Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Internet of Things (IoT) Fall Detection using Wearable Sensor

To cite this article: Loh Mei Yee *et al* 2019 *J. Phys.: Conf. Ser.* **1372** 012048

Internet of Things (IoT) Fall Detection using Wearable Sensor

Loh Mei Yee[1], Lim Chee Chin[1*], Chong Yen Fook[1], Maslia Binti Dali[2], Shafriza Nisha Basah[1], Lim Sin Chee[3]

[1] School of Mechatronics Engineering, University Malaysia Perlis, Perlis, Malaysia
[2] Hospital Tunku Fauziah, Kangar, Perlis, Malaysia
[3] First City University College, Bandar Utama, Petaling Jaya, Malaysia.
*cclim@unimap.edu.my

Abstract. The IoT fall detection system detects the fall through the data classification of falling and daily living activity. It includes microcontroller board (Arduino Mega 2560), Inertial Measurement Unit sensor (Gy-521 mpu6050) and WI-FI module (ESP8266-01). There total ten (10) subjects in this project. The data of falling and non-falling (daily living activity) can be identified. The falling is the frontward fall, while the daily living activity includes standing, sitting, walking and crouching. K-nearest neighbour (k-NN) classifiers were used in the data classification. The accuracy of k-NN classifiers were 100% between falling and non-falling class. The feature was selected based on the percentage of accuracy of the k-NN classifier. The features of the Aareal.z (97.14%) and Angle.x (97.24%) were selected due to the good performance during the classification of the falling and non-falling class. The performance of the Aareal.z (58.41%) and Angle.x (57.78%) were satisfactory during the sub-classification of the non-falling class. Hence, the feature of Aareal.z and Angle.x were selected as the features which were implemented in the IoT fall detection device.

1. Introduction

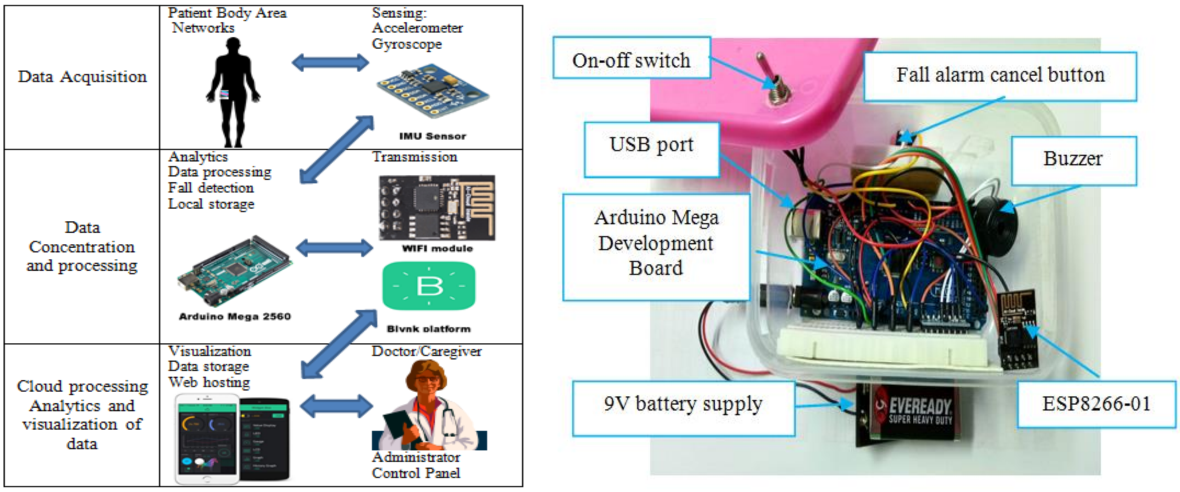
A patient falling will cause bone fracture, haemorrhage and even death. The severity of the injury was divided into four levels, such as minor, moderate, major and death. It can be reduced to the minor level by detecting the patient fall. The fastest the fall detected, the lowest the risk of patient falling [1]. The Internet of Things (IoT) fall detection system is developed to overcome some problems. For instance, limited number of the healthcare staffs in the rural or city based healthcare sector, increasing number of patient fall and limitation of the current systems [2]. The problem of limited number of the healthcare staffs in the rural or city based healthcare sector occurs when the healthcare staff is not enough to provide professional care to the increasing number of patients [3], [4], [5]. The problem of the increasing number of patient fall occurs when the patient fall is unable or hardly to be detected. Lastly, there is some limitation of the current systems. The current systems include a camera-based fall detection system [6], 3D fall detection system [7], and multiple sensor fall detection system [8]. This type of system is heavy in size, non-portable and unable detects fall in real-time [9]. Hence, IoT fall detection system is needed to be developed to overcome these limitations.

