Received 26 April 2016; | Revised 6 June 2016; | Accepted 6 June 2016

DOI: xxx/xxxx

INTERNET TECHNOLOGY LETTERS

Indoor Proximity Detection: the Case Study of a Smart Pet Door

Simone Borreggine[1] | Pietro Serafino[1] | Pietro Boccadoro[1,2]

[1]Department of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy, Email: {name.surname}@poliba.it
[2](CNIT), Consorzio Nazionale Interuniversitario delle Telecomunicazioni.

Summary

The Internet of Things era is revolving sensing and communication capabilities from industrial applications to daily life use-cases. Some of them concern home automation towards the Internet of Animals, in which smart objects help pets' owners to smartly monitor animals. These applications are made possible by low-power technologies enabling interactions based on proximity. This work demonstrates the suitability of both open source rapid prototyping solutions and Bluetooth Low Energy in proximity based applicative scenarios. Here, Received Signal Strength Indication is enhanced with classification algorithms. By comparing Neural Networks and KNearest Neighbor solutions, the system effectively realizes secure access management and continuous monitoring. Classification precision reached the 80% in almost all cases and betters as distance decreases.

KEYWORDS:
Internet of Things, proximity detection, bluetooth low energy, neural network

1 | INTRODUCTION

The Internet of Things (IoT) is a revolutionary paradigm giving raise to a number of smart applications in a huge variety of fields, from environmental monitoring to industrial plants optimization[1]. The IoT contemplates heterogeneous solutions[2] spanning from Wide Area Networkss (WANs) to Personal Area Networks (PANs) scenarios, respectively adopting Low-Power Wide Area Network (LPWAN) or short-range communication protocols. In the latter domain, the Bluetooth Low Energy (BLE)[3] is widely adopted since it specifically addresses one of the major constraints of IoT scenario, for instance low power consumption. This major enhancement gave rise to a massive diffusion of fully-compliant low-cost smart devices and it is nowadays a catalysts for smart objects interactions[4]. This work tackles the so-called Internet of Animals (IoA), addressing monitoring and assistance of animals in domestic environments. The envisioned solution relies on a low-cost platform implementation with smart devices, tracking interactions between animals and their pet doors. BLE devices and open source rapid prototyping technologies, both hardware and software, have been used to create an intelligent tracking system able to locate the pet when it is close to the door, understand how close it is, and securely open the door. The use of the BLE technology has been necessary to perform a data exchange between the smart wearable device (i.e., the Lilypad), and the management station (i.e., a Raspberry Pi). Received Signal Strength Indication (RSSI) values coming from the mobile device have been tested and verified as a proximity indicator. Since radio frequency signals suffer of problems such as multi-path propagation, the simple RSSI could not be sufficient for ranging purposes. Therefore, classification algorithms have been involved in order to improve distance estimation. In particular, a cross-comparison of different classification models (e.g. BallTree, KDTree and AutoKNN) has been carried out to choose the solution that better fits application requirements, for instance a minimum detectable distance of one meter. The present contribution represents a step ahead in the state of the art as it propose a smart home automation application in the IoA context. Even if something similar has been proposed in[5], the solution does not allow any expansion, since it is not based on open source platforms. In this work, instead, a Smart Pet Doors relying on BLE is presented, leveraging low-cost rapid prototyping devices to handle proximity detection. The remainder of the work is as follows: Section 2 focuses on both outdoor and indoor localization problem and motivates the adoption of BLE. Section 3 describes the proposed case study. In Section 4, the experimental proposes the analysis and discusses obtained results. Section 5 concludes the work and proposes future possibilities.

