IEEE 802.11 as a Passive Sensor for Crowd Density Counting in Closed Areas

Sajidah Shahadha Mahmood
Department of Public Relations, University of AlIraqia, Baghdad, Iraq

ABSTRACT

Crowd density counting obtained popularity in recent years with COVID-19 and the social separation constraints that have to be enforced in public areas. Many methods and techniques can be utilized for crowd density counting. However, these techniques depend on expensive equipment and massive deployment of different sensors in the targeted area. In this work, a simple crowd density counting framework based on measuring the received signal strength (RSS) of IEEE802.11, known as, WIFI in closed areas is leveraged. An access point (AP) and a Raspberry PI kit has been located in a closed area to harvest the RSS value when people pass through the area. K-NN machine learning algorithm has been trained with different features extracted from the RSS to predict the number of people in the area. Finally, an Android smartphone App has been written to monitor the counted number to enforce the counting constraint in the closed areas. The model has been deployed in the engineering faculty. Our results show that K-NN with RSS features for passively crowd density counting achieved 88% accuracy. However, this accuracy dropped to 75% with people running scenario.

Keywords: K-NN Algorithm, IEEE802.11 (WIFI), Crowd Density Counting, Machine learning

Corresponding Author:

Sajidah Shahadha Mahmood
Department of Public Relations, University of AlIraqia
Baghdad, Iraq
E-mail: Sajidah.sh.mahmood@aliraqia.edu.iq

1. Introduction

Using data harvested from different sensors to reveal direct or indirect information has proliferated in the past decade [1]. Direct information is easy to be extracted since these sensors have been engineered to measure these physical phenomena. However, indirect information requires process the data and analytic them using different mathematical methods. Machine learning methods have dominated in this area. As an example, smartphones sensors have been used to reveal health, security and tracking information. However, other technical components and devices have been leveraged as passive sensors to reveal indirect information. For example, Electrical usage in houses has been used to predict the working hours of an appliance [2]. Using liquids' connectivity percentage as a metric to predict their type is another example of revealing information from passive sensors [3]. The idea of revealing passive information from passive sensors can be extended for crowed density counting

Crowed density counting is the process of counting the peoples in an area. The area could be indoor or external areas. This process is an important issue with the epidemic of COVID-19 especially in closed areas. Wireless signals received characteristics are impacted with different obstacles occurred in an area, such as, walls, doors and people. This impact is a drawback on the received signal. However, it can be leveraged as a passive sensor for different reasons, such as, localization, tracking and crowed density counting.

Different wireless signals are surrounding us, such as, mobile communication signals, IEEE802.15.4, IEEE802.15, IEEE802.11, near field communication (NFC) and RFIDs. IEEE802.11, known as WIFI has dominated for data wireless networking. This technology is used as indoor wireless data network. It is cheap, easy to configure and their interface cards are founding in smartphones, tablets, computers, smart-TVs and even in simple embedded systems. Their separation has motivated the usage of this network as passive sensors.

Crowd counting utilizing different techniques has proliferated in the last few years. The proposed methods and tools can be divided into two main categories; device free methods and equipment-based methods. In the following subsections these methods will be introduced.

In equipment-based method, different sensors have to be installed on user smartphone devices or the users have to carry different sensors. In [4], the authors proposed cricket technique to localize and estimate users' location in an area, this method depends on developing asmall circuit boards with ultrasonic sensors to measure the distances using time-difference-of-arrival (TDOA). This method has been evaluated in 2 dimensional environments in [5]. In [6], RFID has been used to localize and count the people in a region. All users have to carry RFID tags with them to be localized. In [7], smartphones' ambience Fingerprinting has been proposed to find users' logical locations. It has been shown that each area has a surrounding fingerprint of light, colors, voices and noises. The author attempted to combine all of these sensors to estimate the logical location of the users. 87% accuracy has been achieved with the proposed algorithm. In [8], the authors proposed a crowd density counting method based on counting the number of surrounding smartphones based on Bluetooth fingerprint. An accuracy of 80% has been recorded. In [9], Authors proposed a crowd density counting based on smartphones' audio tones, their system has been constructed as a smartphone's App and have to be deployed in the smartphones in an area. They claimed an accuracy of 90%. All of these methods depend on separating applications or devices among the people to be counted. However, most of the time crowd density estimation has to be implemented passively. In this year, the epidemic of COVID-19 and government policies of reducing the crowd density of people in close areas motivated malls and shops to passively count the number of shoppers. This cannot be done using the equipment-based methods.