2. Methodology

2.1 Structural design of the IoT fall detection

An overview of an IoT fall detection device consists of three main parts including data acquisition, data concentration processing and a back-end system is shown in Figure 1(a). A sensor node of an

IoT-based fall detection system is responsible for acquiring motion data and transmitting the data via a wireless communication protocol to a smart gateway. Depending on particular fall detection systems, the collected data can be pre-processed or kept intact before being transmitted. In this system, the collected data (raw data) is processed by complex algorithms or methods like fall detection based on k-nearest neighbor algorithm. Then, the processed data will be classified into falling and non-falling condition. The data and result will be stored in cloud processing analytics. The visualization of data is in the mobile application.


Data Acquisition
Patient Body Area Networks
Sensing: Accelerometer Gyroscope
IMU Sensor
Data Concentration and processing
Analytics Data processing Fall detection Local storage
Transmission
WIFI module
Arduino Mega 2560
Blynk platform
Cloud processing Analytics and visualization of data
Visualization Data storage Web hosting
Doctor/Caregiver
Administrator Control Panel
On-off switch
Fall alarm cancel button
USB port
Buzzer
Arduino Mega Development Board
9V battery supply
ESP8266-01

(a) (b)

Figure 1. An IoT fall detection device (a) Overview system (b) Prototype

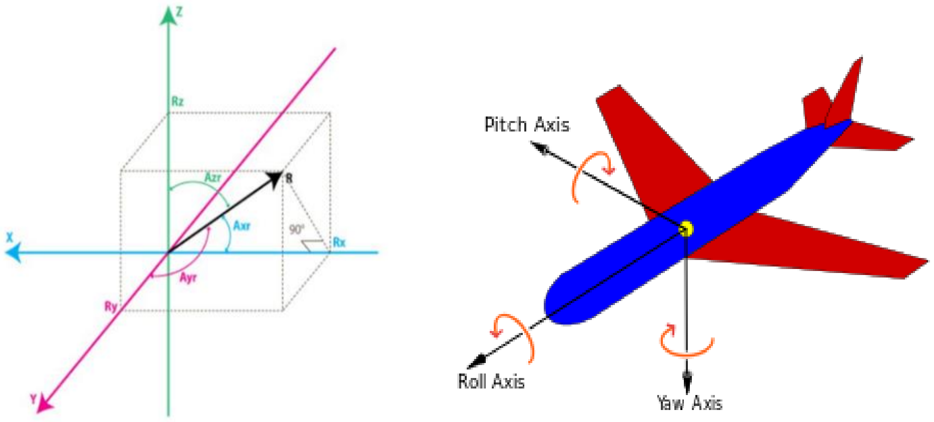
This system includes of six components, such as microcontroller board (Arduino Mega Development Board (AMDB)), IMU sensor (Gy-521 MPU6050), WI-FI module (ESP8266-01), buzzer alarm, on-off switch and push button as shown in Figure 1. Arduino Mega Development Board was equipped with the ATMEGA2560 processor, which features a maximum clock rate of 16MHz. It was embedded with 54 general-purpose digital I/O pins and 6 communications TX/RX pins that used in the connection with WI-FI module (ESP8266-01). It also supports serial communication of I2C and SPI bus which was used in the connection with the IMU sensor (Gy-521 MPU6050) [10].

The construction of WI-FI module based on IEEE 802.11, is able to support transparent transmission mode and multiple network protocol [11]. It is a self-contained SOC with integrated TCP/IP protocol stack that can allow any microcontroller access to the WI-FI network. It features 802.11 b/g/n, WI-FI Direct or WI-FI peer-to-peer (P2P) is capable of hosting an application of offloading all WI-FI networking functions from another application processor. The module has an integrated 32-bit CPU, which can be used as application processor, and 1MB flash memory [12]. ESP8266-01 is low energy consumption and suitable for the application in mobile devices, wearable electronics and networking application design [13]. ESP8266-01 is selected as the IoT device based on the following strength and advantages, such as low power consumption, easy to compute, light weight and cost-effective.

