Finding Dory in the Crowd: Detecting Social Interactions using Multi-Modal Mobile Sensing

Kleomenis Katevas
Imperial College London

Katrin Hänsel
Queen Mary University of London

Richard Clegg
Queen Mary University of London

Ilias Leontiadis
Samsung AI

Hamed Haddadi
Imperial College London

Laurissa Tokarchuk
Queen Mary University of London

ABSTRACT

Remembering our day-to-day social interactions is challenging even if you aren't a blue memory challenged fish. The ability to automatically detect and remember these types of interactions is not only beneficial for individuals interested in their behavior in crowded situations, but also of interest to those who analyze crowd behavior. Currently, detecting social interactions is often performed using ethnographic studies, computer vision techniques and manual annotation-based data analysis. However, mobile phones offer easier means for data collection that is easy to analyze and can preserve the user's privacy. In this work, we present a system for detecting stationary social interactions inside crowds, leveraging multi-modal mobile sensing data such as Bluetooth Smart (BLE), accelerometer and gyroscope. To inform the development of such system we conducted a study with 24 participants where we asked them to socialize with each other for 45 minutes. We built a machine learning system based on gradient-boosted trees that predicts both 1:1 and group interactions with a 30.2% performance increase compared to a proximity-based approach. By utilizing a community detection-based method, we further detected the various group formation that exist within the crowd. Using mobile phone sensors already carried by the majority of people in a crowd makes our approach particularly well suited to real-life analysis of crowd behavior and influence strategies.

ACM Reference Format:
Kleomenis Katevas, Katrin Hänsel, Richard Clegg, Ilias Leontiadis, Hamed Haddadi, and Laurissa Tokarchuk. 2019. Finding Dory in the Crowd: Detecting Social Interactions using Multi-Modal Mobile Sensing. In *SenSys-ML '19: The 1st Workshop on Machine Learning on Edge in Sensor Systems, November 10, 2019, New York, NY, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3362743.3362959

1 INTRODUCTION

The ability to automatically detect social interactions in unorchestrated scenarios is highly sought after in many areas including social and behavioral science, crowd management, and targeted advertising. This ability would facilitate a wide range of technologies, *e.g.*, crowd reconfiguration in evacuation management or networking analytics.

There have been many attempts for detecting social interactions automatically, primarily from video analysis. Most of the initial works use resource-hungry computer vision techniques [2, 7, 13]. Other approaches use custom-made wearable hardware that use sensors such as infrared light [6, 12, 21, 23], accelerometer [12], microphone [6, 12] and Bluetooth [12]. These works report reasonable accuracy but are expensive and problematic to scale in larger environments.

Smartphones and their wide range of embedded sensors enable researchers to explore social interactions in an automated way that depends entirely on the use of mobile sensing technology [16, 24, 28], without the need for additional wearable equipment or computer vision systems. Mobile sensing-based solutions are also easier and more cost efficient to deploy in unknown or new spaces as they only rely on the users' own hardware. Early systems that use mobile sensing report accurate results, but focus on detecting one-to-one social interactions [24] or rely on pre-trained models that only work with specific devices [24]. Others are restricted to controlled only environments that only cover a subset of natural-occurring formations [15], or use the phone's microphone for detecting body distances [28] — an approach that raises concerns about the user's privacy.

In this paper, we investigate an approach for detecting social interactions in a natural, non-artificial social setting that depends on sensors available in a modern smartphone (*i.e.*, Bluetooth Smart, Accelerometer and Gyroscope). We focus on the interactions that usually happen in social gatherings or networking events (*e.g.*, conferences, exhibition etc.) where people form standing interactions with two or more participants. We built a machine learning system based on gradient-boosted trees to detect both 1:1 and group interactions in a short granularity of 1 second windows. We then use a community detection algorithm based on graph theory to detect the various group formation that exist within the crowd. We evaluate our system in a case study with 24 participants interacting together for 45 minutes. Notice that due to software limitations, the phones were not able to broadcast a Bluetooth signal when the device is locked. Therefore, we ended up using coin-shaped beacons as a wearable device that simulates the smartphone's Bluetooth broadcasting function.