2 | ESSENTIAL BACKGROUND

Proximity detection has been considered an hot research topic for a long time in both indoor and outdoor scenarios. Global Position System (GPS) has been widely used outdoor, to enable cars and people to localize themselves and planning or tracking their movements[6]. Unfortunately, the GPS signal cannot be received indoor; therefore, different technologies and approaches have been studied, such as Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA)[7]. In addition, several localization techniques have been proposed so far, which led to a large variety of algorithms aiming at obtaining nodes locations with low computational costs and restricted data set[8]. In such solutions, a small number of nodes, with known positions (also known by the name of anchors or reference points) advertise their locations in order to assist nodes with unknown locations (also referred to as unknown or normal nodes) to estimate their coordinates. Anchors are generally broadcasting their position to advertise neighboring entities. In[9], BLE have been considered in order to experiment radio localization procedures for proximity detection in short range indoor applications. In particular, Bluetooth is used to map signal strength of nearby devices, report registered values to a centralized server and trigger actions. The accuracy of such method has been investigated to date, also in Motion-assisted Device Tracking (MADT)-based system. Wearable BLE devices can also describe human behaviors and detect individual proximity. Embedded devices with BLE-compliant chipsets are low cost, which results in low price tracked tags. Moreover, in Bluetooth-based positioning systems, a group of devices may form a cluster, which lowers the effort of each in locating a specified device. Due to well-known problems that affect radio propagation (e.g., shadowing and multipath fading), RSSI is not sufficient for precise ranging. To bridge this gap, machine learning methods can be used to enhance ranging precision, such as (i) BallTree[10], (ii) K-D Tree[11], and (iii) AutoKNN[12]. In particular, Balltrees are geometric data structures that can be used for spacial indexing in multi-dimensional areas. K-D Trees, instead, are special cases of binary partitioning trees in which every node is a k-dimensional space. These data structure are useful in range and/or nearest neighbor searches. The third possibility is K-Nearest Neighbor (KNN), a non-parametric pattern recognition method[13], for instance, the simplest and yet most efficient classification rules.

3 | PROPOSED CASE STUDY

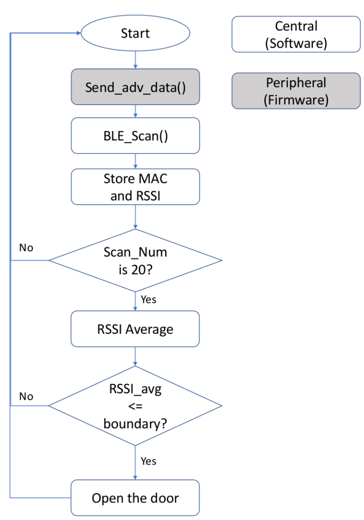

FIGURE 1 Flowchart of RSSI analysis.

A wearable device, which is mobile by design, communicates with a smart door opening system in order to trigger consequent actions, for instance open or keep it closed. The interaction between the mobile device and the smart door is triggered by the RSSI evaluation and analysis performed through BLE. To achieve the door opening functionality, the central node processes information acquired from a devices scanning procedure to evaluate if the door must be opened (Figure 1). The single scan lasts 250 ms and the activity is repeated ten times, thus resulting in a 2.5 s scanning period and the shortest possible period between two consecutive door openings. These values can be considered as general purpose settings, thus allowing transmitting and receiving data in a wide range of applications. By scanning the Advertising Data of each near device, the central node acquires two values: (i) the Media Access Control (MAC) addresses of the found devices and (ii) their RSSI. For known devices (those mapped and already known by their MAC address), a set of ten RSSI values are measured and averaged to reduce essential fluctuations. The resulting value is compared with a reference threshold, derived by classification models, to consequently trigger door opening. Afterward, the scanning cycle starts again.