The piezoelectric buzzer contains the positive and negative pin. The positive pin of the piezoelectric buzzer is connected to the digital pin 13 on the AMDB while the negative pin of the piezoelectric buzzer is connected to the ground of the AMDB. The 3 pins toggle switch acts as on-off switch. It contains pin 1, 2, 3, 4 and 5. When the switch turns to pin 1, the circuit will become “ON” state, while the circuit will become “OFF” state when the switch turns to pin 2. The pin 3, 4, 5 are connected to the ground pin, digital pin 3 and 5V on the AMDB, respectively. The pushbutton act as fall alarm cancel button. When the button is pressed, the fall alarm will be cancelled. The pin 1 is connected to the digital pin 4 on the AMDB and the pin 2 is connected to the ground. The prototype of the IoT fall detection device is shown in Figure 1(b).

The IMU sensor in small size and can be placed in different position of the human body. IMU is unaffected by the external signals or reference [14-15]. Gy-521 MPU6050 is a device that combines the 3-axis gyroscope and a 3-axis accelerometer on the same silicon die together with an on-board

Digital Motion Processor (DMP) as in Figure 2. R represents resulted distance of the Rx, Ry and Rz and calculated by Equation 1. The angle of Axr, Ayr and Azr are calculated by using Equation 2, 3 and 4.



(a) (b)

Figure 2. IMU sensor characteristic: (a) Tri-axial accelerometer [16]; (b) Yaw, pitch and roll orientation [17]

$$R = \sqrt{(Rx)^2 + (Ry)^2 + (Rz)^2} \tag{1}$$

Where, R = Resulted distance
Rx = Distance along the x-axis
Ry = Distance along the y-axis
Rz = Distance along the z-axis

$$Axr = \cos^{-1}\frac{Rx}{R} \tag{2}$$

$$Ayr = \cos^{-1}\frac{Ry}{R} \tag{3}$$

$$Azr = \cos^{-1}\frac{Rz}{R} \tag{4}$$

Where, Axr = angle between R and the x-axis
Ayr = angle between R and the y-axis
Azr = angle between R and the z-axis

The gyroscope was used to determine the yaw, pitch and roll orientation of the subject. The yaw, pitch and roll orientation are shown in the Figure 2 (b), which can be calculated by using Equation 5, 6 and 7. A Quaternion is represented in the form of $q_0 + q_1i + q_2j + q_3k$. Quaternion number is used in the calculations that involving three-dimensional rotations, such as yaw, pitch and roll [18].

$$Roll = \phi = \arctan\left(\frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \tag{5}$$

$$Pitch = \theta = \arcsin\left(2(q_0q_2 - q_3q_1)\right) \tag{6}$$

$$Yaw = \psi = \text{atan}\left(\frac{2(q_0q_3 + q_1 + q_2)}{1 - 2(q_2^2 + q_3^2)}\right) \tag{7}$$

.

2.2 Data Extraction

The I2C clock of the IMU sensor is set up as 400 kHz and the serial communication is initialized as 115200 bits per second. The Arduino interrupt pins are set up or configure to fetch the IMU sensor data. The Digital Motion Processing (DMP) is initialized. Then, the Arduino interrupt detection enabled. The data will be stored in the first in, first out (FIFO) buffer [19]. FIFO buffer is temporary data storage and store the data that arrive to a microcontroller peripheral asynchronously. The interrupt status is set as 0x02 memory location to read the data. Then, the tri-axial accelerometer and gyroscope data were read from the FIFO and displayed on the serial monitor. The parameters of the tri-axial distance and angle were extracted.

