Activity Recognition Using Graphical Features from Smart Phone Sensor

Syeda S. Akter^(✉) , Lawrence B. Holder^(✉) , and Diane J. Cook^(✉)

Washington State University, Pullman, WA 99164, USA
{selina.akter,holder}@wsu.edu, cook@eecs.wsu.edu

Abstract. We develop a graphical feature-based framework that collects data from different kinds of sensor networks, represents the sensor network data as a graph, extracts graphical features from the graph representation, and adds those features to a set of non-graphical features that are typical for the application. Our hypothesis is that the addition of a structural representation and transitional features will improve performance for the corresponding prediction tasks of different networks. We apply our graphical feature-based approach on smart phone GPS sensor data to predict activities performed by phone users. We represent the location category corresponding to each GPS value as a node and movement of users from one GPS location to another as an edge in graph. Then we extract graphical features such as existence of nodes and edges from the graph representation and add them to basic sensor data features coming from the smart phone. We find that using this augmented feature set improves activity recognition accuracy by 7.27% compared to using only basic non-graphical features with feature set augmented with existence of nodes performing the best.

Keywords: Graph mining · Sensor networks · Smart phone sensors · GPS
Graphical features · Activity recognition

1 Introduction

We envision a graphical feature-based framework to represent and analyze smart phone data. This framework collects data from sensor networks, uses graph structure to represent movement-related data, and employs selected graphical features to improve corresponding prediction tasks of those sensor networks. In our previous work [1], we designed an algorithm to perform activity recognition from smart home motion sensor data in which we represented the motion sensor data as a graph, extracted graphical features from it, and classified activities performed by residents. The approach achieved significant improvement in classification accuracy compared to the benchmarks in the area. Next [2], we applied the graphical feature based framework on Nokia Smart Phone sensor data, represented GPS information as a graph, extracted and selected useful graphical features, and trained a support vector machine to perform classification for the target variables of gender, age-group and job type. Our approach outperformed most of the benchmarks in the field of demographic prediction from sensor data while using only one type of generic sensor data through leveraging graph structure and movement

patterns. As part of our goal of evaluating the use of graph representations and graph mining to improve performance on recognition and prediction tasks for sensor networks, in this paper we represent smart phone sensor data as a graph to enhance the task of activity recognition.

Activity recognition from sensor data is an active area of research. Automatically detecting human activities from a range of sensors can have broad applications such as remote patient monitoring and medical diagnosis to shorten hospital stay, child and elderly care, emergency assistance both at home and at assisted living, reminder system for people with cognitive disorder and chronic conditions, and recognition of sports and leisure activities in order to increase the lifestyle quality of people. Smart phones are becoming a ubiquitous part of our daily and social life. Incorporation of diverse and powerful sensors such as GPS, gyroscope, accelerometer, light sensors, temperature sensors, and Bluetooth make it a useful tool for activity recognition from smart phone sensor data. Added benefits include unobtrusiveness, low installation cost, ease of use, and ability to monitor inside and outside the home [4].

We hypothesize that representing smart phone sensor information as a graph and adding transitional features to basic non-graphical sensor data features will improve activity recognition performance. We used the dataset collected through an activity learning mobile app called Activity Learner (AL) designed by Aminikhanghahi et al. [3] We chose GPS information to represent as a graph, extract graphical features, and use these graphical features along with typical features based on non-graphical sensor data such as accelerometer and gyroscope to predict activities. The experiment shows that inclusion of graphical features significantly improves performance over typical non-graphical features with nodes features providing the best results. Analyzing the confusion matrix shows that the addition of edges may improve the performance for some activities. Using only selected features has the potential to improve the performance with the addition of edges.

2 Previous Works

The first and second age of the internet connected people to the internet. The third age of the internet connects not only people but also things to the internet [13]. According to CISCO, 50 billion things will be connected to the internet in 2020 [14]. These things range from very small and static devices (e.g., RFIDs) to large and mobile devices (e.g., vehicles). According to Khalil et al. [14], the role of Wireless Sensor Networks (WSN) in the IoT is that of a virtual skin that connects the things with the network and with each other, makes them aware of their surroundings, and shares this information with other things in order to make informed decisions. In [15], Zeng and Min approached how to build such a complicated system and presented a systematic design framework for IoT Enabled Systems.