4 | EXPERIMENTAL PERFORMANCE EVALUATION

4.1 | Hardware and software components

The developed system is completely based on low-cost, open source rapid prototyping development solutions. The choices are strongly motivated by the clear aim to allow future developments and enhancements possibilities. The main hardware components are: a Raspberry Pi 2 Model B[1] and a BLE module HM-10. The former is a small-sized single-board computer powered by a 1.2 GHz quad-core ARM Cortex A53 Central Processing Unit (CPU) with 1 GB of Low Power Double Data Rate type two (LPDDR2) Random Access Memory (RAM). It features a BLE antenna with symmetric encryption algorithm Advanced Encryption Standard (AES)-128 key and a maximum range of 50 meters (so it does not require any external component to support BLE communications). The Raspberry Pi senses and monitors close devices. Measured RSSI values are, then, averaged to perform classification tasks in order to evaluate the real distance and, eventually actuate door opening mechanism. The latter, instead, is a radio transceiver, featuring the Texas Instruments CC2541[2] System on Chip (SoC). It works in the 2.4 GHz Industrial, Scientific and Medical (ISM) band, with a transmission power up to 4 dBm. It is used to emulate the BLE module worn by the pet approaching the door, whose transmitted RSSI is evaluated by the Raspberry. As for the software, the operating system installed on the central node (for instance, the Raspberry Pi) is Raspbian, a Debian-based open source distribution optimized for Advanced RISC Machine (ARM) architecture. On top of it, a Python routine, based on the Bluepy software library[3], leverages dedicated Application Programming Interfaces (APIs) for enabling interactions between the operating system and the BLE components. The library enables scanning functionalities, used to read the RSSI values related to each device in visibility. Once values are acquired, the evaluation of the distance between the transmitter and the receiver is made through dedicated algorithms. Specifically, Artificial Neural Network (ANN) and KNN have been chosen, as they have been widely investigated in studies concerning RSSI analysis, indoor positioning and localization problems. To create classifications models, the scikit-learn library[4] has been used, as it eases the implementation of machine learning algorithms in Python. The library provides two classification classes MPLClassifier and KNeighborClassifier; the former allows a multi-layer neural network model, whereas the latter enables the K-Nearest Neighbor model usage. For the sake of completeness, (i) Brute Force, (ii) K-D Tree[11], (iii) Ball Tree[10] algorithms have been compared to obtain a KNeighborClassifier. Functionally speaking, Brute Force algorithm, also known as *exhaustive research*, verifies all possible solutions to a problem. This kind of research leads to a precise detection. Such precision has a downside effect: the convergence time is quite slow, thus resulting in potentially huge computational requirements. As a matter of fact, for N samples in D dimensions, this approach scales as $O[D * N^2]$, so Brute Force is neglected for huge datasets. To lower computational and time resources expense, a variety of tree-based data structures has been introduced. K-D Tree is a binary tree structure which recursively partitions the parameter space along the data axes. This allows to avoid D-dimensional distances computation. The nearest neighbor of a query point is defined in a $O[log(N)]$ distance computation. K-D Tree results to be efficient and fast for low dimensional problem[11]. As the problem scales up, inefficiencies of K-D Trees can be solved developing a Ball Tree data structure. This approach does not partition data along Cartesian axes but in hyper-spheres, which greatly suites multi-dimensional problems[10].

[1] https://www.raspberrypi.org/documentation/hardware/computemodule/RPI-CM-DATASHEET-V1_0.pdf
[2] http://www.ti.com/lit/ds/symlink/cc2541.pdf
[3] https://github.com/IanHarvey/bluepy
[4] http://scikit-learn.org/stable/

4.2 | Data acquisition and analysis

During data acquisition, the mobile BLE module was located at known distances from the sensing unit (e.g., the Raspberry Pi). Given RSSI's variability, a precise mapping and the identification of specific distance-related thresholds were mandatory. Distances ranged from 0.25 and 2 m, with a 0.25 m pace. During acquisition phase, the mobile BLE device transmitted Advertising Data with a transmission power of 0 dBm and a 10 ms period. The value set for transmission power has been chosen as to maximize the range of the radio signal. The rationale for the transmission period is related to the possibility of reaching a longer evaluation distance. The central unit has been set up to perform scanning every 50 ms. Those durations have been chosen to allow successful transmissions and receptions of data between Raspberry and the BLE device. To create a consistent dataset, a thousand RSSI values have been acquired for each distance. It clearly emerged that the higher the distance, the more similar the mean (Figure 2), thus resulting in a more difficult discrimination for values beyond 1.25 m. Further, variance values increase together with distance, which leads to uncertainties and suggests the need for dedicated estimations correction routines.

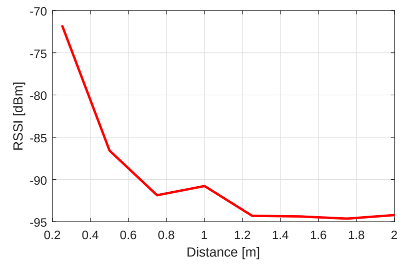

FIGURE 2 Trend of the mean of received RSSI values.

