 8: Example of the filters. In green is the current zone and the one predicted by the blocking filter. In red is the zone predicted by the limiting filter while in blue is the zone predicted by the random forest.

Figure 9: Experimental set-up

It clearly shows that the accuracy increase with N in the logarithm coefficient, while still staying under the accuracy of the raw data and therefore, also under the accuracy of the linear moving average. For the exponential coefficient, we tried many value for the modifier a. With $a = 1$ the results are terrible. However, with $a = 1/8$ or $a = ^1/_{16}$ the results are much better. The constant coefficient is interesting in that it gets its best results with a low N, as opposed to all other function. This suggest that at our walking speed the filters should give less importance to older readings.

Here are some results with some parameters on the kitchen paths. Note that accuracy increase with N, and the more smooth the function the better.

Result for the post-filter are shown in Figure 10 and in Figure 11.

6.3.3. Combining filters

So far, we presented two families of filters we can use to improve the accuracy over raw RFID readings. The first family alters the raw readings to smooth them with a moving av-

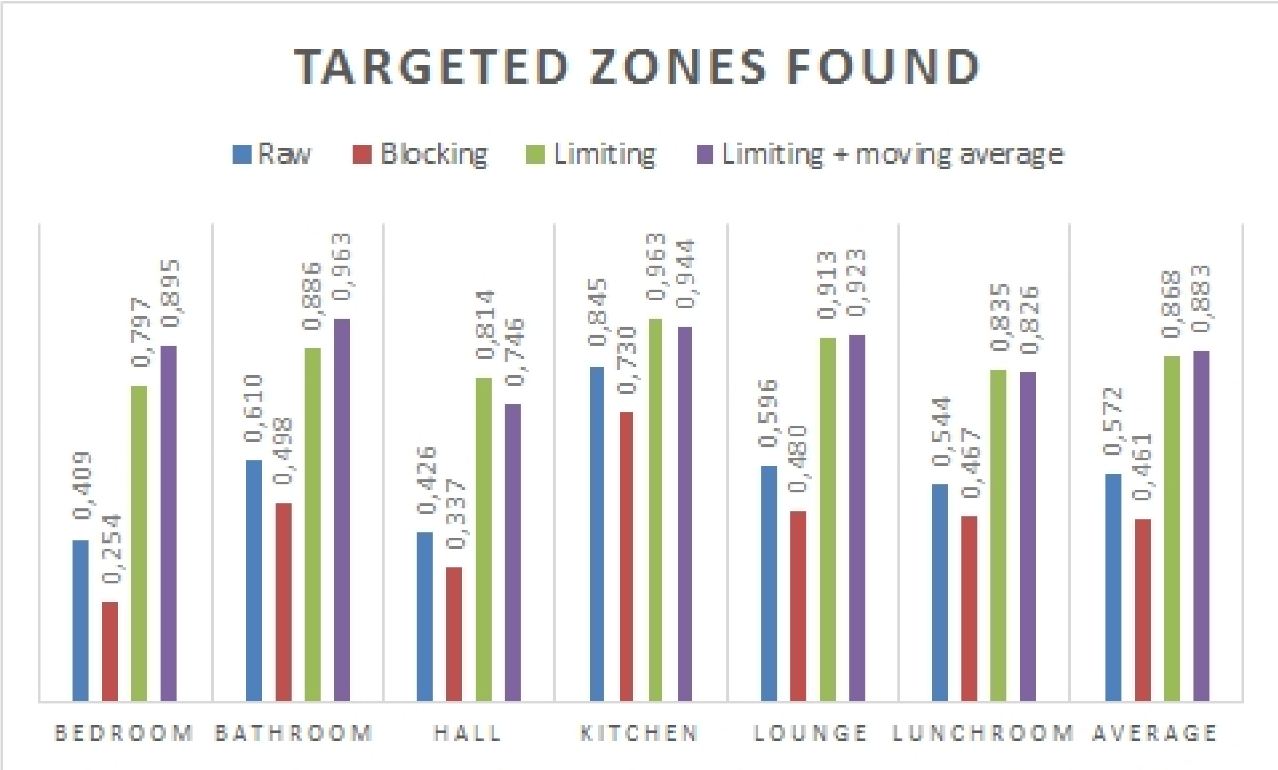

Figure 10: Results for the targeted zones found metric

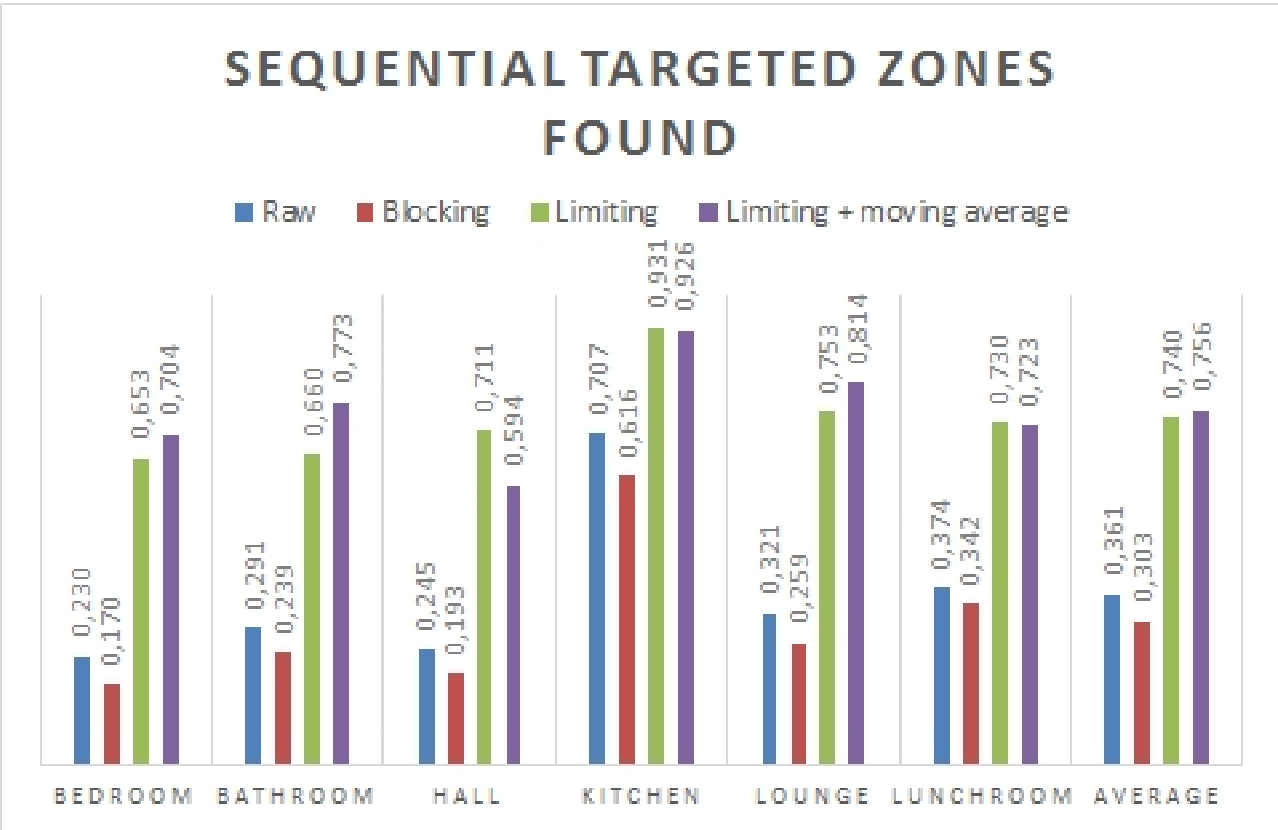

Figure 11: Results for the sequential targeted zones found metric

Table 5: Effect of N and the weight function with the moving average in the kitchen

