Comm2Sense: Detecting Proximity Through Smartphones

Iacopo Carreras, Aleksandar Matic, Piret Saar and Venet Osmani
CREATE-NET
Trento, Italy
name.surname@create-net.org

***Abstract*—Mobile devices are increasingly becoming a proxy to habits and daily patterns of users. This is opening new possibilities in the area of human behaviour analysis, where the many sensors available on smartphones can be used to regularly monitor users' daily activities and interactions in an non-obtrusive way. In this paper we focus on proximity detection, which refers to the ability of a system to recognize the co-presence of two or more individuals. We present *Comm2Sense*, our mobile platform to detect proximity among users exploiting sensing capabilities available in modern smartphones, namely Wi-Fi hotspot and Wi-Fi receiver. The platform estimates the distance between subjects applying data mining techniques to the analysis of the Wi-Fi RSSI. We describe the design and implementation of the platform, together with the technical solutions implemented in each module. We demonstrate that the proposed platform is able to achieve a resolution of 0,5 meters.**

***Keywords*-mobile phone sensing, proximity detection, social computing, middleware.**

I. Introduction

With the advent of modern mobile handsets, sensing is now moving "Out of the Wood" [1] and becoming ubiquitous within everyday life. Handsets that were once exclusively used for voice communications, are nowadays an extremely powerful mobile platform, able to sense the surroundings with various means (e.g., GPS, accelerometer, compass, WiFi, etc.), to store such data locally and to eventually send it to a remote repository for some higher interpretation. Given their personal nature, mobile devices are increasingly becoming a proxy to users' habits and daily patterns. This is opening new possibilities in the area of human behaviour analysis, where the many sensors typically embedded in smartphones can be used to regularly monitor users' daily activities and interactions in an non-obtrusive way. As an example, through the use of GPS it is possible to know users' location over time, while with accelerometers it is possible to infer users' physical activities.

In this paper we focus on proximity detection, which refers to the ability of a system to recognize the co-presence of two or more individuals and it is closely related to a number of research efforts which apply the combination of sensing and modelling paradigms to support the monitoring and understanding of human dynamics. Proximity among individuals has a relevance to a number of different areas, including (but not limited to) i) healthcare - where the amount of socialization may have a direct, positive impact on the self-reported mood of people, ii) social network analysis - where interactions are seen as a proxy towards the social graph of a person [2], iii) productivity - where social interactions are shown to be correlated to the productivity [3] and iv) epidemiology - where contacts among people represent the main cause behind the spreading of an epidemic.

Most of the approaches to monitoring social interactions required users to wear a sensing device. For example, research work carried out in [2] used the *Sociometer*, a wearable Social Badge to analyse proximity of users and their interactions. The *Sociometer* consisted of a wearable badge to be carried by users during their daily activities and, in its initial implementation, was able to detect i) physical proximity through the use of infrared technology and ii) conversations from segments of raw audio. In a similar manner, authors in [4] used an active badge to measure social interactions in closed environments such as hospitals or conferences. The use of dedicated devices provides a fine granularity on the data being collected, but is limited to supervised environments and for limited periods of time (the battery life-time of the dedicated device). Furthermore, since such devices are not part of individuals' everyday life, they can be perceived as obtrusive, or simply may not be always carried by users. Finally, they typically rely on an external infrastructure to collect the data for a later analysis. Therefore, solutions based on dedicated devices are appropriate for scenarios confined in space and time, such as conferences, enterprise environments, hospitals and so on.

Alternative approaches based on the use of smartphones have been proposed in the literature: they can follow users during their daily patterns, while their increasing computational, communication and sensing power afford performing complex tasks. Through the use of smartphones, proximity can be either sensed directly or inferred in terms of users' "co-location". In the former case, the sensing capabilities of mobile devices are exploited to detect nearby users. As an example, by performing a Bluetooth scan, the presence of nearby Bluetooth-enabled devices can be captured and this information can be used to infer users' proximity [5]. In the case of co-location, proximity is inferred whenever

The work of Iacopo Carreras and Piret Saar has been supported by the European Commission within the framework of the EPIWORK project ICT-2007.8.4-231807, www.epiwork.eu.

users share the same spatial location. This can be measured through the use of Bluetooth beacons [6] or Wi-Fi fingerprinting [7]. All these approaches, while being fully distributed, only provide a coarse spatial granularity, in the order of ten meters.