In Device free methods category, each area should be equipped with sensors. The users don't have to carry or install any special Apps on their smartphones. Some of these techniques require the deployment of sensors as a wireless sensor network (WSN) for crowd density counting and tracking, such as, [10]. Other techniques require the deploying of cameras and image processing techniques as in [11]. However, these techniques are costly. The popularity of IEEE802.11, commercially known as WIFI, motivated researchers to utilize the WIFI signal to estimate the crowd density in areas. In [12], WIFI signal has been utilized as a passive sensor for crowd estimation using channel state information (CSI) from OFDM-based system. IEEE802.11n Intel card has been used to obtain the CSI information. A Percentage of nonzero Elements (PEM) metric has been proposed to utilize CSI to count the number of people. The proposed method has shown a relation between moving people in an area and CSI. In [16], CSI has been shown to have 30% more accuracy than FCC method. However, not all IEEE802.11 cards allow the developer to harvest and collect the CSI information. In [17], WIFI packet counting has been used for crowd density counting, the author has shown that this technique has promising results. In [13], RSSI of WIFI has been leveraged for crowd density counting. A WIFI transmitter and a receiver have been used. The proposed method depended on Kullback-Leibler divergence with a probability-model. The proposed model has shown potential on deploying in indoor and outdoor environment. This method, has been deployed in [14], for crowd density counting of people in a building through the wall. WIFI transmitter and receiver have been deployed outside a building. RSSI signal has been used to show the relation between the number of people in the building and the inter-event times carries. In [15], WIFI has been used passivity to count the number of people entering and leaving a door. The WIFI devices have been deployed on the door to count and find the movement direction. 95% accuracy has been claimed.

In this work, WIFI has been used as a passive sensor for crowd density counting of people in closed areas. A model has been constructed using K-NN machine learning supervised algorithm. The model uses the WIFI received signal strength (RSS) as a raw data to estimate the number of people in the area. A WIFI-transmitter, using Cisco Linksys Access point (AP), and a WIFI receiver, using Raspberry PI 4 board are used. Finally, an Android App has been written as a monitoring App to obtain the RSS values from the Raspberry PI 4 and to predict the number of users utilizing the deployed K- nearest neighbor (K-NN) algorithm. Our contribution in this work can be summarized as following

- Proposing a crowd density monitoring framework leveraging WIFI, Raspberry PI and Android smartphone. The framework
- Using Python with Raspberry to train K-NN machine learning model to predict the number of people in closed room passively from the features extracted from the RSS value of WIFI.
- Writing an Android App to retrieve the counted number over the Internet to monitor the closed area

This work defers from other works in the following. First, WIFI RSSI as raw data will be used and fed to K-NN machine learning algorithm to estimate the number of user using a WIFI-access point and a receiver WIFI card. The trained model is designed and deployed in Android smartphone. In this method, the App will monitor the number of users in different location from a single control point. The rest of this paper is organized as follows; section 2 shows the materials and methods follows in this paper. Section 3 introduces the theory and experiments including the the android application and the whole framework with all components. Section 4 views and discusses the obtained results. We conclude this paper in section 5.

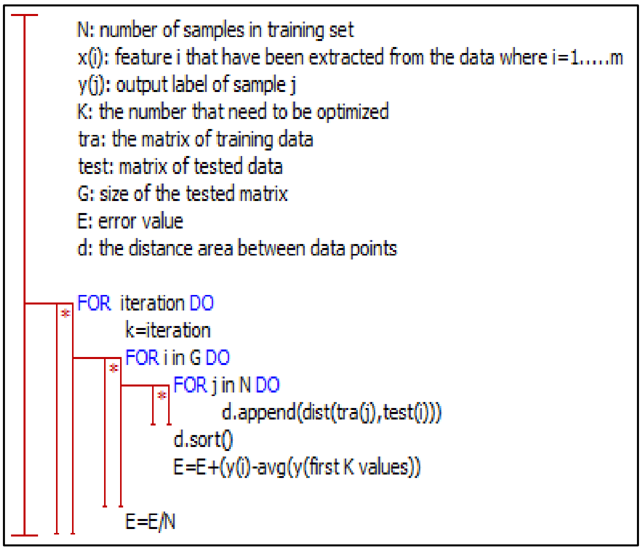
2. Materials and methods

To In this section, the proposed platform is introduced. This section is divided into three main subsection; the K-NN model, the android App and the full platform.