Parameters		Metric	
N	Weight function	TZF	STZF
5	Constant	**0.814**	**0.656**
10	Constant	0.772	0.652
20	Constant	0.745	0.575
30	Constant	0.687	0.518
40	Constant	0.673	0.496
5	Log	0.367	0.288
10	Log	0.489	0.374
20	Log	**0.593**	**0.430**
30	Log	0.565	0.385
40	Log	0.552	0.400
5	Linear	0.417	0.315
10	Linear	0.533	0.393
20	Linear	0.763	**0.648**
30	Linear	0.783	0.600
40	Linear	**0.789**	0.565
5	Exp a=1/8	0.426	0.329
10	Exp a=1/8	0.666	0.498
20	Exp a=1/8	0.685	0.493
10	Exp a=1/16	0.661	0.499
20	Exp a=1/16	**0.733**	**0.580**
10	Fact a=1/4	0.623	0.455
10	Fact a=1/8	0.821	0.677
10	Fact a=1/16	**0.886**	**0.776**
10	Fact a=1/32	0.875	0.756
10	Fact a=1/128	0.878	0.720

Table 6: Limiting filter with the moving average

Metric	TZF	STZF
Bedroom	0.820	0.650
Bathroom	0.946	0.827
Hall	0.762	0.514
Kitchen	0.970	0.937
Living room	0.926	0.754
Dining room	0.843	0.734
Average	*0.878*	*0.736*

erage while the other compares a newly predicted zone with the current one to compute what should be the right zone. As they works on different sides of the same problem, they can be combined without any interference between them. We tried all possible combination and some results are shown in Table 6. Results show that the best combination is the limiting filter with the moving average using the factorial distribution, as shown in Figure 10 and in Figure 11. They were both the best filters of their family and they work well together, bringing an overall increase of 27% over the raw accuracy.

6.4. Execution time

In the previous sections, we mentioned several times that our system can do real-time localization. Given the nature of the algorithms in use, it is not possible to do hard real-time, that is when we can count the exact number of operations performed. However, it is possible to put a limit on the required time for each modules. In our implementation, this limit is still relative as we use classic Java and let the garbage collector runs when it seems fit. Empirically, we measured some statistical indicators on our datasets by running them in the ITS. Results are presented in Table 7. The longest the ITS ever take on our data is about 13ms. Considering that data collection happens every 20ms in an asynchronous fashion, even the longest data is still classified fast enough, with a 7ms in reserve. In average, the ITS takes about half the available time. The pre-filter and post-filter modules are faster thant the IPS, at about 2ms for the best configuration presented above. As they also execute in their own thread, they use 10% of their budget time. The IPS is the slowest module and therefore is the one that might need optimization before being used to track every objects of a smart home. A C++ implementation instead of a Java implementation could help if the need arises.

7. Discussion

In the previous section, we presented the results of our algorithm on a moving object. Results show that the accuracy of the system drops by a large margin. However, filters can be used to diminish the impact of interference. We applied a first

Table 7: Execution time of the ITS

Room	Max time	Min time	Mean time
Bedroom	12.3207ms	0.0429ms	0.0753ms
Bathroom	6.6098ms	0.0429ms	0.0734ms
Hall	7.8145ms	0.0308ms	0.0465ms
Kitchen	11.1712ms	0.0604ms	0.0945ms
Living room	12.9440ms	0.0489ms	0.0839ms
Dining room	13.1949ms	0.0305ms	0.0513ms

filter directly on the RFID readings to try to smooth variations between them. Physics tell us than the power of a wave diminishes as it moves away from its source. This variation is always monotone. Accordingly, no RFID reading should ever change drastically if the tag moves normally. Knowing this, it is normal to consider bad readings from interference as outliers and to try to smooth them using a moving average. In a future work, it could be interesting to try to infer correct readings to replace those outliers, like with a Hidden Markov Model.

The results also shows a greater accuracy in the kitchen. This room has the highest concentration of RFID reader of the smart home. Accordingly this is the room where the signal is the stronger and where the interference is at its lowest.