To the best of our knowledge, there is only one work that exploits transmitting mechanisms embedded in smartphones to achieve proximity detection with a granularity of a few meters. The project, called Virtual Compass [8], detects nearby mobile devices with an average accuracy of 1.9 m. However, in order to achieve high resolution distance estimation, the proposed system relies on more neighboring devices to calculate inter-distances in a two dimensional plane.

In contrast to previous studies, we propose *Comm2Sense*, a mobile platform that exploits the sensors embedded in mobile phones to detect the proximity between two (or more) individuals. The approach is based on the ability of modern devices to act both as Wi-Fi transmitters (known as *tethering* or *portable hot spot* mode) and Wi-Fi receivers. We exploit this feature to estimate the distance between a transmitting and a receiving peer by analyzing RSSI (Received Signal Strength Indicator), which provides the signal strength of the transmitting peer.

The proposed approach has four significant advantages with respect to the state of the art: i) it does not rely on any external infrastructure or dedicated devices, thus facilitating its wide deployment iii) it is based on pair-wise interactions, requiring the information solely from two, or more, phones in order to infer proximity iii) users are not constantly aware of the behavior measurement taking place, since it relies on users' personal mobile phone and iv) it provides a distance estimation median accuracy between 0.5 and 1 meter over a wide range of environments. The main contribution of this paper can be summarized as follows:

- We present *Comm2Sense*, a platform for proximity detection. This is achieved by estimating distances between smartphones through data mining techniques applied to the Wi-Fi RSSI.
- We introduce a duty-cycle for letting smartphones alternatively act as portable hot spots and Wi-Fi clients. This is a necessary requirement to enable phones to mutually detect each other.
- We present a method for deploying the system, without the need for time demanding calibration procedures.

The reminder of this paper is organized as follows. In Sec. II, we present the design of the *Comm2Sense* system, including the algorithmic solutions implemented. Sec. III, we draw the main conclusions of this work.

II. *Comm2Sense* System Design

In this section, we present the design of *Comm2Sense*, our proximity detection system intended for detecting users' proximity using technologies available in modern smartphones. Our approach to proximity detection between two smartphones (one acting as a Wi-Fi receiver, the other one as a Wi-Fi transmitter) is based on the Wi-Fi RSSI analysis and data mining techniques in order to estimate the distance between them.

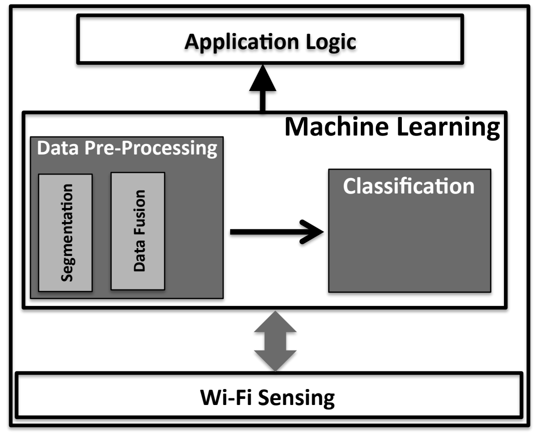

Figure 1. *Comm2Sense* software architecture.

A. Design Goals and Architecture

The main design goals of the *Comm2Sense* system can be summarized as follows:

- fine distance recognition: we expect the system to achieve a fine-grained distance recognition, in the order of a few meters, with a granularity of 0,5 meters. This will allow us to apply the proposed platform to a wide range of applications scenarios;
- not rely on external infrastructure: we expect *Comm2Sense* to operate in a fully distributed fashion, without relying on any external infrastructure, or on any additional hardware, but smartphones. Proximity should then be inferred solely on the basis of the information that is possible to acquire through the smartphone;
- ease of deployment: we expect the platform to be easily deployable, without requiring users to perform time-consuming configuration operations. Further, we expect the platform to be deployable over off-the-shelf components, without the need for dedicated hardware.