A smartphone is light, inexpensive, user-friendly, multipurpose, and portable device that can be easily used by people in their daily lives and includes all technologies needed for IoT [16]. Smartphones are equipped with a range of built-in sensors such as accelerometers, motion sensors, position sensors, environmental sensors, microphone, and

camera providing images and videos. Special sensors measuring health vital signs, such as body temperature, ECG value, blood glucose level, stress level, body fat percentage, heart rate, etc., can be integrated into the smartphone. In [16], Khaddar et al. termed smart phone as the brain of the IoT world as smart phones come with a variety of connectivity technologies such as NFC, Bluetooth, Wi-Fi, and cellular allowing it to connect and interact with other devices and sensors. Next we discuss related works that used accelerometer, GPS and other sensor information from smart phones for the task of activity recognition.

Bouchard et al. [5] discussed different types of spatial features such as distance, position, shape and gesture. Their experiments with a user's raw latitude and longitude as features showed improved accuracy for activity recognition. Aminikhanghahi et al. [3] developed an algorithmic approach called Thyme for adapting prompt timing based on the context of the user's activity. In the first phase of Thyme, an Activity Learner (AL) smart phone app collects smart phone sensor data and learns a mapping from raw sensor data to activity labels through the use of classification algorithms Linear Support Vector Classification (SVC), Naïve Bayes (NB), K-Nearest Neighbour (KNN), Decision Tree (DT) and Random Forest (RF) with Random Forest resulting in the highest 82% accuracy with leave-one-out validation. This result is for training and testing done for each user separately; for combined users the average accuracy is 78%. Along with extracting standard signal processing features from raw sensor data, AL also computes higher-level information about the five-second data window, including heading change rate (percentage of points in the sequence that change direction), stop rate (percentage of points in the sequence that exhibit a significant drop in velocity), overall trajectory from start to finish of the data sequence, and normalized distance to the user's mean location. They also have used a cost-sensitive classifier by adding weight to each data point during training to help the learning algorithm handle the imbalanced class problem. Our method shows that use of generic graphical features can improve prediction accuracy without use of well thought-out, application-specific features or adding special methodologies to handle an imbalanced class distribution.

Liao et al. [8] approached location-based activity recognition from GPS traces. They train a conditional random field to iteratively re-estimate significant places and activity labels. Initial activity estimation consists of a sequence of locations and the most likely activity performed at that location, and these estimates are inferred by applying belief propagation. A set of significant places are extracted from this activity estimate and then used to classify individual activities again based on whether they belong to a significant place. This process is repeated until the activity sequence does not change. Their proposed method is evaluated on a fairly small dataset for four participants. The data does contain complete GPS traces for seven days of data per person, but the method is prediction task dependent. The method provided good results on this particular dataset, but it still needs to be explored whether this method can be applied in general to different sensor networks and different prediction tasks.

Chetty et al. [6] and Garcia-Ceja et al. [7] both performed activity recognition from three-dimensional accelerometer data and showed improved accuracy in predicting simple [6] and complex activities [7]. We would like to show that using a graph representation of spatial features and extracting transitional features from the representation

improve the activity recognition accuracy even further over these non-graphical accelerometer-based features.

3   Graphical Feature Based Framework

We collect location-related sensor information, namely latitude and longitude from smart phone sensor data in order to predict activities performed by smart phone users. We converted this geolocation information to location categories. We used OpenStreetMap (OSM), which is a map of the world built by a community of mappers that contribute and maintain data about roads, trails, cafes, railway stations and much more. [11] It is time consuming to access existing world maps that are available only online to probe about categories of each geo-location. To address this issue, we used a tool called Nominatim [12] through which we can download OSM data, import to a local database, and perform reverse geo-coding using another tool, geopy [9] for large amounts of geo-location data locally in significantly less time.

Whenever any user visits a place, we represent that location category as a node in the graph. When a user moves from one location to another, we add an undirected edge between the corresponding two nodes in the graph. From this graph representation, we extract graphical features that we present in Table 1. In Table 1, we also show some basic features that we can directly obtain from smart phone sensors as a typical set of features. We add selected graphical features to this basic feature set and feed this augmented feature set to a classifier for predicting activities. The workflow for activity classification from GPS data based on the Graphical Feature Based Framework is shown in Fig. 1.