2 RELATED WORK

One of the first attempts to identify stationary, face-to-face interactions in an automated way is the *Sociometer* by Choudhury and Pentland [6], a wearable device that can be placed on each person's

SenSys-ML '19, November 10, 2019, New York, NY, USA

ACM ISBN 978-1-4503-7011-0/19/11…$15.00
https://doi.org/10.1145/3362743.3362959

shoulder and identify other people wearing the same device using Infrared (IR) sensors. In addition, it is equipped with an accelerometer sensor to capture motion and a microphone to capture speech information. Olguin *et al.* [23] developed a successor, called the *Sociometric* badge, that is smaller is size and includes Bluetooth, IR, microphone and accelerometer sensors. Huang *et al.* [12] designed a low-power wearable device capable of detecting human interactions using ultrasonic signal. They evaluated their device in a series of human experiments with both sitting and standing interactions. Montanari *et al.* [21] created a wearable device named *Protractor* that uses near-infrared light to monitor the user proximity and relative body-orientation. Even though the evaluation of this work is focused on the social behavior of an existing group that is interacting, Protractor could also be used to detect human interactions within crowds by using the estimated proximity and relative orientation between participants.

Opposed to deploying custom made sensors and badges, novel work focused on leveraging off-the-shelf devices and smartphone sensors. Palaghias *et al.* [24] presented a real-time system for recognizing social interactions using smartphones. They used the RSSI of Bluetooth Classic radios and a 2-layer machine learning model to detect the user's interactive zone and an improved version of *uDirect* research [11] that utilizes a combination of accelerometer and magnetometer sensors to estimate the user's facing direction with respect to the earth's coordinates. This work reported results of 81.40% accuracy for detecting social interactions, with no previous knowledge of the device's orientation inside the user's pocket. However, it has been evaluated in a limited dataset with eight participants while an observer was keeping notes that were later used as ground truth. Moreover, it is only capable of detecting one-to-one social interactions using a specific device model (HTC One S) and has not been evaluated in scenarios of interactions with dynamic sizes. Finally, it assumes that a Bluetooth connection is maintained between devices for continuously monitoring the RSSI, having an impact on the device's battery. Zhang *et al.* [28] developed a system that detects social interactions in the context of encountering with the use of audio sensing. They first used a combination of the smartphone's accelerometer, microphone and speaker, and with the use of inaudible acoustic signals they detected when two people approach and stop in front of each other. Next, they applied voice profiling on the audio recordings to confirm if the pair is engaged into an actual conversation. They evaluated their approach in a real-world use case with 11 participants for 1 hour using self-reported questionnaires at the end of the study as ground truth. The evaluation of this work that reports 6.9% false positives and 9.7% false negatives, was conducted over the complete case study (*i.e.*, who met with whom during the event) instead of a more fine-grained evaluation over shorter windows. Thus, it is not capable of capturing information such as the duration of an interaction, or more advanced crowd dynamics such as type of group formations that were conducted over time. Moreover, such approach requires a continuous audio recording from each user's smartphone, a process that raises ethical and privacy concerns when using it in real-world scenarios. Katevas *et al.* [16] presented a simplistic proximity-based approach for detecting stationary interactions in planned events, using the interpersonal proximity estimated by the device's Bluetooth Smart sensor. They evaluated the social interactions that took place in a controlled environment with six participants for four minutes, reporting a performance of 90.9% precision and 92.4% recall. This work was evaluated in a limited dataset (approx. 5 minutes long) with artificially created interactions instructed by the designer of the study. Moreover, the proximity-based algorithm they used is similar to the baseline used in this work.

3 EXPERIMENTAL SETUP

In order to identify and evaluate the sensors needed for detecting stationary interactions in a natural setting, data was collected from participants during a social networking event.

37 potential participants were recruited via email and flyers.; 24 of those took part in the actual study of which 9 were male and 15 female. Participants were selected based on mobile phone model (iPhone 4 or higher) and operating system version (iOS 7 or higher) and availability of the iBeacon sensor. Two devices experienced errors during the study (*i.e.*, Bluetooth Smart sensor reported an internal error and did not collect data) and were excluded from the data analysis, resulting into 22 valid participants.