2.1. The proposed K-NN model

K-NN [19] is one of the simplest supervised machine learning algorithms. The training stage of this algorithm is the process of finding the K value to reduce the error. The algorithm works as follows. First, the harvested features and the output labels are divided into three classes; training, testing and validation. Second, the testing data is used to find the optimal K value for the algorithm. The process start by an initial value of K. second, the Euclidean distance between the first testing sample and the training data is calculated. Subsequently, the training data is arranged in ascending order according to their distances. Finally, the first K training samples outputs are averaged as the output of the testing sample. Subsequently, mean square error (MSE) is used to calculate the cost of the model. This process is repeated for different K values until an optimal value with lowest cost is found. Eq. (1) shows the MSER. Fig. 1. Shows how K-NN is trained.

$$MSER = \frac{1}{n}\sum_i (y_i - y_i^{`})^2 \tag{1}$$

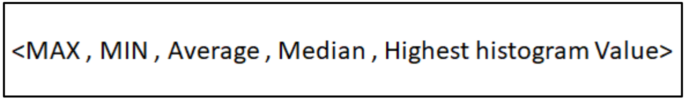


```
N: number of samples in training set
x(i): feature i that have been extracted from the data where i=1.....m
y(j): output label of sample j
K: the number that need to be optimized
tra: the matrix of training data
test: matrix of tested data
G: size of the tested matrix
E: error value
d: the distance area between data points

FOR iteration DO
    k=iteration
    FOR i in G DO
        FOR j in N DO
            d.append(dist(tra(j),test(i)))
        d.sort()
        E=E+(y(i)-avg(y(first K values))
    E=E/N
```

Figure 1. K-NN Training Process

In this work, the harvested RSS values of the WIFI have been used to extract five features. A window of 1 second has been used to gather 100 RSS values. These 100 values in the window have been leveraged to extract the features. These features are; the median value, the maximum value, the average value, the highest histogram value and the minimum value. The output is only one column of the number of users in the room. To extract the highest histogram value, the histogram of the recorded values is calculated since the value of the received RSS is a negative integer value. The absolute value is used to convert it into a positive value. Subsequently, the histogram is calculated. Fig. 2 shows an example of data tuple.



<MAX , MIN , Average , Median , Highest histogram Value>

Figure 2. Example of input data tuple

3. Theory and experiemnts

3.1. Android smartphone application

MIT App inventor 2 [20] has been utilized to write a simple Android smartphone App. The application can be written in two different methods. First, the App can be a reporting application only. In this way, the application is responsible of generating an HTTP request to a web-server installed on the raspberry PI. The server answers with the numbers of the last 10 seconds. The App averages these values and it shows only the number of counted people in the room. The second method to write the application is to allow it to calculate and predict the number of users. In this way, the HTTP requires receives only the raw data from the access point, 1k RSS values will be send from the Raspberry to the smartphone App.

The app will use the received data to extract the five features of the K-NN. Subsequently, it will be responsible of arranging the data points in the training dataset according to the new data point and it will average the output data. Both of these methods can be utilized, however, we adopted the first scenario for two reasons. First, the training dataset is heavy to be installed on every smartphone. Second, the battery constraint of the smartphone motivated us to convert the App into a reporting App only. However, if the dataset is on the smartphone, the user may re-train the algorithm when the AP or the Raspberry PI location changes in the room. Fig. 3 shows the main activity of our application.