From the experiments we conducted, it appears that the choice of the classifier does not have a big impact on the positioning accuracy. Many classifiers present an accuracy over 90%. Trees, especially, all produce similar results. This implies that rule-based methods in general can solve the indoor positioning problem when presented as a classification problem. This is an advantage over probabilistic methods as they offer a better traceability of the decision. Another implication based on our experimental outcomes is also the main drawback of a tree classifier: what happens if an antenna gets a bad reading? A bad reading, no matter the cause, can prevent a rule to be met and may then result in apparent teleportation. We could mitigate this problem by building a tree where the physical proximity of the leaves is taken into account.

Our systems offer many strengths. They are summarized in the five following points:

- real-time localization thanks to the low complexity of the classifier;
- high accuracy which makes it possible to find weak spots and act accordingly (add an antenna, move metallic structure to reduce interference, etc.);
- multiple objects concurrent localization. The limit will come from interference at the readers, not our algorithms;
- scales to many more zones and many more antennas. The given model can take much more than 20 antennas, especially if their range have minimal overlap;
- arbitrary chosen precision. The IPS can mix precision of many meters in a warehouse and few centimetres in the adjacent office in the same system, for instance.

7.1. Effect of speed

We said earlier that our IPS was trained with the antennas set at 750ms between readings and that we reduced it to 20ms for this experiment. Such a change could have greatly affected the accuracy of our IPS. To make sure it did not, we conducted a simple experiment. We designed a special path over the kitchen counter and we walked it with the water bottle two times, one at normal speed with antennas set at 20ms and one at very slow speed with antennas set at 750ms. Our simple experiment indicated that the accuracy was better at 20ms when no filters were used. However, this is certainly because we were still moving too fast when at 750ms between readings, reducing the chance for our random forest to predict the right zone at least one time. Still, the difference in accuracy was small. Further experiments should be conducted on this subject.

8. Conclusion

In this paper, we presented an indoor tracking system for objects. This ITS addresses the problem as a classification problem and uses a random forest to predict relative positions. The random forest as an average accuracy of 97% on static objects. With the help of some filter, the whole system has a tracking accuracy of 75% on the sequential targeted zones found metric with moving objects. One big advantage of this system is that it does not require any calibration and it uses fast algorithms, making it suitable for real-time tracking. It is based on a low cost and non intrusive technologies, passive RFID. It is designed to enable fine-grained activity recognition. Despite those strengths, this ITS still has some weaknesses. The two biggest are the training phase of the ITS and the effect of human interference on the readings. Effectively, it takes significant time to collect the data needed to train the random forest. Future work on this subject could be to develop an automated solution, like a robot, to record all needed readings. One way to overcome the human interference problem could be to increase the strength of the signal, which was not possible with our antennas. It would also help to add more antennas in a way that the object is always directly facing an antenna despite being hold by an human. Some new filters might also help on this subject.

The research presented in this paper opens several new research directions. A first one would be to find a way to reduce the offline learning time. A way to do this could be to look how many training example are really needed. A second direction might be to introduce reference tag at fixed position (in a cupboard door, for instance) to propose an hybrid method between our system and the LANDMARK system. As indicated before, the filter we designed are very simple. More filters could be created to increase accuracy. There are many general filters for time series that we have not tried, like the Kalmann filter. In this work, we only used the raw readings from the RFID antennas. Future work could explore advanced features extraction from the datasets we provide. Feature considering the readings as a time series might also improve the tracking by naturally smoothing bad readings. As mentioned before, future research on trees could try to regroup leaves based of the physical proximity of the class they represent. This way, we think apparent teleportation could be avoided as a bad reading at one antenna would likely produce a classification closer to the right zone. Finally, the future work we intend to do is to use this tracking system for activity recognition within the DOMUS laboratory. The resulting expert system will most likely be coupled with a cooking assistant for people suffering from a head trauma to follow them as they cook and offer precise help on the recognized steps.

8.1. Contributions

As stated in the introduction, the contributions of this paper are three-folds. The first contribution is practical and consists of an extensive dataset collected from a real smart home setup. The DOMUS laboratory is a real and complete smart apartment where people could live. It possesses everything we expect to find in a housing. While collecting the RFID readings, there were real obstacles between the bottle and the readers, like chairs around the table and a kettle on the counter. A computer simulation cannot reproduce those readings.