Table 1. List of features

| Types of features	List of features	Raw sensor data that are used
Basic features (statistics applied to raw sensor data)	Max, min, sum, mean, median, standard deviation, median absolute deviation, zero crossings, mean crossings, interquartile range, coefficient of variation, skewness, kurtosis, simple moving average (SMA), log SMA, power, autocorrelation	Accelerometer data across x, y, z-axis, rotation across x, y, z-axis, yaw, pitch, roll, month, day of week, hour, minute and seconds
Graphical features (generated from graph representations)	Existence of nodes, existence of edges, and existence of both nodes and edges	Latitude and longitude
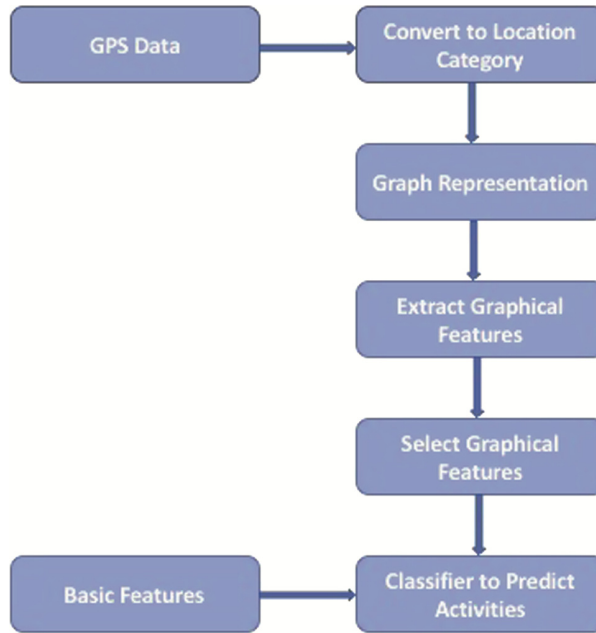
Fig. 1. Graphical feature-based framework for activity prediction

4 Computational Details and Results

4.1 Dataset

Aminikhanghahi et al. [3] designed a mobile app that collects 5 s of data at intervals specified by the user on iOS and Android platforms. The app, called Activity Learner (AL), collects the following 16 types of raw sensor data: accelerometer data across x, y and z-axis, rotation across x, y and z-axis, yaw, pitch, roll, course, speed, horizontal accuracy, vertical accuracy, latitude, longitude and altitude along with time and date information (month, day of week, hour, minute and seconds). The total number of instances in this dataset is 17933. We convert the GPS sensor data to location category and represent each location category as a node in the graph representation. Some example location categories are library, parking, motel, cycle way, hotel, park, supermarket, place of worship, school, bar, restaurant, bus stop, and road.

We construct one undirected graph for each activity performed by users in this dataset. The app AL guesses and periodically queries the participant about their current activity to obtain the labels of these activities. The user can agree to the predicted activity through the AL interface. Alternatively, they can proactively provide input about the activity they are currently performing. [3] In the dataset we analyze, there are a total of 214 unique activities performed by 47 participants. In many cases they labeled the same activities with different names or spellings. We map similar activities with slightly different names or with different spellings into a general name. For example, mapping
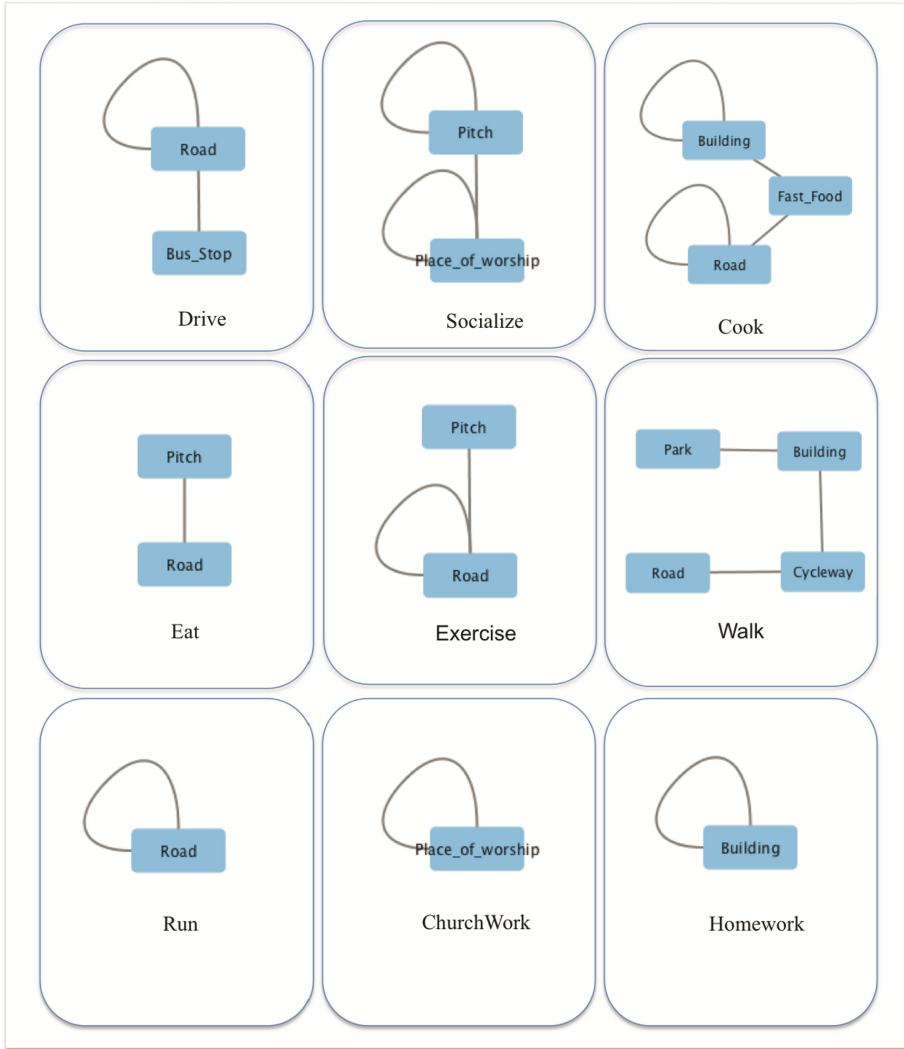
Fig. 2. Graph representations of different activities