We have designed and implemented *Comm2Sense* over a legacy Android platform. The overall software architecture is shown in Fig. 1. It has been designed in the form of a middleware platform, which is decoupled from the application logic. The middleware autonomously detects proximity and issues call-backs to the application layer, whenever one is detected. The application is then responsible for handling such information in the most appropriate way, given the scenario being supported.

In the following we will describe each system component.

B. Wi-Fi Sensing

The Wi-Fi sensing module exploits the ability of modern smartphones to operate in a Portable Hot Spot (PHS) mode, as well as Wi-Fi clients. This allows them to be discovered by peers close by – when acting as PHS –, but also to discover other peers – when acting as Wi-Fi clients–. The Wi-Fi sensing module is in charge of sensing the presence of nearby PHSs and forward such data to the data pre-processing component.

1) Duty Cycling: In a "static" setting, nodes either act as Wi-Fi clients or PHS. However, in this case, while Wi-Fi clients can detect the presence of PHS, the opposite does not hold, thus making proximity detection uni-directional. This is visually explained in the upper part of Fig. 2: as it is possible to observe, while node A is able to detect the presence of node B (in PHS mode), the same does not apply to nodes B and C, since both are in hot spot mode and can only be discovered.

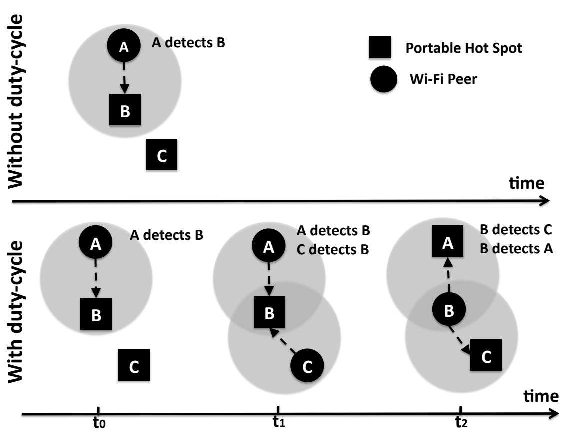

Figure 2. Proximity detection via RSSI with and without the duty-cycle.

In order to overcome this limitation and make proximity detection bi-directional, each node implements a duty-cycle, acting alternatively as PHS and Wi-Fi client. This is illustrated in the lower part of Fig. 2, where nodes change their role over time. As it is possible to observe, by time t_2 all nodes in the Wi-Fi communication range are able to discover each other.

The duty-cycle is depicted in Fig. 3 and comprises four states, each one characterized by the following permanence time:

- T_{HSM}: time spent in Portable Hot Spot (PHS) Mode. In this state, nodes are discoverable by other peers, but can not discover other peers. This parameter can be configured at design time.
- T_{SM}: time spent in Scan Mode (SM). In this state, node are searching for other peers at regular time intervals T_{SP}. In this state nodes can discover other PHSs, but are not discoverable by other peers. This parameter can be configured at design time.
- T_{WUM}: time required for switching from PHS to SM Mode. This time interval is composed by the time needed to switch OFF portable hot spot and the time needed to tear up the Wi-Fi client. In this time interval nodes are not discoverable, nor can discover other peers.
- T_{WDM}: time required for switching from SM to PHS Mode. This time interval is composed by the time needed to switch OFF the Wi-Fi client and the time needed to tear up the portable hot spot. In this time interval nodes are not discoverable, nor can discover other peers.