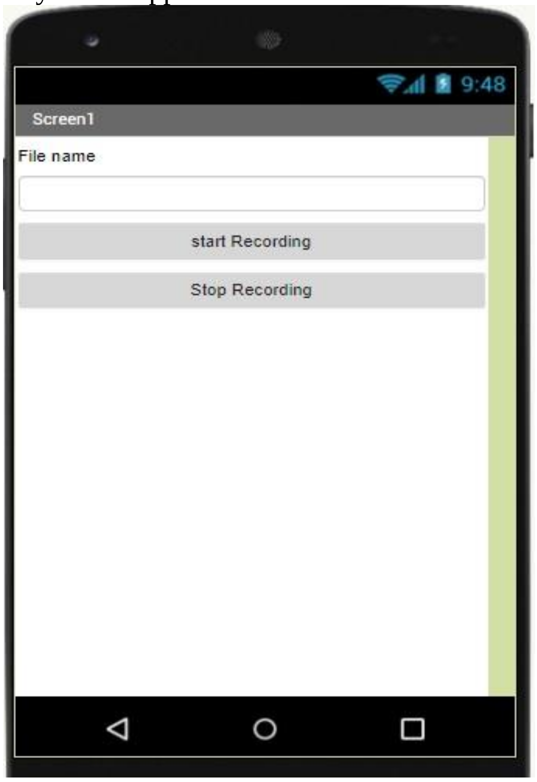


Figure 3. Android smartphone application

3.2. The whole framework

Fig. 4 below shows the proposed framework to passively count the number of people in a closed area and obtain this count using a smartphone App. The training dataset is saved in the Raspberry that located beside one walls of the room opposite to the AP. The raspberry is closed and the dataset has been encrypted. K value of K-NN has been optimized. The Raspberry located in the room has four main tasks. First, it is responsible of recording the RSS of the access point every 10ms. Second it responsible of windowing the recorded data every 1 second. Windowing means that the data point is divided 100 points every time a new data is recorded. Third, it responsible of extracting the four utilized features in this work; the maximum RSS value recorded over the 100 values, the minimum value, the average value and the median value. To obtain the median, the data has to be recorded in ascending order and the value number 50 is used. Third, it is responsible of executing the K-NN algorithm and to save the number of users predicted each second. Finally, the kit has a configured web-server to receive HTTP requests and to responses with the number of predicted people in the room.

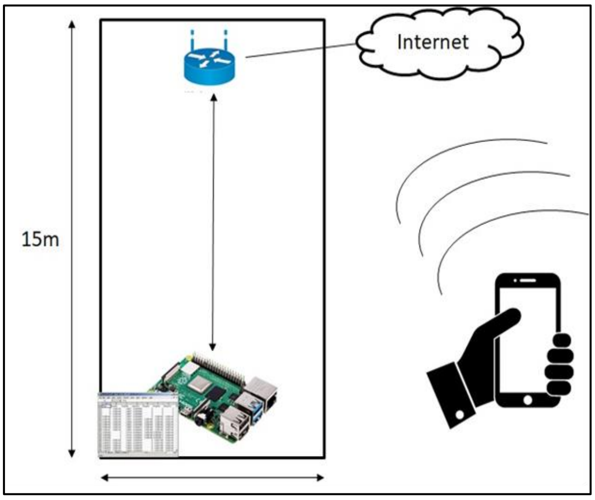


Figure 4. The framework of the system

3.3. Experiments

To train and evaluate the proposed framework, a Raspberry PI 4 and a Cisco AP has been located in corridor in the Engineering faculty. The corridor is 15m long and 6m width. The AP and the Raspberry have been placed beside the walls in the middle of the corridor over a 60cm tables as shown in Fig 3. 7 students have entered the corridor in this experiment in four different scenarios. The following subsections show the configuration in each scenario.

3.3.1. The first scenario

This scenario is called the standing scenario. It can be observed in closed locations with queue of peoples, such as, banks and supermarkets. The configuration of all experiments in this scenario depends on number of students standing without moving in single location for a period of time. In this scenario, eight different experiments have been conducted. In the first experiment, the RSS value of the empty corridor has been recorded for 30 seconds. Subsequently, the first student entered the corridor and stood in the first location near the Raspberry for 10 seconds. After that, the student moved to the second location far from the Raspberry and stayed for another 10 seconds. This operation has repeated for six different locations. Fig.5.a shows these locations in the corridor. In the second experiment, two students have stood in these six locations. This means the experiment has been repeated three times. In the next experiment three students entered the corridor and the experiment conducted two times. After that, four students, five and six students have entered the area for only one time. The locations utilized are from location ‘L1’ to location ‘L4’ and so on. In the last experiment with seven students in the area, the 7th student stood directly in front of the AP.

3.3.2. The second scenario

This scenario has been configured as the first scenario with eight experiments. The different between this one and the first scenario is that the students have been asked to sit down on chairs in the six locations shown in figure 4.a. The time intervals have been configured to be the same as in the first scenario. This scenario attempts to mimic restaurants and classrooms where people are sitting down for long time period.

3.3.3. The third scenario

This scenario is called the walking scenario. It attempts to emulate consumers and pedestrian in malls, universities and hypermarkets. In this scenario, the same six locations have been leveraged. However, students have been asked to move between these locations and vice versa for 30 seconds. Eight different configurations or experiments have been implemented. In the first experiment, one student has moved between two locations for 30 seconds. After that, the same student has moved to walk between another two locations until the student cover all the six locations. This means that this experiment has been repeated three times. Subsequently, two students have walked between the locations for 30 seconds. After that, three, four, five, six and seven students have walked between the locations. Fig.5.b shows the walking patterns between the locations.