4.3 | Classification models training

To classify measured values, the dataset has been organized as follows: (i) Class 0, with RSSI value for a distance of 0.25 m, (ii) Class 1, with RSSI value for a distance ranging from 0.25 m up to 1.25 m, and (iii) Class 2, with RSSI value for a distance greater than 1.25 m. Classification models have been created to verify the existence of a relationship between RSSI recurrence and distances. To this aim, Neural Networks and K-Nearest Neighbor have been implemented. For both, the 70% of the whole dataset has been used for training purposes, whereas the remaining 30% as a real dataset to evaluate accuracy. The resulting value is used to indicate the percentage of correctly predicted data points out of the processed data. Neural network has been trained with a stochastic gradient descent, a logistic activation function and a low learning parameter. In order to find the best model, different number of hidden layers and neurons have been taken into account. A Neural network with only one hidden layer has shown the best accuracy whit a neuron population of 400 (Figure 3). The confusion matrix of the network (Table 1)

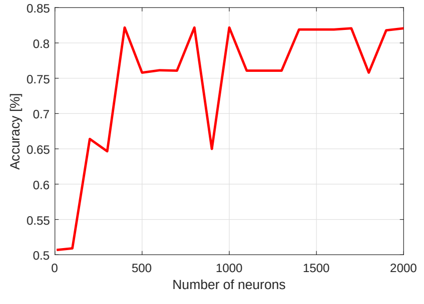

FIGURE 3 Neural Network accuracy with one hidden layer.

demonstrates that Class 0 is always correctly identified, whereas misclassifications appear for the other two classes, with a higher recurrence near the fringe between Class 1 and Class 2. For the sake of verifying the obtained results, a second hidden layer has

	Class 0	Class 1	Class 2
Class 0	111	0	0
Class 1	2	239	91
Class 2	0	61	368

TABLE 1 Classified RSSI values.

been added, but the comparison showed lower accuracy values, due to an over-fitting phenomenon. Overall, the model shows a 82% precision, with a 96% peak, for Class 0. Therefore, the resulting Neural Network (NN) model can predict the distance based on the following RSSI values: Class 0 (from 0dBm to -76.3dBm), Class 1 (from -76.3dBm to -89.5dBm), and Class 2 (beyond -89.5dBm). To apply the K-Nearest Neighbor method, several values of k [$\in$ 0, 1000] have been tested to maximize accuracy. The set of values was increased until accuracy did no longer experience improvements. The classification procedure can be considered as general, and strictly dependent on the training set. Results for the BallTree algorithm, together with the Brute one, are reported in Figure 4 . In the first case, the higher accuracy value recorded was 82.86%, with k=6. With Brute algorithm, the

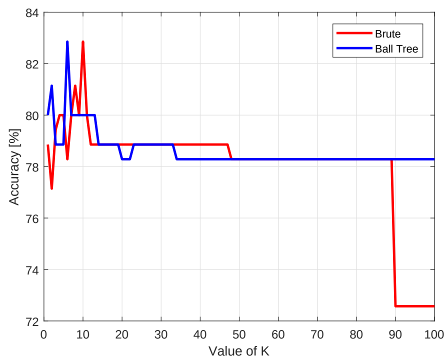

FIGURE 4 Ball tree and Brute accuracy trend due to K values changing.

same accuracy value has been achieved with a higher value of k (for instance, 10). Training continued with a fitting procedure using each algorithm and the two measured values of k (6 and 10), depending on the usage of Ball Tree or Brute algorithms. The results reflect the different KNN implementations. As k goes above 90, performances do not further improve for Ball Tree, whereas the Brute case worsens. The trend can be explained as the Brute solution has a fast convergence but coarser value. This leads to unstable results when used on large data set, which, overall, affects performance.