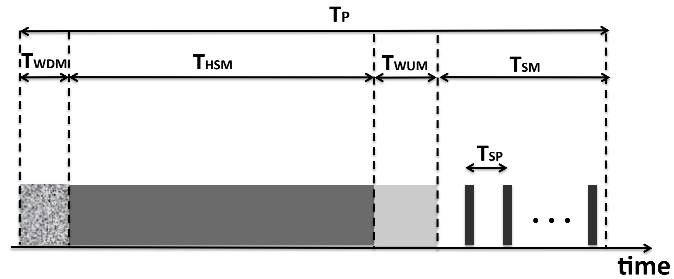

Figure 3. Duty cycle implemented by *Comm2Sense* nodes.

The duty-cycle will be defined at design time, and will be determined by the *time-resolution* to be achieved, where the time-resolution refers to the minimum amount of time that 2 nodes are required to stay in proximity in order to detect each other. Specific randomization is introduced in order to avoid synchronized duty-cycling of nodes. As an example, the time spent in PHS mode is not T_{HSM}, but rather uniformly distributed in $[0, 2 \times T_{HSM}]$
In order to properly identify the time spent in each one of the four states, we have first measured the time needed for tearing up and down a portable hot spot. The results for an HTC Nexus One are reported in Tab. 1.

	min (ms.)	mean (ms.)	max (ms.)	std (ms.)
PHS activation	n.a.	915	n.a.	n.a.
PHS deactivation	1246	1305	1764	206
WiFi activation	4452	4573	4990	76
WiFi deactivation	7	25	103	16

Table I
TIME NEEDED FOR TURNING ON AND OFF A PORTABLE HOT SPOT OVER AN HTC NEXUS ONE.

An important finding is that fully activating a PHS may require up to 10 sec. to complete. This is the time needed for having a fully functional PHS. However, it requires less than 1 sec. for the PHS, to start announcing its SSID, and therefore be discoverable by Wi-Fi clients in proximity. And since the Wi-Fi sensing is based only on the RSSI analysis available from the SSID, without the need to establish a data connection, this can be assumed as the time required to activate a PHS. Having accurate measurements of this would require dedicated equipment and this is the reason why we do not provide the complete statistics in the table.

With respect to the parameters of the *Comm2Sense* duty cycle, we have that T_{WUM} (PHS deactivation and Wi-Fi client activation), is approximatively 5878 ms., while T_{WDM} (Wi-Fi client de-activation and PHS activation) is approximatively 1025 ms.. T_{sp}, as measured from dedicated experiments, can be configured to be around $0,5$ sec..

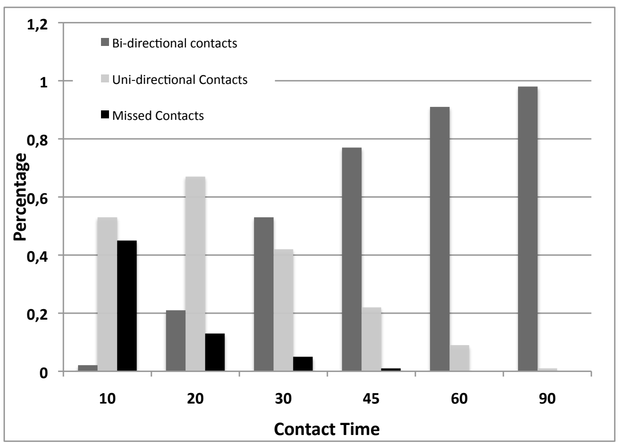

Figure 4. Proximity detection probability in the case of $T_{HSM} = 10$ sec. and $T_{SM} = 10$ sec., $T_{WUM} = 5.878$ sec and $T_{WDM} = 1.025$ sec..

In order to properly dimension the values of the duty cycle, we have implemented a simulator, which receives the system parameters (T_{WUM}, T_{WUM}, T_{HSM} and T_{SM}), and provides the probabilities of (i) missing a contact (ii) having uni-directional contacts (only one node detecting the presence of the other) (ii) having bi-directional contacts (nodes mutually discovering each other). The simulator can reproduce the case of two or more nodes meeting. In Fig. 6, we present the detection probability in the case of two nodes meeting, with $T_{HSM} = 10$ sec., $T_{SM} = 10$ sec $T_{WUM} = 5.878$ sec and $T_{WDM} = 1.025$ sec.. As it is possible to observe, already with a contact duration of 60 sec., nodes can be mutually detected each other with 80% probability.