3.3.4. The fourth scenario

It is called the running scenario, the configuration of this scenario is the same as the third scenario however, the students have been asked to speed up between the locations. We did not measure the speed of the students, however, they have been asked to move 20 times between the two locations. Fig. 5.b shows the patterns for walking and running.

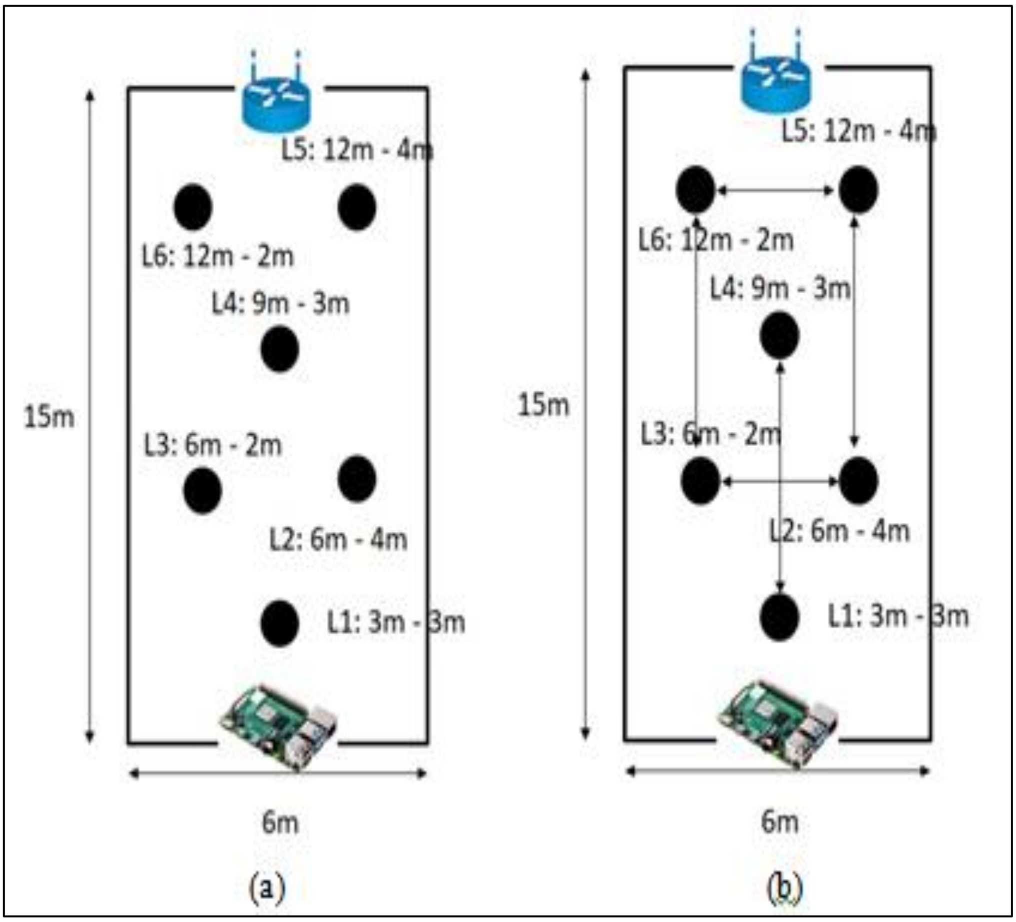


Figure 5. The experiment Configurations: (a) Locations for Scenario 1 and 2 (b) Patterns for Scenarios 3 and 4

3.3.5. The dataset

The RSS values have recorded with frequency of 100Hz. In one second, 100 reading have been recorded for RSS values. From the first and the second scenarios 96K RSS values have been recorded. From the third and the fourth scenarios 60K RSS have been recorded. Each scenario has been leveraged alone to train the K-NN model to extract the K value. In the end all the data has been used to train one model. The recorded data has been divided into training, testing and validation with the percentages of 70%, 15% and 15%. Finally, the system has been validated form a smartphone outside the corridor for about an hour to assess its accuracy.

4. Results and discussion

As mentioned, five different K-NN models have been trained. Each model has been trained with the data harvested from each scenario. In the end, all the data has been collected and one model has been trained from all scenarios. The following subsections show the results of each model.