2.3 Data Testing and Validation

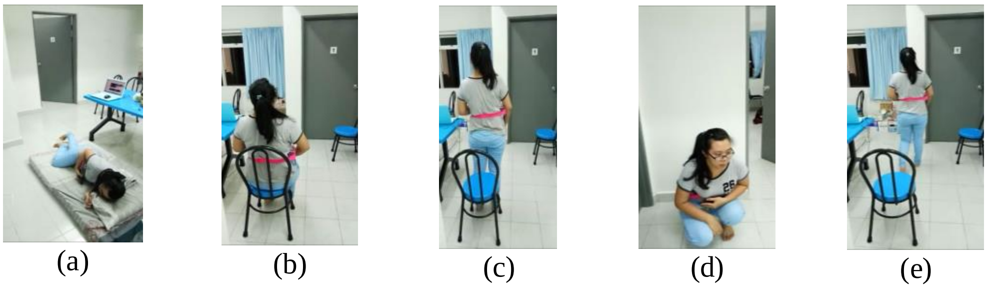
The tri-axial distances were tested in the range from the 0 m to 0.5 m, while the angles were tested from 0 degrees to 180 degrees. The efficiency of the prototype was validated by calculating the average time taken for the IMU sensor calibration process. Ruler, tape, and projector were used during testing and validation process. Each axis was tested and validated by using the same procedures. Firstly, the IMU sensor was placed on the 0 cm of the ruler as the origin of the x-axis. Then, the IMU sensor was moved along the fixed axis of ruler and stopped at the 10 cm of the ruler. The distance travelled was identified by the IMU sensor data and displayed on the serial monitor of the Arduino IDE software. Next, the testing process for the 20 cm, 30 cm, 40 cm, and 50 cm was carried out by using the same procedures.

For the testing and validation of the angle, the IMU sensor was placed horizontally beside the origin of the projector, and then the IMU sensor was elevated to 30 degrees along the semicircle of the projector. The measured angle was identified through the observation of the serial monitor of the Arduino IDE software. The testing for the degree of 60, 90, 120, 150 and 180 was using the same procedure. The IMU sensor data were validated by calculating the average of the total percentage error in the five trials [20] by using Equation 8.

$$\%\ \mathrm{E} = \left|\frac{Actual\ value - Measured\ value}{Actual\ value}\right| * 100\% \tag{8}$$

2.4 Data Collection

Data was collected from 10 (5 males and 5 females) University Malaysia Perlis (UniMAP) students. The average age of subjects was 24 ± 1 year. The average weight was 60.1 kg and the average height was 166.1 cm. Each subject performed 4 different Activity Daily Living (ADL) tasks and frontward falling as shown in Figure 3. The 4 different ADL tasks were the sitting, standing, walking and crouching. For the falling protocol, the subject will perform frontward fell on a mattress with a height of 9.5 cm. For sitting protocol, the subject will sit on a chair with height of 45 cm. For standing protocol, the subject will stand steadily about 2 seconds. For crouching protocol, subject will perform a vertical crouched. For walking protocol, the subject will walk with distance of 120 cm. Each task will be performed 5 times. The features of the acceleration, distance and angle in the x, y and z-axis will be recorded.



(a) (b) (c) (d) (e)

Figure 3. Protocol: (a) Falling; (b) Sitting; (c) Standing; (d) Crouching; (e) Walking;

2.5 Features Extraction

Eight features had been extracted, which are Aareal.x, Aareal.y Aareal.z, Dist.x, Dist.y, Dist.z, Angle.x and Angle.z. The Aareal.x, Aareal.y and Aareal.z were the acceleration in the x, y and z-axis respectively. Then, the Dist.x, Dist.y and Dist.z were the distance of x, y and z-axis respectively. The

Angle.x was angle between subject's motion axis and x-axis; while angle.z was the angle between subject's motion axis and z-axis. The acceleration in the x, y and z-axis were calculated by using the DMP of IMU sensor. The function of mpu.getGetQuaternion (&q, fifoBuffer) and mpu.dmpGetGravity were used to extract the quaternion angle and gravity vectors. The quaternion angle and gravity vectors were used for the calculation of acceleration in the x, y and z-axis.

The distance in x, y and z-axis were calculated through the subtraction between current acceleration and previous acceleration. The result was not multiplied by the time taken because the time taken was set as constant. The angle in the x and z-axis were calculated by the equation 3 and 5. Angle.y was not selected because it was too small and be neglected. Since the motion of the subject was on the reference axis (Aareal.y) during the ADL activities and falling action, hence, the angle between the subject's motion and the reference axis was too small and be neglected.