both 'Driving' and 'Drive' to one activity 'Drive', mapping 'Errands', 'RunErrands', 'Store', 'Walmart' to 'Errands', etc. Al comes with an option to provide this activity mapping file. After activity mapping, there are 116 unique activities in the dataset. In Fig. 2, we show example graph representations for instances of different activities out of these 116 activities such as 'Drive', 'Socialize', 'Cook', 'Eat', 'Exercise', 'Walk', 'Run', 'ChurchWork' and 'HomeWork'. After representing each activity as a graph, the total number of unique nodes is 42 and the total number of unique edges including self-loops is 94. We use this set of unique nodes and edges as our graphical feature set and add it to the basic feature set for each instance. For each activity, we construct a graph

and when a node exists in this user-activity graph, we mark that corresponding feature as 1, otherwise we mark it as 0. Similarly, for each edge in this user-activity graph, we mark the corresponding feature in our feature set as 1, otherwise it is marked as 0. In this way we construct and add the graphical feature set for each instance to the basic feature set. We tried Decision Tree, Random Forest, Gradient Boost, Extra Tree, Bagging and Ada Boost for classifying with Basic Features. Extra Tree Classifier performed the best among these six classifiers. Initially, we select the 100 best features for inclusion in the model. To classify activities, we apply the ExtraTreeClassifier using the SelectKBestFeatures feature selection method with Mutual Information as the scoring function for the features.

4.2   Result

In Table 2, we compare performance of graphical features with basic features for classifying activities. As demonstrated in Table 1, basic features include basic statistical computation (max, min, sum, mean, median, standard deviation, median absolute deviation, zero crossings, mean crossings, interquartile range, coefficient of variation, skewness, kurtosis, simple moving average (SMA), log SMA, power, autocorrelation) of accelerometer data across x, y, z-axis, rotation across x, y, z-axis, yaw, pitch, roll, and date information (month, day of week, hour, minute and seconds). Graphical Features include existence of nodes, existence of edges, and existence of both nodes and edges.

Table 2.   Accuracy in percentage for basic features vs graphical features for six classifiers

Features	Basic	Basic + Nodes	Basic + Edges	Basic + Nodes + Edges
Decision tree	45.27	46.25	**46.64**	46.18
Random forest	50.39	**50.96**	49.56	49.92
Extra tree	51.45	**55.19**	53.86	53.73
Gradient boost	33.91	**37.17**	37.05	36.69
Bagging	50.93	51.90	52.19	**52.31**
Ada boost	**16.53**	**16.53**	16.13	16.13

We apply six different classifiers, namely, Decision Tree, Random Forest, Extra Tree, Gradient Boost, Bagging and Ada Boost with 3-fold cross-validation to classify 116 activities performed by 47 participants. The best performing feature set along a row (for each classifier) is boldfaced. We observe that adding only nodes or only edges or both nodes and edges improve the result compared to only basic features except Ada Boost. Decision Tree provides the best result with combination of basic features augmented with edges. For Bagging, combination of basic, nodes and edges as feature sets is the best performer. Random Forest and Extra Tree produce the best result with basic and nodes feature sets. However, the best result among all classifiers and feature sets is produced by Extra tree with basic feature set augmented with nodes which is 7.27% improvement over using only basic features.