The second contribution concerns our methodology. While many researchers focus their work on the algorithm, we instead tried to model the problem differently and apply smart filters. As the first part of this work shows, most algorithms perform well and thus the learning part is not what has the biggest impact on tracking. To the best of our knowledge, few researchers tried to consider the localization and the tracking problems as classification problems. This qualitative localization seems less precise than a Cartesian localization. However, as shown in Bouchard et al. (2013), qualitative spatial information is easier to use for ADL recognition by expert systems in smart homes.

The third contribution comes from our tracking experiment, where our realistic paths showed some theoretical limit to our localization data. Still, those limits formed recognizable patterns that we could detect and correct by designing some simple filters.

9. Acknowledgements

We thank the INTER research group and the FRQNT for their financial support.

10. References

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, *6*, 37–66.

Altini, M., Brunelli, D., Farella, E., & Benini, L. (2010). Bluetooth indoor localization with multiple neural networks. In *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on* (pp. 295–300). IEEE.

Arora, R. (2012). Comparative analysis of classification algorithms on different datasets using weka. *International Journal of Computer Applications*, *54*.

Bekkali, A., Sanson, H., & Matsumoto, M. (2007). Rfid indoor positioning based on probabilistic rfid map and kalman filtering. In *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on* (pp. 21–21). IEEE.

Bouchard, K., Fortin-Simard, D., Gaboury, S., Bouchard, B., & Bouzouane, A. (2013). Accurate trilateration for passive rfid localization in smart homes. *International Journal of Wireless Information Networks*, *21*, 32–47. URL: `http://dx.doi.org/10.1007/s10776-013-0234-4`. doi:`10.1007/s10776-013-0234-4`.

Bouckaert, R. R. (2008). Bayesian network classifiers in weka for version 3-5-7. *Artificial Intelligence Tools*, *11*, 369–387.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Brusey, J., Floerkemeier, C., Harrison, M., & Fletcher, M. (2003). Reasoning about uncertainty in location identification with rfid. In *Workshop on Reasoning with Uncertainty in Robotics at IJCAI* (pp. 23–30).

Calderoni, L., Ferrara, M., Franco, A., & Maio, D. (2015). Indoor localization in a hospital environment using random forest classifiers. *Expert Systems with Applications*, *42*, 125–134.

Chahuara, P., Portet, F., & Vacher, M. (2013). Making context aware decision from uncertain information in a smart home: A markov logic network approach. In *International Joint Conference on Ambient Intelligence* (pp. 78–93). Springer.

Chai, J., Wu, C., Zhao, C., Chi, H.-L., Wang, X., Ling, B. W.-K., & Teo, K. L. (2017). Reference tag supported rfid tracking using robust support vector regression and kalman filter. *Advanced Engineering Informatics*, *32*, 1–10.

Chan, M., Campo, E., Estève, D., & Fourniols, J.-Y. (2009). Smart homescurrent features and future perspectives. *Maturitas*, *64*, 90–97.

Chawathe, S. S. (2008). Beacon placement for indoor localization using bluetooth. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on* (pp. 980–985). IEEE.

Cheng, C.-Y., Lan, T.-H., & Chan, C.-H. (2013). An improved localization algorithm with wireless heartbeat monitoring system for patient safety in psychiatric wards. *Engineering Applications of Artificial Intelligence*, *26*, 905 – 912. URL: `http://www.sciencedirect.com/science/article/pii/S0952197612003090`. doi:`http://dx.doi.org/10.1016/j.engappai.2012.11.013`.

Cook, D. J., & Krishnan, N. (2014). Mining the home environment. *Journal of intelligent information systems*, *43*, 503–519.

Cooper, G. F., & Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, *9*, 309–347.

Crow, B. P., Widjaja, I., Kim, L., & Sakai, P. T. (1997). Ieee 802.11 wireless local area networks. *IEEE Communications magazine*, *35*, 116–126.