This, while not excluding the possibility of missing contacts, it is sufficiently accurate for the considered application scenarios (e.g., healthcare, social network analysis) where relevant social interactions are supposed to last for more than 60 seconds.

2) Configuring the Transmission Power: The accuracy of the proposed approach strongly depends on the transmitting power of the Wi-Fi radio interface: the higher the transmission power, the larger the range and therefore the number of nodes receiving the RSSI signal of the PHS. Further, a higher transmission power leads to a limited accuracy for the estimation of distances in the short ranges.

In order to evaluate the feasibility of distance estimation based on Wi-Fi RSSI, and of its accuracy, we have then explored the RSSI dependence on distance for three different transmitting power levels: 32 dBm (1.6 W) maximal available power level, 0dBm (1 mW) - minimal power level, and 13 dBm (20 mW). We have performed experiments, recording 300 samples with the sampling rate of 1Hz for distances comprised between $0,5$ m. and 8 m, and for all power levels. Experiments were run on an HTC Desire phone, although similar results were obtained for other Android smartphones. The transmitting power of 0 dBm provided the smoothest and the most monotone characteristics thus proving to be the best fit for short distance estimation. We have then used 0 dBm as the reference transmission power for our system.

3) Implementation Notes: Setting the transmission power to 0 dBm and controlling the switching on and off of the portable hot spot can be done via software, but requires phones to be rooted [1]. We have then installed the CyanogenMod version of the Android operating system, which allows us to access the system primitives reserved to root users. We have tested out implementation over HTC Nexus One and Samsung Nexus S phones.

C. Machine Learning for Proximity Detection

The machine learning module consists of two parts: (i) a pre-processing module which is in charge of preparing the data generated by the data acquisition module for classification (ii) a classification module which determines, based on the sensed context, whether there are one, or more, nodes in close proximity.

1) Data-preprocessing: The data-preprocessing module receives from the Wi-Fi sensing module streams of raw data, which are then prepared for the machine learning module. Such process is summarized in Fig. 5.

The pre-processing consists of a *segmentation* phase, where raw data is aggregated over discrete time intervals, and the most relevant features for proximity detection are calculated. With respect to *RSSI* values, we grouped consecutive samples within 20 seconds and calculated signal characteristics for each group separately. This corresponds to approximatively 20 samples, although it may be less. It turned out that among the various tested signal characteristics (such as standard deviation, minimum and median), the combination of the mean and maximal value was proven empirically to provide the highest accuracy in distance estimation.

[1]Rooting a phone refers to the possibility of assuming superuser (root) privileges, gaining access to system files and being able to change files that normally are marked as read only.

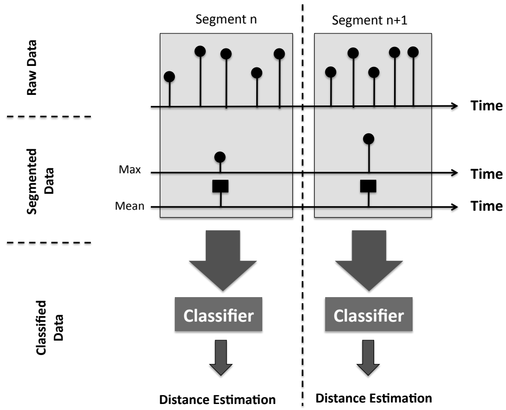

Figure 5. Machine learning process.

2) Contacts Classification: In order to infer the proximity of any two devices, we applied machine learning techniques and in particular a classification algorithm. The motivation for this approach lays in the instability and fluctuations of the Wi-Fi signal, typically due to environmental factors. From an initial study, we verified that a simple RSSI threshold analysis (assigning ranges of RSSI values to corresponding distances) did not suffice to obtain the required system accuracy. This led us to apply well-known machine learning algorithms to perform pattern matching between an unknown distance and the features extracted from the segmented RSSI signal.