3.1 Procedure

Participants installed a sensor data collection app, based on SensingKit for iOS v0.5 continuous sensing framework [15]. The app automated the sensor calibration, participant registration and data collection. Participants were invited to an indoor location space, 6.57×5.36 meters, with $3.90m$ height; a natural space that is often used for social events and performances. Two HD cameras were fixated at a DMX lightning rig ($3.27m$ height) to record video (but not audio). These videos were annotated to provide the ground truth for social interaction (see Section 4.1). Before the study began, participants were asked to read the information sheet and sign the consent form. Participants were equipped with a Radius Networks RadBeacon Dot[1] each (coin shaped Bluetooth 4 -based low energy beacons), to place in one of their pockets. All coin-shaped beacons were pre-configured to $10ms$ advertising interval (highest) and $-18dBm$ broadcasting power. Half of the participants were instructed to place the beacons in the left pocket and the other half in the right pocket. The phone was always placed in the other pocket to avoid interference. During the setup process, participants were guided through the mobile app configuration. This process included a facial photograph for ground truth identification and demographics (age, gender, weight, and height). Finally, participants were asked to collectively perform a gesture with the phone in front of of the cameras. The hereby recorded sensor data of each participant was later synced with the $25fps$ video feed, achieving a sync accuracy of ±40ms.

Participants were then instructed to socially network for a total of 45 minutes. The discussion topic was intentionally left open, trying to simulate a realistic networking scenario. After the session, participants returned the beacons, submitted the collected data and were reimbursed with £20 for their time. In total, 99 one-to-one interactions were observed with a mean duration of $254.9sec$ (±161.7) and 22 group interactions (*i.e.*, interactions that include more than

[1] https://radiusnetworks.com

two participants) with a mean duration of 117.2*sec* (±139.4). A separate interaction begins when the members of a group change. If the group configuration consisted less than 5*sec*, it is not counted.

3.2 Sensor Data Set

The dataset collected for each participant contains the following sensor data:

- **iBeacon Proximity:** The RSSI from the mobile device with all beacons in range.
- **Linear Acceleration:** The device measured acceleration changes in three-dimensional space. This excludes the 1g acceleration produced by gravity.
- **Gravity:** The orientation of the device relative to the ground, by measuring the 1g acceleration produced by gravity.
- **Rotation Rate:** The device's rate of rotation around each of the three spatial axes.

The sampling rate was set to the maximum supported (100Hz) for all motion and orientation sensors. iBeacon Proximity sensor has a fixed (non-customizable) sample rate of 1Hz.

4 DETECTING SOCIAL INTERACTIONS

4.1 Ground Truth

Video recorded from two different angles was annotated by two independent annotators using ELAN multimedia annotator software [27]. As the aim of the study is to detect stationary interaction only, the annotators logged the beginning and end of each stationary interaction for each participant separately. The annotations were cross-validated afterwards and finally verified by a third person. The instructions that the annotators followed were based on Kendon's F-formation system [17]:

> *An interaction begins at the moment two or more stationary people cooperate together to maintain a space between them to which they all have direct and exclusive access.*

4.2 Target Variable

The dataset has a total of 645,895 labels for each combination of the 22 valid participants interacting. The target variable is binary, with the following two classes: {1} when a pair of participants is interacting together, and {0} when they are not. That resulted into 38,332 labels in class 1 (6.31%), and 607,563 labels in class 0 (93.69%). The dataset is naturally imbalanced since it includes one label for all combinations of the participants interacting with each-other per second. The level of this imbalance depends on the number of people interacting, but also on the type of interaction (*e.g.*, one-to-one, groups of three etc.).

4.3 Sensor Data Pre-processing

The data and video feed were synchronized based on the synchronous wave-gesture participants performed in front of the cameras. Each device was recording sensor data using the internal CPU time base register as timestamp, so pre-alignment between different types of sensor data (*e.g.*, accelerometer with iBeacon Proximity) was not required. For all iBeacon Proximity data, all data reporting *Unknown* values (where RSSI is −1) were excluded. This usually occurs at the beginning of iBeacon ranging process due to insufficient measurements to determine the state of the other device [1], or for a few seconds after the device gets out of the beacon's broadcasting range. All measurements from each user's own beacon (*i.e.*, from a participant's phone to their beacon) were also excluded.