4.4 | Proximity detection results

To verify the theoretical results, a new dataset was created, with 50 averaged RSSI values for each class. Acquisitions were performed in 20 rounds, 0.5 seconds long each. The number and period of the scans have been chosen to obtain a simulation of the operative conditions in which the automatic opening door could be used. Measured values were obtained changing the receiver's positions within the related identified class. Changes in position were useful to simulate a general use case and to analyze the whole distance spectrum related to the tested class. The class to which each measure belongs is predicted by comparing the averaged RSSI value, taken at certain distances, with the RSSI boundaries created by the classification models. The results demonstrated that it is always possible to discriminate Class 0 (almost 100% accuracy). Thanks to the NN boundaries, the predictions were wrong in only one case out of the total of 50. However, misclassifications occurred for Class 1 values, wrongly attributed to Class 2. The KNN results resulted similar to the NN ones: misclassifications occur for Class 1, whereas is a 100%

discrimination of Class 0 distances. To prove the need for averaging procedure, a set of non averaged RSSI values for Class 0 and Class 1 have been used. Here, for Class 0, RSSI values ≥ -80 are misclassified, thus leading to wrong distances evaluations.

5 | CONCLUSIONS AND FUTURE RESEARCH

This work focused on proximity detection and BLE's suitability in IoA applications. The experimental setup efficiently uses RSSI values to leverage the potential of open source low-cost rapid prototyping hardware and software solutions to realize a Smart Pet Door. To improve detection, different classification methods have been tested, thus allowing a minimum 80% precision in close proximity conditions (within 1 meter). In the future, the setup will be evaluated in terms of (i) different experimental conditions, (ii) increased number of connected devices, with a (iii) focus on performances and energy footprint. Moreover, step counting algorithms [4] will be considered.

References

1. Davidson P., Piché R.. A Survey of Selected Indoor Positioning Methods for Smartphones. *IEEE Communications Surveys Tutorials.* 2017;19(2):1347-1370.
2. Lin J., Yu W., Zhang N., Yang X., Zhang H., Zhao W.. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal.* 2017;4(5):1125-1142.
3. Bluetooth SIG . *Bluetooth Specification v.4.1.* 2013.
4. Gu F., Khoshelham K., Shang J., Yu F., Wei Z.. Robust and Accurate Smartphone-Based Step Counting for Indoor Localization. *IEEE Sensors Journal.* 2017;17(11):3453-3460.
5. Own C. M., Teng C. Y., Zhang J. R., Yuan W. Y., Tsai S. C.. Intelligent pet monitor system with the internet of things. *Proc. of International Conference on Machine Learning and Cybernetics.* 2011;2:471-476.
6. Ferreira A. F. G., Fernandes D. M. A., Catarino A. P., Monteiro J. L.. Localization and Positioning Systems for Emergency Responders: A Survey. *IEEE Communications Surveys Tutorials.* 2017;19(4):2836-2870.
7. Patwari N., Ash J. N., Kyperountas S., Hero A. O., Moses R. L., Correal N. S.. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine.* 2005;22(4):54-69.
8. Mahjri I. Dhraief A. Drira K., A. Belghith. Weighted localization in mobile wireless networks. *Internet Technology Letters.* ;1(1):e12.
9. Faragher R., Harle R.. Location Fingerprinting With Bluetooth Low Energy Beacons. *IEEE Journal on Selected Areas in Communications.* 2015;33(11):2418-2428.
10. Nitin Bhatia Vandana. Survey of Nearest Neighbor Techniques. *International Journal of Computer Science and Information Security.* 2010;8(2):302-305.
11. Nielsen F. Barlaud M.. Tailored Bregman Ball Trees for Effective Nearest Neighbors. *Proc. of European Workshop on Computational Geometry.* 2009;37:29-32.
12. Lodwich Aleksander, Shafait Faisal, Breuel Thomas. Efficient Estimation of k for the Nearest Neighbors Class of Methods. *arXiv preprint arXiv:1606.02617.* 2016;.
13. Fuli Jiang, Chu Chu. Application of kNN improved algorithm in automatic classification of network public proposal cases. *Proc. of IEEE International Conference on Cloud Computing and Big Data Analysis.* 2017;:82-86.

How to cite this article: Simone Borreggine, Pietro Serafino and Pietro Boccadoro (2018), Indoor Proximity Detection: the Case Study of a Smart Pet Door, *Internet Technology Letters*, *2018;00:1–6*.