2.6 Data Classification

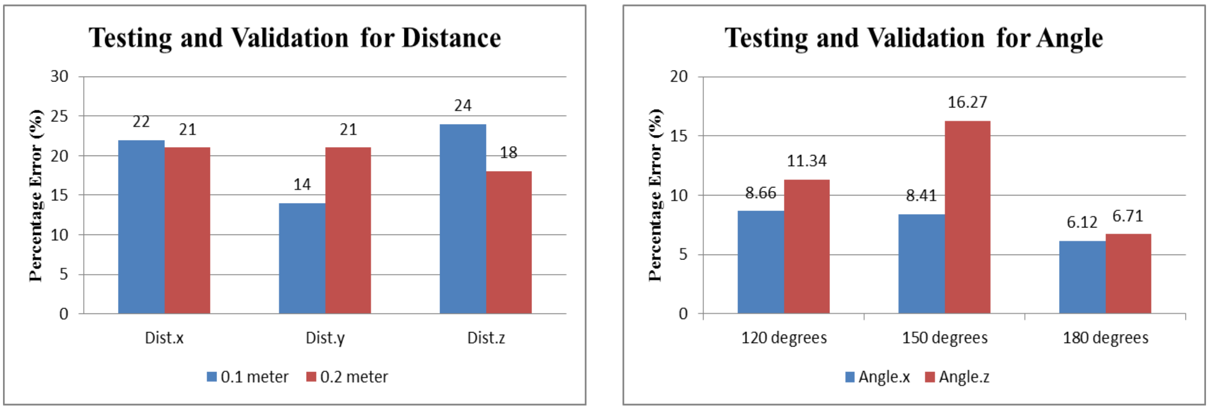
The data had been classified by using the k-NN. k-NN classifier is a type of instance-based learning that is simple to implement in the data classification. The working principle is based on the measuring the distance or similarity between the tested data and the training data. The k-value is ranged from 1 to 10 and needs to be varied in order to find the match class between the testing and training data. The original sample will randomly partition into k equal sized of subsamples. One of the subsamples will be retained as the validation data for testing of the model, while the remaining k-1 subsamples used as training data. All data will be used for both training and validation by using the k-fold cross-validation method [10, 21]. The performance of the k-NN classifier depends significantly on the distance used [22]. The distances of the Euclidean, city-block, cosine and the correlation are used.

3. Result and Discussion

The result and discussion had been divided into four parts, such as testing and validation, classification between the falling and non-falling class, sub-classification of the non-falling class, and feature selection.

3.1 Testing and Validation

There are only five features out of eight features were selected for testing and validation, which were Dist.x, Dist.y, Dist.z, Angle.x and Angle.z. However, Aareal.x, Aareal.y and Aareal.z were not selected because acceleration is calculated from the differentiation of distance with respect to time.



(a) (b)

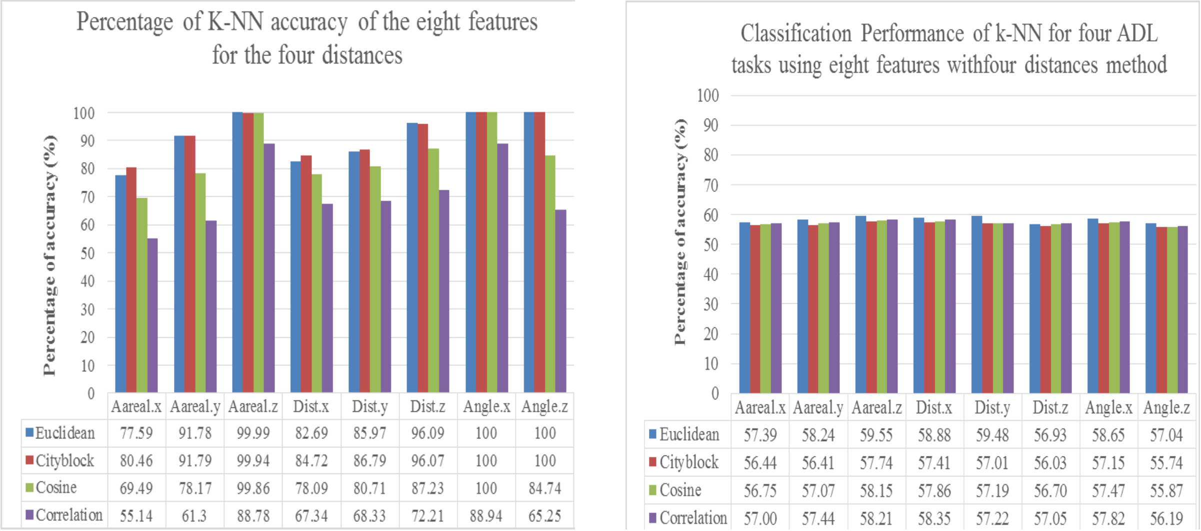
Figure 4. Testing and Validation of (a) Distance (b) Angle