Regarding the distances relevant for social interactions, we decided to map the observed RSSI samples to a pre-defined set of the distances (0.5m, 1m, 1.5m, ..., 8m). The reason being that we are not interested in estimating the precise distance between two mobile nodes, but rather to classify proximity ranges among users. As such, we are not interested in a regression model, but in a classification of the distance range in which the user will most likely be.

In order to properly design the classification algorithm, a training dataset was build taking RSSI measurements in three offices with dimensions of 12×8 m, 6×5 m and 6×3 m, a balcony of 12×2.5 m and a meeting room of 10×8 m. We used a pair of phones one in transmitting and the other one in receiving mode. Following a grid of 0.5m, RSSI was measured for 5 minutes on each distance between phones starting from 0.5m to the point in which either signal degraded to its minimal level or it was the furthest accessible point within room dimensions. The maximal distance in the experiments was between 5 and 8 m thus covering all the distances relevant to the detection of social interactions. The measurements were uploaded on a server and were used to evaluate different classifiers applying a cross-validation method.

The outcomes of this preliminary evaluation can be summarized as follows:

- a Naive Bayes with Kernel Density Estimation classifier proved to be the most accurate option. However, several classification techniques that we tested demonstrated similar performance in distance estimation. In terms of accuracy, it is possible to obtain a median estimation error (50th percentile) of approximately 0.5m.
- the classification accuracy depended on the models of phones that were used for building the training and test set. In particular, performance degraded if different phones were used to train and evaluate the classifier. This is due to the fact that RSSI patterns highly depend on the receiver characteristics [9], which are likely to be different across different phone models.

The second conclusion implies that, in order to obtain the desired distance estimation accuracy, it would be necessary to acquire a RSSI training set for any new phone model, i.e. it is unlikely to have one generic training set that would provide satisfactory accuracy across multiple phone models. However, acquiring training set which contains measurements from all the above mentioned points (0.5m, 1m ... 4.5m, 5m) for any new phone model intended to be used for the application, may be laborious and time consuming. To address this problem, we opted to measure RSSI only for a couple of minutes at a fixed distance and then to build the training set utilizing the following propagation model:

$$P(d) = P(d_0) - 10 \times n \times log\frac{d}{d_0} - X \qquad (1)$$

which describes the received signal strength $P(d)$ as a function of the signal power $P(d_0)$ at a reference distance d_0 from the transmitter phone and the distance d from the transmitter and the emitter. In the equation, n is the path loss exponent, X is a component which reflects the sum of losses induced by each wall between the transmitter and receiver. All values are expressed in *dBm*. This model, while being sensitive to reflections and multiple-path propagation, proved to be sufficiently accurate in the case of line of sight, which is the case when detecting proximity. We have found empirically [2] that the best suited value for the coefficient n is 1.5, while X is zero (there are no walls or other obstacles between points).

Starting from the above equation, we have artificially synthesized a training set using the following procedure, described also in Algorithm 1:

1) collect a set of RSSI measurements at a pre-defined distance (e.g., 1 m.);
2) create a synthetic training set by applying the propagation model to all the points collected in step 1 and

[2] For this, we have collected various RSSI samples for various distances, and used such training set to identify the parameters of the formula.

Algorithm 1 Synthetization of the training set in the case of a calibration performed at distance d_0, with n training samples.

for $i = 1 \to n$ **do**
  $training[i][d_0] = RSSI_i$
end for

for all d IN $\{1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$ **do**
  for $i = 1 \to n$ **do**
    $training[i][d] = (training[i][d_0] - 10 \times 1.5 \times log\frac{d}{d_0})$
  end for
end for

for all distances relevant to *Comm2Sense* application scenarios;

3) train the classifier starting from the generated synthetic training set.

The proposed algorithm will run over the smartphone and will require only an initial calibration by end-users in order to collect the RSSI measurements at a known distance. From this initial calibration, *Comm2Sense* will train a phone-specific classifier, which will be able to detect the proximity of any other PHS.