Fahim, M., Fatima, I., Lee, S., & Lee, Y.-K. (2012). Daily life activity tracking application for smart homes using android smartphone. In *Advanced Communication Technology (ICACT), 2012 14th International Conference on* (pp. 241–245). IEEE.

Fortin-Simard, D., Bouchard, K., Gaboury, S., Bouchard, B., & Bouzouane, A. (2012). Accurate passive rfid localization system for smart homes. In *Networked Embedded Systems for Every Application (NESEA), 2012 IEEE 3rd International Conference on* (pp. 1–8). IEEE.

Friedman, J., Hastie, T., Tibshirani, R. et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, *28*, 337–407.

Gama, J. (2004). Functional trees. *Machine Learning*, *55*, 219–250.

Giroux, S., Leblanc, T., Bouzouane, A., Bouchard, B., Pigot, H., & Bauchet, J. (2009). The praxis of cognitive assistance in smart homes. In *BMI Book* (pp. 183–211).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*, 10–18.

Harter, A., Hopper, A., Steggles, P., Ward, A., & Webster, P. (2002). The anatomy of a context-aware application. *Wireless Networks*, *8*, 187–197.

He, J., Geng, Y., & Pahlavan, K. (2014). Toward accurate human tracking: Modeling time-of-arrival for wireless wearable sensors in multipath environment. *IEEE Sensors Journal*, *14*, 3996–4006.

Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., & Hall, M. (2002). Multiclass alternating decision trees. In *Machine learning: ECML 2002* (pp. 161–172). Springer.

Hutabarat, D. P., Patria, D., Budijono, S., & Saleh, R. (2016). Human tracking application in a certain closed area using rfid sensors and ip camera. In *Information Technology, Computer, and Electrical Engineering (ICITACEE), 2016 3rd International Conference on* (pp. 11–16). IEEE.

Joshi, G. P., Acharya, S., Kim, C.-S., Kim, B.-S., & Kim, S. W. (2014). Smart solutions in elderly care facilities with rfid system and its integration with wireless sensor networks. *International Journal of Distributed Sensor Networks*, *2014*.

Kim, S.-C., Jeong, Y.-S., & Park, S.-O. (2013). Rfid-based indoor location tracking to ensure the safety of the elderly in smart home environments. *Personal and ubiquitous computing*, *17*, 1699–1707.

Knighten, J., McMillan, S., Chambers, T., & Payton, J. (2015). Recognizing social gestures with a wrist-worn smartband. In *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on* (pp. 544–549). IEEE.

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD* (pp. 202–207). Citeseer.

Kourogi, M., Sakata, N., Okuma, T., & Kurata, T. (2006). Indoor/outdoor pedestrian navigation with an embedded gps/rfid/self-contained sensor system. *Advances in Artificial Reality and Tele-Existence*, (pp. 1310–1321).

Krishnan, N. C., & Cook, D. J. (2014). Activity recognition on streaming sensor data. *Pervasive and mobile computing*, *10*, 138–154.

Kwok, N., Liu, D., & Dissanayake, G. (2006). Evolutionary computing based mobile robot localization. *Engineering Applications of Artificial Intelligence*, *19*, 857 – 868. URL: `http://www.sciencedirect.com/science/article/pii/S0952197606000509`. doi:`http://dx.doi.org/10.1016/j.engappai.2006.01.020`.

Landwehr, N., Hall, M., & Frank, E. (2005). Logistic model trees. *Machine Learning*, *59*, 161–205.

LitePoint Corporation (2013). Ieee 802.11ac: What does it mean for test? URL: `http://litepoint.com/whitepaper/80211ac_Whitepaper.pdf` [Online; accessed 27-April-2018].

Liu, X., Peng, J., & Liu, T. (2011). A novel indoor localization system based on passive rfid technology. In *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on* (pp. 4285–4288). IEEE.

Ma, H., Wang, Y., & Wang, K. (2018). Automatic detection of false positive rfid readings using machine learning algorithms. *Expert Systems with Applications*, *91*, 442–451.