The result of testing and validation of distances for the 0.1 meter and 0.2 meter is shown in Figure 4(a). There are two distances (0.1 and 0.2 meter) were used because average height of the subject is within 1.66 cm. The result of testing and validation of angles for the 120 degrees, 150 degrees and 180 degrees is shown in Figure 4(b). The 120 degrees, 150 degrees and 180 degrees are used because the values within this range represent the angle between subjects' motions with relevant axis. The results are the average of percentage error for the parameters in five trials. The percentage error of the

Angle.x is lower than the Angle.z is due to the subject was fell toward the x-axis during the frontward fall. Hence, percentage of accuracy for the Angle.x will be higher than the Angle.z.

3.2 Feature Selection

Figure 5(a) show the performance of k-NN classification between falling and non-falling of the eight features using four distances. The features of the Aareal.z (97.14%) and Angle.x (97.24%) show the highest percentage of accuracy among the total four distances. This is because the Aareal.z is affected by the height of the subject while the Angle.x is affected by the angle between the subject's motion axis and the x-axis. The value of the Aareal.z and the Angle.x in the falling class is different in polarity with the value of the Aareal.z and the Angle.x in the non-falling class. Hence, the difference of distance measured between these two classes is large and the percentage of accuracy is high. As the result, Aareal.z and the Angle.x are the best features among the eight features.



(a) (b)

Figure 5. k-NN classification performance four distances for each eight features. (a) Falling vs. Non-falling classification (b) Four activities daily living (ADL) classification

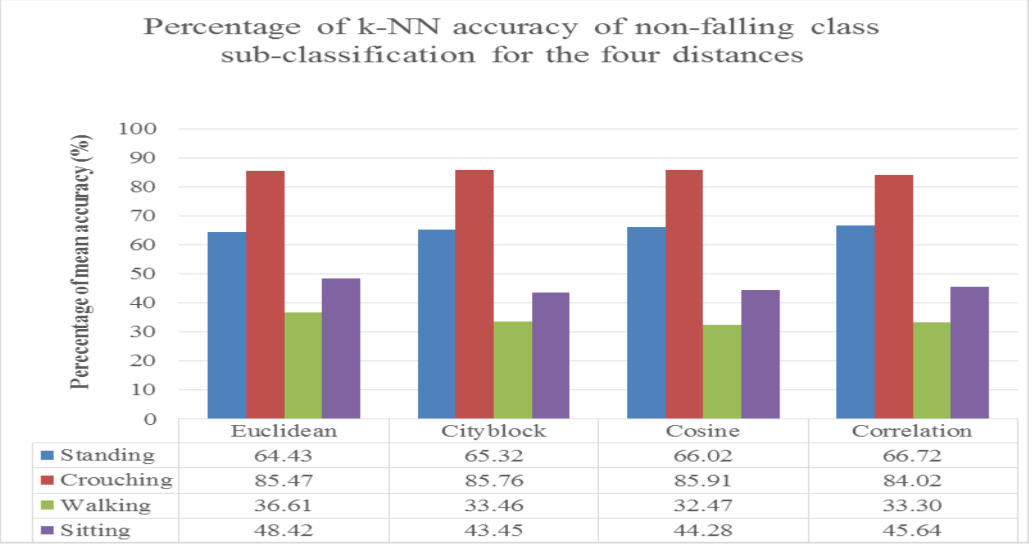
Based on Figure 5(b), the average accuracy for four distances of features Aareal.z (58.41%) and the Angle.x (57.78%) is the highest among the others eight features. This is because the polarity of the Aareal.z and Angle.x is same in four ADL subclasses. Hence, the difference in distance measured between the input data and the dataset is relatively small. The accuracy for four ADL subclasses is lower as compared with the classification between falling and non-falling class. This is because the feature value of the dataset in the falling and non-falling class is big in different. Thus, the dataset is easily in classification and high percentage of accuracy is resulted. In conclusion, the features of the Aareal.z and Angle.x were selected as the best features that implemented in the Arduino hardware.