The signal from all motion data was re-sampled and interpolated to 100Hz. Finally, the magnitude (resultant vector) was computed from the three axes to counteract different physical phone alignments in the participants' pockets. The iBeacon sensor was the only sensor that reported missing values. Since most machine learning algorithms do not accept features with unknown values, a data imputation process was required. Thus, missing values (corresponding to 71,88% of the collected beacon data) were replaced with the maximum available distance; the reason was mostly sensors being out of broadcaster range and meaning interaction is not feasible.

4.4 Proximity Estimation

The Path Loss Model (PLM) was applied in order to estimate the proximity (d) between each device and all beacons in range using the RSSI ($P(d)$), as shown in the following formula:

$$d = 10^{\frac{P(d_0)-P(d)-X}{10\times n}}, \tag{1}$$

where $P(d_0)$ is the Measured Power (in dBm) at 1-meter distance, n the path loss exponent, d the distance in which the RSSI is estimated and X a component that describes the path loss by possible obstacles between the transmitter and the receiver. The value $n = 1.5$ was set as a default constant for indoor environments [19]. The value $X = 0$ was also chosen as it was required to measure a direct contact where no obstacles (*e.g.*, other participants) between the two devices exist. In the situation that another participant exists in between, PLM would report a longer distance due to the decreased RSSI, and consequently, the accuracy of the distance estimation will decrease. This is a desired effect as it is only wanted to cluster whether the two users are within a range that a social interaction can be achieved. According to Hall [10], personal social interactions are achievable between 0.5 and 1.5 meters distance.

4.5 Normalized Proximity

The *Normalized Proximity* (NP) is suggested by this work as an easy to compute approach for detecting social interactions using proximity-based information. More specifically, the distance of two participants is used (computed using the Path Loss Model discussed in Section 4.4) with all unknown values (*i.e.*, when the pair is out of beacon range) being replaced with the max of all distance estimations. A proximity value x is normalized into the range $[0, 1]$ as follows:

$$\hat{y} = \frac{x_{max} - x}{x_{max} - x_{min}}, \tag{2}$$

where $\hat{y}$ is an estimate as to whether the pair is interacting, and x is the estimated proximity between the pair and the x_{min} and x_{max} are the minimum and maximum values of x for all pairs in the data set. Because $\hat{y}$ is in the range $[0, 1]$ it can be compared to probability estimates.

4.6 Feature Engineering

A series of common features were computed for all $C(22, 2) = 231$ combinations of the participant pairs. Features reflecting the current moment were initially computed, in a static window of 1 sec, following with features reflecting past information. A set of features that are commonly included in mobile sensing problems were used, *e.g.*, features extracted from motion and orientation sensors. To compute these features, the magnitude (resultant vector) of the 3-axis data were used in order to account for different physical alignment of each device within the users' pockets. Thus, no alignment of each user's motion and orientation sensors was pre-required.

Interpersonal Space Features. iBeacon Proximity sensor data of a pair includes two measurements: Let $rssi_{ij}$ be the RSSI between the two participants as measured from the device of user i and $rssi_{ji}$ be the RSSI from the same distance as measured from the device of user j. The mean of the two measurements was computed as an indication of how close the two participants are in space:

$$f_{prox\_rssi\_mean} = (rssi_{ij} + rssi_{ji})/2 \tag{3}$$

In addition, a feature that represents the absolute difference between the two measurements was computed:

$$f_{prox\_rssi\_diff} = |rssi_{ij} - rssi_{ji}| \tag{4}$$

Note that in this case, the raw RSSI was used as the same hardware was used for broadcasting a beacon signal across all participants, and thus, a *Measured Power* constant is not required. In the case of multiple devices being used, then a feature that estimates the interpersonal distance based on a calibrated *Measured Power* constant would be required, using the PLM equation mentioned in Section 4.4.

Device Position Features. Information about the device position is also important as body orientation is expected to influence the RSSI signal between the two devices. For that reason, four features have been developed that includes the information of the device position (left vs. right per participant) using one-hot encoding.