Majeed, I., & Arif, O. (2016). Non-linear eigenspace visual object tracking. *Engineering Applications of Artificial Intelligence*, *55*, 363 – 374. URL: `http://www.sciencedirect.com/science/article/pii/S0952197616301361`. doi:`http://dx.doi.org/10.1016/j.engappai.2016.08.005`.

Martin Woolley (2018). Bluetooth 5. go faster. go further. URL: `https://www.bluetooth.com/bluetooth-technology/bluetooth5/bluetooth5-paper` [Online; accessed 27-April-2018].

Mehr, H. D., Polat, H., & Cetin, A. (2016). Resident activity recognition in smart homes by using artificial neural networks. In *Smart Grid Congress and Fair (ICSG), 2016 4th International Istanbul* (pp. 1–5). IEEE.

Misra, P., & Enge, P. (2006). *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press.

Morato, C., Kaipa, K. N., Zhao, B., & Gupta, S. K. (2014). Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *Journal of Computing and Information Science in Engineering*, *14*, 011006.

Morganti, E., Angelini, L., Adami, A., Lalanne, D., Lorenzelli, L., & Mugellini, E. (2012). A smart watch with embedded sensors to recognize objects, grasps and forearm gestures. *Procedia Engineering*, *41*, 1169–1175.

Ni, L. M., Liu, Y., Lau, Y. C., & Patil, A. P. (2004). Landmarc: indoor location sensing using active rfid. *Wireless networks*, *10*, 701–710.

Ou, C.-W., Chao, C.-J., Chang, F.-S., Wang, S.-M., Liu, G.-X., Wu, M.-R., Cho, K.-Y., Hwang, L.-T., & Huan, Y.-Y. (2017). A zigbee position technique for indoor localization based on proximity learning. In *Mechatronics and Automation (ICMA), 2017 IEEE International Conference on* (pp. 875–880). IEEE.

Pirsiavash, H., & Ramanan, D. (2012). Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 2847–2854). IEEE.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Ransing, R. S., & Rajput, M. (2015). Smart home for elderly care, based on wireless sensor network. In *Nascent Technologies in the Engineering Field (ICNTE), 2015 International Conference on* (pp. 1–5). IEEE.

Rossey, J., Moerman, I., Demeester, P., & Hoebeke, J. (2016). Wi-fi helping out bluetooth smart for an improved home automation user experience. In *Communications and Vehicular Technologies (SCVT), 2016 Symposium on* (pp. 1–6). IEEE.

de Sá, A. O., Nedjah, N., & de Macedo Mourelle, L. (2016). Distributed efficient localization in swarm robotics using min–max and particle swarm optimization. *Expert Systems with Applications*, *50*, 55–65.

Tesoriero, R., Tebar, R., Gallud, J. A., Lozano, M. D., & Penichet, V. M. R. (2010). Improving location awareness in indoor spaces using rfid technology. *Expert Systems with Applications*, *37*, 894–898.

U.S. Air Force (2017). Gps accuracy. URL: `https://www.gps.gov/systems/gps/performance/accuracy/` [Online; accessed 27-April-2018].

Webb, G. I. (1999). Decision tree grafting from the all-tests-but-one partition. In *IJCAI* (pp. 702–707). Citeseer.

Wikipedia contributors (2018). Radio-frequency identification — Wikipedia, the free encyclopedia. URL: `https://en.wikipedia.org/w/index.php?title=Radio-frequency_identification&oldid=836773617` [Online; accessed 27-April-2018].

Wu, D., & Butz, C. (2005). On the complexity of probabilistic inference in singly connected bayesian networks. In *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing* (pp. 581–590). Springer.

Yim, J. (2008). Introducing a decision tree-based indoor positioning technique. *Expert Systems with Applications*, *34*, 1296–1302.

ZigBee Alliance (2013). Zigbee specification faq. URL: `https://web.archive.org/web/20130627172453/http://www.zigbee.org/Specifications/ZigBee/FAQ.aspx` [Online; accessed 27-April-2018].

Zulas, A. L., Crandall, A. S., & Schmitter-Edgecombe, M. (2014). Caregiver needs from elder care assistive smart homes children of elder adults assessment. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (pp. 634–638). SAGE Publications volume 58.