3.3 Classification between falling and non-falling

The falling and non-falling class has been classified by using the k-NN classification algorithm in Arduino hardware. There are on using Aareal.z and Angle.x. The k-fold cross-validation method is used for k-NN classifier. A total of 250 data had been folded into 10 sets of the data. The highest the k-fold, the highest the number of the training set obtained. There had obtained 250 testing data and 2250 training data, when 250 data are folded in 10-fold cross-validation. Hence, it aims to optimize the percentage of accuracy by reducing the percentage of error [23],[24]. Data was classified based on the k-value (1 to 10). If the k-value was set at 10, the effect of the noise was reduced during the data classification. The four different types of the distance were applied which produces 100% accuracy.

3.4 Sub-classification of non-falling class

Based on the Figure 6, the percentage of mean accuracy of the sub-classification of non-falling class for the four distances with k-NN classifier is shown. The data for the sub-class is the average of the 10 k-value. The percentage of accuracy of the standing sub-class for the correlation distance is the highest (66.72%). Then, the percentage of accuracy of the crouching (85.91%) and walking (36.61%) sub-class for the cosine distance is the highest. The percentage of accuracy of the sitting sub-class for the Euclidean distance is the highest (48.42%). Different type of distance is suitable for different sub-class. Hence, the suitability of the sub-class with the distance is indicated by the percentage of accuracy. The Euclidean distance is the best computation method because the overall percentage of accuracy is the highest (58.73%) among the distances.



	Euclidean	City block	Cosine	Correlation
🟦 Standing	64.43	65.32	66.02	66.72
🟥 Crouching	85.47	85.76	85.91	84.02
🟩 Walking	36.61	33.46	32.47	33.30
🟪 Sitting	48.42	43.45	44.28	45.64

Figure 6. Percentage of accuracy of the sub-classification of non-falling class for the four distances with k-NN classifier.

4. Conclusion

In conclusion, this project aims to develop an IoT fall detection system by using wearable sensor. The IoT fall detection system had be developed with the microcontroller board (Arduino Mega 2560), IMU sensor (Gy-521 mpu6050) and WI-FI module (ESP8266-01). The parameters of IoT fall detection system include tri-axial distance and tri-axial angle had been tested and validated by calculated the percentage of error. The percentage error for the distances in x, y and z-axis are less than 25%, while the percentage error for the angle in x and z-axis is less than 20%. Thus, the IoT fall detection device is a good device that is capable to detect the fall with high accuracy. The dataset had been constructed through the data collection of the 10 UniMAP students. Eight features had been extracted from the dataset and classified through the k-NN. The features of the Aareal.z and Angle.x were selected based on the good performance during the classification. The data of the falling and daily living activity had been differentiated through the classification of the falling and non-falling. Then, the percentage of accuracy for the k-NN classifier by using k-values (1-10) and distances (Euclidean, city-block, cosine and correlation) are 100%. The non-falling class had been further classified into the sub-classes of standing, crouching, walking and sitting.

References

[1] Bouldin E L, Andresen E M, Dunton N E, Simon M, Waters T M, Liu M, Daniels M J, Mion L C and Shorr R I 2013 Falls among adult patients hospitalized in the United States: prevalence and trends *J Patient Saf* **9(1)** 13-17.

[2] Dykes P C, Carroll D L, Hurley A, et al. 2010 Fall Prevention in Acute Care Hospitals: A Randomized Trial *JAMA* **304(17)** 1912–1918.

[3] Campbell D 2014 NHS staff shortages pose risk to patients, warns watchdog | Society | The Guardian https://www.theguardian.com/society/2014/oct/17/nhs-staff-shortages-risk-patients-cqc-report.

[4] Jacobson R 2015 Widespread Understaffing of Nurses Increases Risk to Patients - Scientific American https://www.scientificamerican.com/article/widespread-understaffing-of-nurses-increases-risk-to-patients/.