Motion and Orientation Features. By using the measurements of the linear acceleration sensor, a feature that indicates the time since the participant has moved (in seconds) was added. A threshold of $0.15g$ was empirically chosen, indicating whether a user is moving or not, and computed the absolute difference between the pair. It is expected that if two users are moving, they will stop at the same moment and engage into a conversation, and thus, the value of that feature will be close to zero. When both users had the status 'in motion', the feature was set to NaN (Unknown).

For all motion sensor data (*i.e.*, linear acceleration, gravity, rotation rate), a cross correlation function was applied on an overlapping window of 10 seconds and extracted the maximum correlation, as well as the distance (in seconds) from the max correlation, as an indication of how similar a pair is behaving on those windows. The 10 seconds constant was chosen as indicated by [19], but further investigation in the range of 2 to 60 also verified it as the optimal constant.

Past Information Features. In order to take advantage of past information available in the data set, the *min*, *max*, *mean* and *std* was computed on all time-series features (*i.e.*, excluding the one-hot encoded device positioning features), in an overlapping window of 10 seconds.

4.7 Evaluation Procedure

For evaluating the performance of the model, a standard 10-fold cross-validation schema was used. The dataset was initially split over time, however, due to the time-series nature of our study, a significant overfitting was reported. More particularly, since participants were changing their interactive state at any given moment, the model was memorizing the features per split and inferring them back with very high performance, due to information leakage. Thus, the data was split per participant combination (*i.e.*, 23 samples out of 231 due to the 10-fold schema) rather than over time.

In the context of this work, *Precision* is: from the detected interactions, how many of them did the model detect correctly, whereas *Recall* is: from all interactions taking place, how many of them did the model detect. Depending on the use case, applications can emphasize one measure over the other. The evaluation metrics that will be used in the rest of this report is *Precision-Recall (PR)* curve. Although ROC curves are heavily used when reporting performance in classification problems, due to the nature of our dataset being unbalanced, PR plots as suggested for this case by [26] and [8] were used.

4.8 Model Choice

As a learning model we use XGBoost [5]. XGBoost is a state-of-the-art gradient boosting regression tree algorithm that has emerged as one of the most successful feature-based learning models in recent machine learning competitions. We empirically found XGBoost consistently outperformed other well-established classifiers, such as Logistic Regression [22], Support Vector Machines [9], or Random Forests [4]. We used XGBoost v0.7.2.1 as part of the Python library scikit-learn [25] v0.19.1 and its XGBoost Python wrapper.

A parameter tuning was performed on a 20% subset of the dataset (*i.e.*, 46 samples out of 231). This subset was only used for the model tuning task and was never used in the training/validation procedure. The aim was to discover the model's configuration that maximizes the Average Precision (AP) performance. More specifically, a grid search algorithm over all possible combinations of the most influential parameters was followed. The configuration with the best performance of AP 80.4% (*i.e.*, performance using the 20% subset) had the parameters `max_depth=4`, `colsample_bytree=0.2`, `subsample=0.5` and `learning_rate=0.05`. This configuration is used in the rest of this section for training and validating the model with the remaining 80% of the data set.

4.9 Detecting Group Formations

Detecting communities is important for a variety of applications including mobile social networks, recommender systems, security, and crowd management. One of our objectives is to automatically detect such group formations and classify the formed communities. Our concept for detecting group formation is based on graph theory. Each moment (in seconds) is represented as an undirected weighted graph $G = (V, E, w)$, with a set of vertices V and weighted edges $E(w)$. Each vertex corresponds to a participant, each weighted edge corresponds to the probability of a pair interacting, as detected

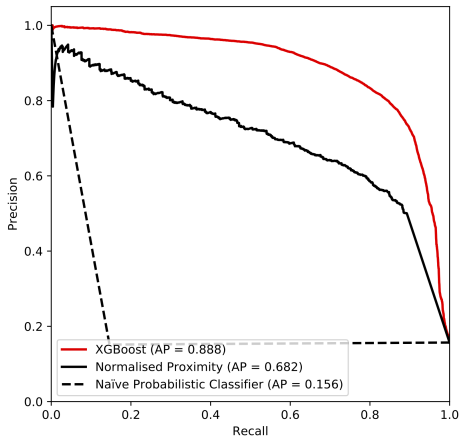

Figure 1: Performance of XGBoost classifier using a Precision-Recall (PR) Curve. The figure also includes the performance of the Naïve Probabilistic Classifier (NPC) and the Normalized Proximity (NP) for easy comparison.

using the XGBoost classifier, and each detected community C corresponds to a group formation.