4.1. The standing scenario

Fig. 6 shows the validation of the K-NN model for the first scenario. It can be observed form the figure that K-NN predicts model caught the real recorded people in the area 6 of the 8 configurations. Moreover, we can observe that error in prediction in the last two configurations was only with one person less than the real number. The value of K has been trained to be 8 for this scenario

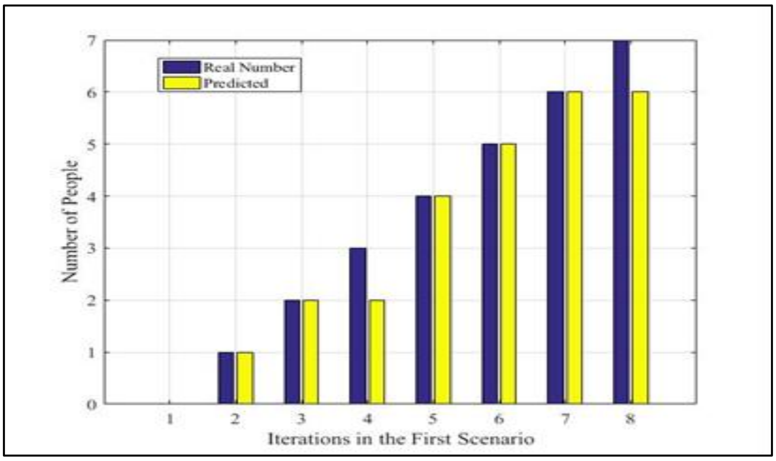


Figure 6. The standing scenario

4.2. The sitting scenario

Fig. 7 shows the validation of the K-NN model for the second scenario. It can be observed form the figure that K-NN predicts model caught the real recorded people in the area 7 of the 8 configurations. Moreover, we can observe that error in prediction in the last configuration was one person less than the real number. The value of K has been trained with the same value as in the first scenario.

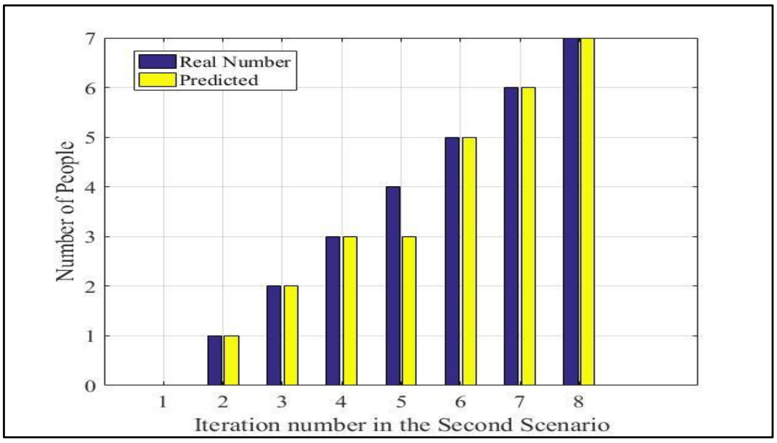


Figure 7. The sitting scenario

4.3. The walking scenario

Fig.8 shows the validation of the K-NN model for the third scenario. It can be observed form the figure that K-NN predicts model caught the real recorded people in the area 7 of the 8 configurations. Moreover, we can observe that error in prediction in the last configuration was one person higher than the real number. The value of K has been trained with six for this scenario