[5] Chin L C, Basah S N, Affandi M, Shah M N, Yaacob S, Juan Y E and Din M Y 2017 Home-Based Ankle Rehabilitation System: Literature Review and Evaluation *Jurnal Teknologi* **79(6)** 1-21.

[6] Chin L C, Basah S N, Yaacob S, Juan Y E and Kadir A K A 2015 Camera Systems in Human Motion Analysis for Biomedical Applications *AIP Conference Proceedings* **1660(1)** 090006.

[7] Shanyu C, Chin L C, Basah S N and Azizan A F 2019 Development of Assessment System for Spine Curvature Angle Measurement *Proceedings of the 2019 8th International Conference on Software and Computer Applications (ICSCA '19), ACM* 397-402.

[8] Delahoz Y S and Labrador M A 2014 Survey on fall detection and fall prevention using wearable and external sensors *Sensors (Switzerland)* **14(10)** 19806–19842.

[9] Todd C and Skelton D 2004 What are the main risk factors for falls among older people and what are the most effective interventions to prevent these falls? *Copenhagen, WHO Regional Office for Europe (Health Evidence Network report)* http://www.euro.who.int/document/E82552.pdf , accessed 29 June 2019.

[10] Fook C Y, Hariharan M, Yaacob S, and Adom A H 2012 Malay speech recognition in normal and noise condition *IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA 2012)* 409–412.

[11] Juang H S and Lurrr K Y 2013 Design and control of a two-wheel self-balancing robot using the arduino microcontroller board *IEEE Int. Conf. Control Autom. ICCA* 634–639.

[12] Sharma M L, Kumar S, Mehta N 2017 Smart Home System Using IOT *International Research Journal of Engineering and Technology (IRJET)* **4(11)** 1108–1112.

[13] Ilapakurti A, Vuppalapati J S, Kedari S, Kedari S, Chauhan C, and Vuppalapati C 2017 IDispenser-big data enabled intelligent dispenser *3rd IEEE International Conference Proceeding of Big Data Comput. Serv. Appl. BigDataService* 124–130.

[14] Datasheet E 2004 ESP8266 Serial Esp-01 WIFI Wireless *ESP8266 Ser. Esp-01 WIFI Wirel.* https://components101.com/sites/default/files/component_datasheet/ESP8266%20Datasheet.pdf, accessed 30 June 2019.

[15] Chin L C, Basah S N, Ali M A, Fook C Y 2018 Wearable Posture Identification System for Good Sitting Position *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* **10** 135-140.

[16] Gadget Gangster 2017 Accelerometer & Gyro Tutorial: 3 Steps http://www.instructables.com/id/Accelerometer-Gyro-Tutorial, accessed 30 June 2019.

[17] Nancy Hall 2017 Aircraft Rotations https://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html, accessed 30 June 2019.

[18] Riedesel P 2016 Tilt sensing with low-cost inertial measurement units (IMUs) : Sensor calibration, accuracy specifications and application range http://urn.kb.se/resolve?urn=urn:nbn:se:hig:diva-22497, accessed 12 July 2019.

[19] Choi J, Boyle D K, Dunton J, and Staggs N 2013 Safety &amp; Outcomes *J. Nurs. Adm. Med. Care J. Nurs. Care Qual.* **43(271)** 586–591.

[20] Stratify Labs 2013 A FIFO Buffer Implementation https://stratifylabs.co/embedded design tips/2013/10/02/Tips-A-FIFO-Buffer-Implementation, accessed 13 July 2019.

[21] Aijuan S, Guangyuan S I, and Qiongchan G U 2016 Study on Remote Medical Monitoring System Based on MSP430 AND CC2530 *2016 Chinese Control and Decision Conference (CCDC), Yinchuan* 2415–2418.

[22] Dangeti P 2017 Statistics for Machine Learning *Packt Publishing Ltd.* https://books.google.com/books?id=C-dDDwAAQBAJ, accessed 13 July 2019.

[23] Prasath V B S, Alfeilat H A A, Lasassmeh O and Hassanat A B A 2017 Distance and similarity measures effect on the performance of k-nearest neighbor classifier—a review. arXiv preprint arXiv:1708.04321

[24] Ghosh A K 2006 On optimum choice of k in nearest neighbor classification *Comput. Stat. Data Anal* **50(11)** 3113–3123.