We use a modularity optimization approach [3] that is fast to compute even in large networks and relies on the time-based stability of the network conditions at short time intervals [18], also known as *resolution* constant. Initially, every vertex V_i is assigned to a community C_j. Each vertex is then evaluated separately to join its neighbor's community. The join that achieves the maximum positive gain in modularity is the one that is committed. If no positive modularity is achieved, the vertex remains in its initial community. This process is applied to all vertices sequentially until it converges. Next, a new network is created using the communities as vertices (C), one edge between the connected communities with $C(w)$ the sum of all $E(w)$ that belong to that community, and a self-loop edge for the internal vertices. The algorithm is repeated until a maximum modularity is achieved.

We applied the community detection algorithm per second using NetworkX[2] v2.1 to handle graph operations on the network and considered a group formation when a community exists within the graph. We evaluate the performance of our approach in three ways: (a) *link-level*, where a link represents an interaction between a pair of participants, (b) *node-level*, where a node represents a participant that belongs to the correct interactive group, and (c) *group-level*, where a group is detected to include the correct participants.

5 RESULTS AND DISCUSSION

Our results provide evidence that it is possible to detect interactive groups of various sizes relying on data collected from mobile devices, with a reasonable performance. That is a *link-level* detection performance of 88.8% AP (*i.e.*, 30.2% increase from NP and 469.2% increase from a Naïve Probabilistic Classifier baseline). Figure 1 shows the performance using a Precision-Recall (PR) curve plot.

[2]https://networkx.github.io

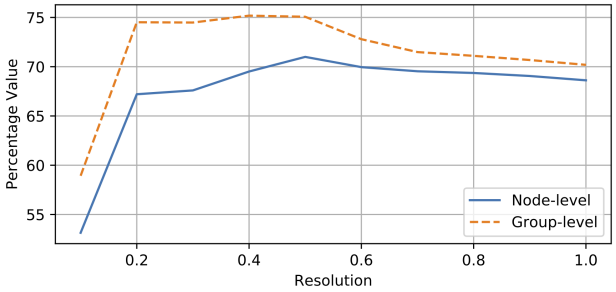

Figure 2: Performance of group detection at node- and group-level, using different resolution constants.

Moreover, our work evaluates the interactions in high granularity of 1 second windows. This is an improvement compared to other related works that binary detect if a pair has interacted during the event [24, 28], or use longer windows of a few seconds [19, 20].

Figure 2 shows the performance of the group detection as described in Section 4.9. It displays the group detection accuracy on node- and group-level, using different community detection resolution constants within the range of 0.1 and 1.0. The optimal resolution constant in this case, shown in Figure 2, is 0.5, achieving a node-level performance of 71.1%, and group-level performance of 75.2%. Applying the same method on the NP baseline with the optimal resolution of 0.2 gives node-level performance of 48.7%, and group-level performance at 50.9%.

The dataset that has been analyzed, even though extended compared to other similar studies [16, 24], only represents a subset of what is expected in similar social gatherings, such as conferences or other networking events. Other social interactions have not been investigated such as people interacting in a coffee table, walking interactions etc. In addition, the device position that has been tested is the trousers pocket which is a popular position according to [14]. However, other positions should also be considered, such as the shoulder bags, backpacks, or even holding the device at hand.

6 CONCLUSIONS AND FUTURE WORK

In this work, we introduced a supervised machine learning approach capable of detecting stationary social interactions of a variety of sizes inside crowds. Our work does this in a relatively large (as compared to other related works) study, achieving a performance of 88.8% AP when evaluating the interactions of the participants on link-level. Our approach is capable of detecting group formations at a node-level performance of 71.1%, and group-level performance of 75.2%.

We believe that our work will be particular useful to researchers and practitioners wishing to explore crowd dynamics in social gatherings, event organizers aiming to monetize their events by providing rich analytics about their attendees, or event attendees wishing to remember their contacts without the need for exchanging business card or social media details. In future work, we aim to apply a real-time version of this work in a large-scale social event and explore the ways in which the crowd is interacting in planned events.