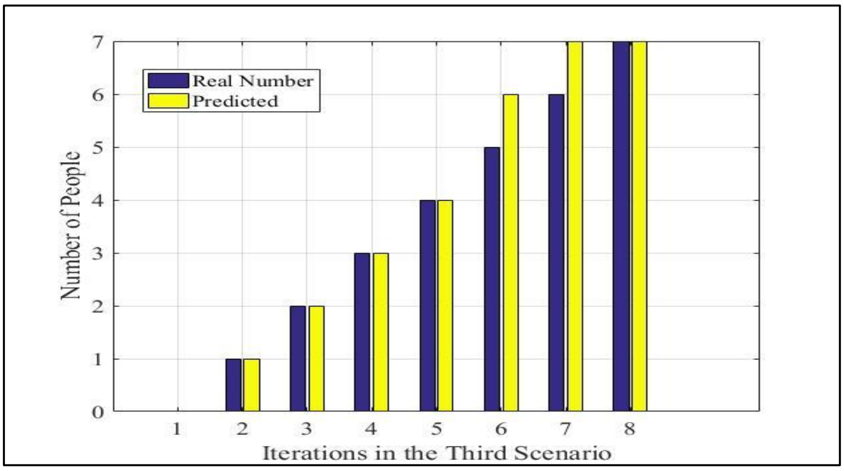


Figure 8. The walking scenario

4.4. The running scenario

Fig. 9 shows the validation of the K-NN model for the third scenario. It can be observed form the figure that K-NN predicts model caught the real recorded people in the area only 4 of the 8 configurations. The value of K has been trained with three for this scenario. This configuration got the worst prediction value since the moving pattern of the students impact the signal shape more than the other setups. However, it is very hard to find people moving in a fast way in closed areas as in this scenario.

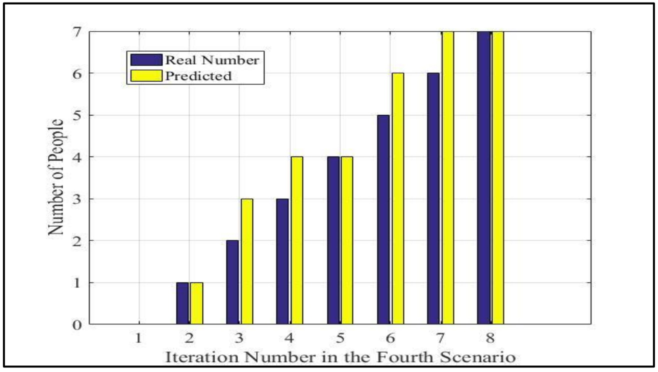


Figure 9. The running scenario

4.5. The mixed scenario

In this configuration, the data collected from all scenarios have been mixed and only one K-NN model has been trained. However, we have created two different mixed models; the first one with all the collected data and the second with all the data neglecting the last scenario. Subsequently, we utilized this model in the Raspberry and tested the model in real time for one hour. We have counted the people entering the corridor and compared them with the predicted values. Subsequently, the two values have been subtracted to find the prediction error value. Fig. 10 shows the CDF of the prediction error value for all the recorded data. We can observe that the accuracy of this model is above 75%. However, when the last scenario is neglected from the training data, the accuracy has changed.

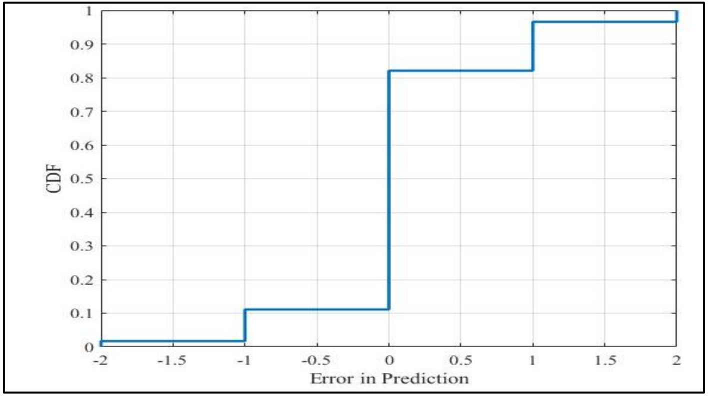


Figure 10. The prediction error with the data of all scenarios

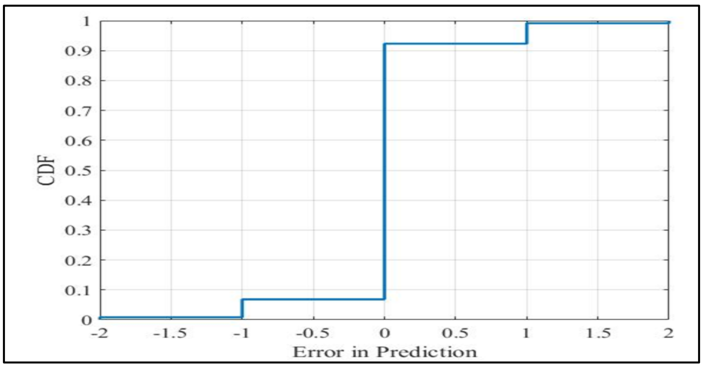


Figure 11. The Prediction Error neglecting the Last Scenario

Fig. 11 shows the CDF of the prediction error. We can observe that the accuracy exceeded 88%.