	Activities that contain Edges	Total Error Nodes	Total Error Edges	Total Error Nodes+ Edges	Number of Instances that contain Edges	Total Number of Instances
1						
2	Eat	305	307	282	28	1070
3	Drive	318	297	310	70	819
4	Read	76	81	70	23	408
5	Bath	73	75	68	5	295
6	Sleep	179	175	192	12	745
7	Internet	103	99	104	47	315
8	Play	93	92	90	7	266
9	Bus	16	14	14	3	69
10	Sports	11	11	9	3	53
11	Shop	42	41	40	5	150
12	Phone	38	37	36	15	118
13	Socialize	263	261	265	20	875
14	ComputerGame	49	47	49	12	145
15	Dress	54	52	58	3	237
16	Study	94	94	93	11	595
17	School	100	103	99	3	457
18	RA	6	5	7	3	21
19	Homework	70	67	70	2	342
20	Travel	43	40	42	2	195
21	Cook	66	65	60	2	348
22	Hospital	6	6	6	3	22
23	Praying	2	2	2	3	13
24	WorkAtHome	48	50	48	5	277
25	Coloring	9	9	10	3	40
26	Walk	248	250	248	43	692
27	Meeting	35	36	37	5	117
28	Clean	49	49	52	7	155
29	Hygiene	27	29	31	3	111
30	Watch	586	618	590	61	1463
31	Computer	125	129	129	21	417
32	Exercise	86	90	92	5	356
33	Class	246	248	258	11	929
34	Relax	448	498	463	30	1274
35	Work	1085	1144	1139	182	3215
36	LeaveHome	10	11	11	2	19
37	BusStop	5	5	6	2	28
38	Gardening	11	13	14	2	38
39	MakeLunch	1	1	1	2	4
40	Biking	8	8	9	2	38

Fig. 3. Total errors in confusion matrix for activities that contain edges

To investigate the reason for performance decrease when edges are added to feature set, we look at the confusion matrix for activities that contain edges. There are 51 activities in the dataset where transitions between locations occurred and hence these activities contain edges. We remove 12 activities that have only one instance that contain edges. Figure 3 shows the list of these activities that have at least two instances in the dataset that contain edges. It also shows the total number of instances for each activity

and the number of instances where edges occurred. We find false positive, false negative values for each of these activities from the confusion matrix and compute the total error (false positive + false negative).

We compute the total error for three cases: nodes added to basic features, edges added to basic features, nodes and edges added to basic features and present this information in Fig. 3 in columns 'Total Error for Nodes', 'Total Error for Edges' and 'Total Error for Nodes and Edges'. Among them 20 activities that are colored in red and blue in Fig. 3 demonstrated reduced error using either edges or both nodes and edges. Blue colored activities showed reduced error with addition of edges and red colored activities showed reduced error with use of both nodes and edges. 19 activities that are colored black in Fig. 3 showed increased error with the use of nodes and edges.

As a next step, we tried some basic feature selection techniques to test whether it may improve the result. Through feature selection, we may be able to keep and use only useful features and eliminate features that had a negative effect on accuracy. This may help with the problem of overfitting as well. We present the result of our initial experiment of feature selection with Extra Tree classifier in Table 3. We used the k-best features selector from scikit-learn based on the mutual information criteria to select 100-best features from the basic feature set, nodes added to the basic feature set, edges added to the basic feature set and nodes and edges added to the basic feature set. As demonstrated in Table 3, better overall accuracy is achieved with the addition of edges only and both nodes and edges compared to basic features. The basic features with edges showed the best performance. In future work, we plan to experiment with varying the value of k and with other feature selection methods to see whether the overall accuracy can be improved further.

Table 3. Accuracy in percentage for basic features vs graphical features with feature selection

Features	Accuracy in percentage
Basic	57.65
Basic + Nodes	57.13
Basic + Edges	**57.67**
Basic + Nodes + Edges	57.50

5 Discussion

Compared to a non-graphical typical feature set, graphical feature sets provide improvements over non-graphical features with nodes performing the best among all graphical feature sets. However, adding edges decreased the performance in some cases.