REFERENCES

[1] Apple Inc. Getting Started with iBeacon v1.0. https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf, 2014. [Online; accessed 07-April-2018].

[2] L. Bazzani, M. Cristani, and V. Murino. Decentralized particle filter for joint individual-group tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1886–1893, June 2012.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[4] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.

[5] T. Chen and C. Guestrin. XGBoost: a scalable tree boosting system. In *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 785–794, 2016.

[6] T. Choudhury and A. Pentland. Sensing and modeling human networks using the sociometer. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, ISWC '03, pages 216–, Washington, DC, USA, 2003. IEEE Computer Society.

[7] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. Del Bue, G. Menegaz, and V. Murino. Social interaction discovery by statistical analysis of f-formations. In *Proceedings of the British Machine Vision Conference*, pages 23.1–23.12. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.23.

[8] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[9] N. Deng, Y. Tian, and C. Zhang. *Support Vector Machines: Optimization Based Theory, Algorithms, and Extensions*. Chapman & Hall/CRC, 1st edition, 2012.

[10] E. T. Hall. *The hidden dimension*. Doubleday & Co, 1966.

[11] S. A. Hoseinitabatabaei, A. Gluhak, and R. Tafazolli. udirect: A novel approach for pervasive observation of user direction with mobile phones. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 74–83, March 2011.

[12] W. Huang, Y.-S. Kuo, P. Pannuto, and P. Dutta. Opo: A wearable sensor for capturing high-fidelity face-to-face interactions. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys '14, pages 61–75, New York, NY, USA, 2014. ACM.

[13] H. Hung and B. Kröse. Detecting f-formations as dominant sets. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, ICMI '11, pages 231–238, New York, NY, USA, 2011. ACM.

[14] F. Ichikawa, J. Chipchase, and R. Grignani. Where's the phone? a study of mobile phone location in public spaces. In *2005 2nd Asia Pacific Conference on Mobile Technology, Applications and Systems*, pages 1–8, Nov 2005.

[15] K. Katevas, H. Haddadi, and L. Tokarchuk. SensingKit: Evaluating the sensor power consumption in ios devices. In *Intelligent Environments (IE), 2016 12th International Conference on*, pages 222–225. IEEE, 2016.

[16] K. Katevas, H. Haddadi, L. Tokarchuk, and R. G. Clegg. Detecting group formations using iBeacon technology. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, pages 742–752, New York, NY, USA, 2016. ACM.

[17] A. Kendon. *Conducting interaction: Patterns of behavior in focused encounters*, volume 7. CUP Archive, 1990.

[18] R. Lambiotte, J.-C. Delvenne, and M. Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*, 1, 12 2008.

[19] A. Matic, V. Osmani, A. Maxhuni, and O. Mayora. Multi-modal mobile sensing of social interactions. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, pages 105–114, May 2012.

[20] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer. A study of bluetooth low energy performance for human proximity detection in the workplace. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 90–99, March 2017.

[21] A. Montanari, Z. Tian, E. Francu, B. Lucas, B. Jones, X. Zhou, and C. Mascolo. Measuring interaction proxemics with wearable light tags. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):25, 2018.

[22] J. A. Nelder and R. J. Baker. *Generalized linear models*. Wiley Online Library, 1972.

[23] D. O. Olguín and A. S. Pentland. Social sensors for automatic data collection. *AMCIS 2008 Proceedings*, page 171, 2008.

[24] N. Palaghias, S. A. Hoseinitabatabaei, M. Nati, A. Gluhak, and K. Moessner. Accurate detection of real-world social interactions with smartphones. In *Communications (ICC), 2015 IEEE International Conference on*, pages 579–585, June 2015.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[26] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[27] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. Elan: a professional framework for multimodality research. In *Proceedings of LREC*, volume 2006, page 5th, 2006.

[28] H. Zhang, W. Du, P. Zhou, M. Li, and P. Mohapatra. Dopenc: Acoustic-based encounter profiling using smartphones. In *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking*, MobiCom '16, pages 294–307, New York, NY, USA, 2016. ACM.