5. Conclusion

In this paper, a new monitoring and counting framework for crowd density based on passive WIFI RSS measuring has been proposed. K-NN supervised machine learning algorithm has been leveraged to train the prediction model. The model measure predicts and sends the predicted value to an Android smartphone as a monitoring station. The model has been trained in four different scenarios of peoples entering and opened area. Subsequently, the trained model has been evaluated in real time for one hour and its accuracy has been recorded. The model achieved 88% accuracy with standing and slowly moving people. However, the accuracy of the model drops to approximate 75% with running people. To enhance the accuracy of the model, other features can be extracted from the RSS, such as, frequency domain features. Moreover, other supervised machine learning algorithms can be deployed, such as, deep learning and support vector machine (SVM). However, for the simplicity in this work, K-NN has been leveraged.

References

[1] M. Masoud, Y. Jaradat, A. Manasrah, and I. Jannoud, “Sensors of Smart Devices in the Internet of Everything (IoE) Era: Big Opportunities and Massive Doubts," *Journal of Sensors 2019*, 2019.

[2] K. Basu, L. Hawarah, N. Arghira, H. Joumaa, and S. Ploix, "A prediction system for home appliance usage," *Energy and Buildings* 2013 *67*, pp. 668-679.

[3] M. Masoud, Y. Jaradat, A. Manasrah, and B. Taleb, "What I Have in My Cup? A Liquid Identification Mechanism Based on Electrical Connectivity," *In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 2019, pp. 689-693.

[4] C. Lin, F. Gao, H. Dai, L. Wang, and G. Wu,"When wireless charging meets fresnel zones: Even obstacles can enhance charging efficiency," *In 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 2019, 1-9.

[5] N. Priyantha, A. Chakraborty, and H. Balakrishnan,"The cricket location-support system," *In Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 2000,32-43.

[6] Y. Wang, S. Goddard, and L. C. Perez,"A study on the cricket location-support system communication protocols," *In 2007 IEEE international conference on electro/information technology*, 2007, pp. 257-262.

[7] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “Landmarc: indoor location sensing using active rfid,” Wireless networks, 2004, vol. 10, no. 6, pp. 701–710.

[8] M. Azizyan, I. Constandache, and R. R. Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," *In Proceedings of the 15th annual international conference on Mobile computing and networking*, 2009, pp. 261-272.

[9] J. Weppner, and P. Lukowicz,"Collaborative crowd density estimation with mobile phones," *Proc. of ACM PhoneSense*, 2011.

[10] P. Kannan, S. P. Venkatagiri, M. C. Chan, A. L. Ananda, and L. Peh, "Low-cost crowd counting using audio tones," *In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012, pp. 155-168.

[11] Y. Yuan, C. Qiu, W. Xi, and J. Zhao,"Crowd density estimation using wireless sensor networks," *In 2011 seventh international conference on mobile Ad-hoc and sensor networks*, 2011, pp. 138-145.

[12] M. Li, Z. Zhang, K. Huang, and T. Tan, “Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection,”2010, *in Proceedings of ICPR2008*, pp. 1–4.

[13] W. Xi, J. Zhao, X. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, "Electronic frog eye: Counting crowd using WiFi," *In IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, 2014, pp. 361-369.

[14] S. Depatla, A. Muralidharan, and Y. Mostofi,"Occupancy estimation using only WiFi power measurements," *IEEE Journal on Selected Areas in Communications*2015, *33*, no. 7, pp. 1381-1393.

[15] S. Depatla, and Y. Mostofi, "Crowd counting through walls using WiFi," *In 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)* 2018, pp. 1-10.

[16] Y. Yang, J. Cao, X. Liu, and X. Liu, "Wi-count: Passing people counting with COTS WiFi devices," In *2018 27th International Conference on Computer Communication and Networks (ICCCN)* 2018, pp. 1-9.

[17] J. Li, P. Tu, H. Wang, K. Wang, and L. Yu, "A novel device-free counting method based on channel status information, 2018" *Sensors 18,* no. 11, pp. 3981.

[18] O. Yaik, K.Z. Wai, I. K. Tan, and O.B. Sheng, "Measuring the accuracy of crowd counting using Wi-Fi probe-request-frame counting technique," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* 2016 *8*, no. 2, pp. 79-81.

[19] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data 2016," *Neurocomputing 195*, pp. 143-148.

[20] S. C. Pokress, and J. J. D. Veiga, "MIT App Inventor: Enabling personal mobile computing," *arXiv preprint arXiv*: 2013 *1310*, pp. 2830.