In the current dataset all sensor events were collected at one second intervals. Both basic and graphical features of each instance are computed from sensor events collected in a five second window. Each five second window has an activity label provided by the user. Most activities continue past one window. We already obtained better performance for some activities using transitional information available in only a five-second window. Some activities may benefit from a larger window in order to allow for more

transitions in the activity graph, but that would require an additional data collection effort in the future.

Also, there may be noise both in labeling activities and in determining location categories. Activity labels are dependent on getting correct information from users about the activity they are doing. There are many instances of activities in the dataset during which no geolocation data is collected. While extracting the location category from raw latitude and longitude values using the Nominatim Database, for some location categories "None" is returned, indicating unknown location category. The ability to get more accurate location category information may improve predictions.

We are predicting classes across behavior of all users in this experiment. User-wise activity prediction may give better result because movement patterns can vary from user to user.

6   Conclusion

We present a Graphical Feature based Framework with the goal to improve prediction tasks for different sensor networks by representing the sensor networks in graph form, extracting graphical features from these graphs, and adding those features to the typical set of features for the task, to be fed to classifiers. In this work, we apply this framework to smart phone sensor data for the task of activity recognition. The results demonstrate that adding spatial and transitional features improves the activity recognition accuracy compared to typical non-graphical and non-spatial feature sets. Without feature selections, nodes perform the best. However, analyzing the confusion matrix shows that adding edges can decrease total error in many activities. Initial investigation with feature selection shows that use of feature selection may help through eliminating extra and non-helpful edges and also may help with overfitting due to the large number of graphical features. We plan to try other feature selection methods and find the set of helpful selected graphical features. In the future, using larger window sizes to help extract more transitions for ongoing activities can be used to further improve the performance of adding graphical features. User-wise activity prediction along with graphical features will reflect an individual's movement pattern and hence may improve activity prediction. Along with the use of edges and the combination of nodes and edges, larger sub-graphs can be considered as part of the graphical features to discriminate among activities.

References

1. Akter, S.S., Holder, L.B.: Activity recognition using graphical features. In: 13th International Conference on Machine Learning and Applications (ICMLA), Michigan (2014)
2. Akter, S.S., Holder, L.B.: Using graphical features to improve demographic prediction from smart phone data. In: Proceedings of the 2nd International Workshop on Network Data Analytics, Chicago (2017)
3. Aminikhanghahi, S., Fallahzadeh, R., Sawyer, M., Cook, D.J., Holder, L.B.: Thyme: improving smartphone prompt timing through activity awareness. In: 16 th International Conference on Machine Learning Applications, Cancun (2017)

 4. Avci, A., Bosch, S., Marin-Perianu, M., Marin-Perianu, R., Havinga, P.: Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: a survey. In: 23rd International Conference on Architecture of Computing Systems (ARCS), Hannover (2010)
 5. Bouchard, K., Holder, L.B., Cook, D.J.: Extracting generalizable spatial features from smart phones datasets. In: AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments, Phoenix (2016)
 6. Chetty, G., White, M., Akther, F.: Smart phone based data mining for human activity recognition. Proc. Comput. Sci. **46**, 1181–1187 (2015)
 7. Garcia-Ceja, E., Brena, R.: Long-term activity recognition from accelerometer data. Proc. Technol. **7**, 248–256 (2013)
 8. Liao, L., Fox, D., Kautz, H.: Location-based activity recognition. In: Advances in Neural Information Processing Systems (2006)
 9. Python Software Foundation, "geopy1.11.0". https://pypi.python.org/pypi/geopy
10. Hall, M.A.: Correaltion-based feature selection for machine learning (1999)
11. Haklay, M., Weber, P.: Openstreetmap: user-generated street maps. IEEE Pervasive Comput. **7**(4), 12–18 (2008)
12. https://nominatim.openstreetmap.org/
13. Manrique, J.A., Rueda-Rueda, J.S., Portocarrero, J.M.: Contrasting internet of things and wireless sensor network from a conceptual overview. In: 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 252–257. IEEE, December 2016
14. Khalil, N., Abid, M.R., Benhaddou, D., Gerndt, M.: Wireless sensors networks for internet of things. In: 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 1–6. IEEE, April 2014
15. Zeng, J., Min, J.: A systematic framework for designing IoT-enabled systems. Serv. Trans. Internet Things **1**, 23–31 (2017). https://doi.org/10.29268/stsc.2017.0002
16. El Khaddar, M.A., Boulmalf, M.: Smartphone: the ultimate IoT and IoE device. In: Smartphones from an Applied Research Perspective. InTech (2017)