3) Preliminary Evaluation of the Classifier: In Fig. 6 we report a preliminary evaluation that we obtained by comparing the results using the classifier build according to the method described in Sec. II-C2 and the ground truth, obtained from the *RSSI* measurements in the five environments (three offices, balcony and a meeting room). As it can be observed, out of the tested models, Nexus One and HTC Desire provided the accuracy with the median error of 0.5m. On the other hand, Nexus S achieved lower accuracy regarding 50th percentile error 0.7m (Naive Bayes KDE) and 1m (GP) but performed better in terms of 80th percentile error 1m (Naive Bayes KDE). HTC Desire and Nexus One estimated distance with an error smaller than approx. 2m in 95% of cases.

III. Conclusions and Future Directions

In this paper, we have presented *Comm2Sense*, a middleware for mobile devices to detect proximity of people. The middleware has been developed over an Android platform, and does not rely on any external infrastructure or hardware, but smartphones. *Comm2Sense* exploits the ability of modern smartphones to act both as portable hot spots and Wi-Fi clients, and applies machine learning techniques to the analysis of RSSI signal in order to detect the proximity of phones.

Preliminary results demonstrate the ability of the system to detect proximity with a median error of 0.5m over a wide range of environments, both indoor and outdoor. Future

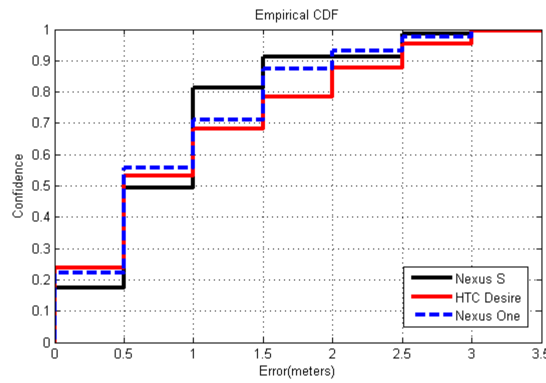

Figure 6. Distance classification accuracy using training set generated by applying the propagation model, in the case of various smartphone models.

directions regard the utilization of *Comm2Sense* in a real-life experimentation, to detect social interactions.

References

[1] D. Cuff, M. Hansen, and J. Kang, “Urban sensing: out of the woods,” *Commun. ACM*, vol. 51, no. 3, pp. 24–33, 2008.

[2] T. Choudhury and A. Pentland, “Sensing and modeling human networks using the sociometer,” *Wearable Computers, IEEE International Symposium*, vol. 0, p. 216, 2003.

[3] K. Fischbach, P. A. Gloor, and D. Schoder, “Analysis of informal communication networks - a case study,” *Business and Information Systems Engineering*, vol. 1, pp. 140–149, 2009.

[4] C. Cattuto, W. Van den Broeck, A. Barrat, V. Colizza, J.-F. Pinton, and A. Vespignani, “Dynamics of person-to-person interactions from distributed rfid sensor networks,” *PLoS ONE*, vol. 5, no. 7, p. e11596, 07 2010.

[5] I. Carreras, F. D. Pellegrini, D. Miorandi, D. Tacconi, and I. Chlamtac, “Why neighbourhood matters: Interests-driven opportunistic data diffusion schemes,” in *Proc. of CHANTS (MobiCom Workshop)*, San Francisco, US, 2008.

[6] L. Vu, Q. Do, and K. Nahrstedt, “Jyotish: A novel framework for constructing predictive model of people movement from joint wifi/bluetooth trace,” in *Proc. of PerCom*, march 2011, pp. 54 –62.

[7] J. Krumm and K. Hinckley, “The nearme wireless proximity server,” in *Proc. of UBICOMP*, Nottingham, England, 2004.

[8] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner, “Virtual compass: Relative positioning to sense mobile social interactions,” in *Proc. of Pervasive Computing*, Helsinki, Finland, 2010.

[9] P. Bhagwat, B. Raman, and D. Sanghi, “Turning 802.11 inside out,” *ACM SIGCOMM Computer Communication Review,*, vol. 34, no. 1, pp. 33–38, Jan. 2004.