# REAL-TIME OF A MACHINE LEARNING BASED GAIT EVENT DETECTION

abdalla ussama abdalla

*Mechatronics enigneering*
*International Islamic University Malaysia (IIUM)*

Kuala Lumpur, Malaysia.

abdalla.ussama@live.iium.edu.my, gmaamk48@gmail.com

*Muhammad Afif Bin Husman*

*Mechatronics enigneering*
*International Islamic University Malaysia (IIUM)*

Kuala Lumpur, Malaysia.

afifhusman@iium.edu.my

***Abstract*— Gait analysis is a technique used in many different fields and it may be a good way to indicate some disease which effecting the walking pattern. In addition, gait analysis can be used as an input for walking aid and prosthetic control. Of particular interest in gait analysis is identifying the terrain a subject is walking onto. This paper describes the use of an inertial sensor to identify in real-time whether a subject is walking in two different terrains, namely the ramp and the stairs, in both ascending and descending motions. The identification of ramp and stairs ascent or descent pattern is carried using the Arduino and a single Inertial Measurement Unit IMU sensor which outputs the acceleration and angular velocity profile of the walking in three different axes. A signal conditioning circuit is designed to collect the required dataset to be processed and this will serve as an input for a machine learning algorithm which will be applied to identify the gait pattern in real-time.**

## I. INTRODUCTION

Event refer to something abnormal action happening frequently in normal situation; therefore, event detection is the way to find the abnormal action based on set of data related to this event. In this paper the event to be detected is the change in the walking style for ramp and stairs ascent and descent.

There are two main categories for the different gait analysis methods namely, Semi-Subjective Analysis Techniques and Objective Techniques.

The gait analysis techniques and methods focusing on the measurement for the different gait parameter which are presented in the table1which is showing the different parameters for the gait analysis in three different fields ((Muro-de-la-Herran et al., 2014)) [5]

TABLE 1

| Gait Parameter | Application | | |
|---|---|---|---|
| | Clinical | Sports | Recognition |
| Stride velocity | X | X | X |
| Step length | X | X | X |
| Stride length | X | X | X |
| Cadence | X | X | X |
| Step Width | X | X | X |
| Step Angle | X | X | X |
| Step time | X | | |
| Swing time | X | | |
| Stance time | X | | |
| Traversed distance | X | X | |
| Gait autonomy | X | | |
| Stop duration | X | | |
| Existence of tremors | X | | |
| Fall | X | | |
| Accumulated altitude | X | X | |
| Route | X | X | |
| Gait phases | X | X | X |
| Body segment orientation | X | X | |
| Ground Reaction Forces | X | X | |
| Joint angles | X | X | |
| Muscle force | X | X | |
| Momentum | X | X | |
| Body posture (inclination, symmetry) | X | X | X |
| Long-term monitoring of gait | X | X | |

Semi-Subjective Analysis: consist of analyzing the gait in clinical environment and it is carried by specialist, by observing the patient's gait parameter and the analysis is while the patient is walking, the most of this most of this analysis techniques is using primitive and rudimentary methods.

A good example for semi-subjective analysis are timed 25-foot walk, POMA, GARS "Gait Abnormality Rating Scale"

The second method is "Objective Analysis" which is based on the data analysis using the different set of tools and devices to capture and collect the data of the different parameter for the gait and using the computer ability to create the necessary model to analysis those parameters, this model is the reference to compare the different set of data to it.

According to (Muro-de-la-Herran et al., 2014)[5] there are three main methods for the objective analysis, namely floor sensor, image processing (IP) based method and wearable sensor method.

A real-time gait identifier had been used in some of the projects as the algorithm created by [6] Zhou et al which aims to detect the gait events on three walking terrains in real-time by obtain the analysis of acceleration jerk signals the used sensors. (Zhou et al., 2016)

On the other hand ,H.F et al used gyroscopes sensor to create a real-time algorithm to detect the gait events and gait phase to be used to optimize the performance of the lower limb prosthetic controller (H.F. et al., 2017)[3]

The algorithm used in those two real-time event detection project is the threshold method, however the aim of this paper to use WS method with the aid of machine learning algorithms such as KNN(K-nearest neighbor), SVM (support vector machine), random forest, and deep learning to obtain the real-time gait pattern identifier.[1][2]

## II. METHODOLGY

There are three stage in order to complete the scope of work of this paper:

- Stage one: Date acquisition and the creation of the dataset
- Stage two: Chosen the necessary gait parameter to be analyzed offline and Creating an algorithm for identify the gait pattern
- Stage three: implementing the algorithm for real-time identification

### *A. Stage one*

By using the objective gait analyses methods and specifically the (WS) wearable sensors method, a device can be created to use IMU sensor to obtain the necessary data from the gait, therefore we are using the Arduino with IMU sensor to get the data and the Arduino in the same time is connected to SD card shied which allow the Arduino to save the date from the IMU to CSV file.

By using the low level programming syntax for the Arduino, it will be faster to access the register of the sensor and obtain the date with high sample rate to get higher accuracy since the high sample rate will make very good representation for the gait in short amount of time. the code for data acquisition will be uploaded to the Arduino, and it will be attached to external power sources.

At this point the Arduino is ready to be attached to the subjects to get the gait parameter through the IMU sensor, so each gait from the subjects will be saved on CSV file on the SD card which will be easy to work with the data in this format to preprocess the data with python language or excel to isolate the unnecessary information from the data set and organize the dataset as well.

### *B. Stage two*

The IMU sensor will obtain six variables for each reading which are three variables for the three axis from the gyroscope and three variables for the accelerometer, there will be some of these variables which will not be changing that much during the gait due to the position of the dive as well as the gait pattern itself , yet this data will be a guild line for the classification algorithms to score higher accuracy as it will be shown in the result part .

In this paper we will use the (ML) machine learning as the event detection algorithm, we are interested in the classification algorithm from the (ML) which will detect the gait pattern, there are different type of the classification algorithm in the ML such as (KNN) k-nearest neighbor, and (SVM) support vector machine. Therefore, it will be better to test the different types of classification algorithm in order to see the highest accuracy one to be implemented later in the stage three.

The classification algorithms are supervised learning algorithms which mean that both the features also known as the dataset and the targeted values which is also known as the labels of the dataset should be given to the classifier as the input to the classifier in order to train it.

In this case the dataset are the values form the IMU and the targeted values is the gait pattern so for each dataset from the 25 will have one label as the gait pattern and the 25 dataset will be melting in one lager dataset with the labels. the role of sum in order to test the accuracy of the classifier is to divide the dataset in two subsets, the taring dataset and the testing dataset with ratio of 80% to 20% which mean in our case 20 to 5 the first 20 datasets will be used to train the classifier and the last 5 to test the accuracy of it. after choosing the classification algorithms with the higher accuracy the second stage should be done.

The size of the dataset increased as it will be explant in next section.

### *C. Stage three*

The To implement the classifier as real-time event detection algorithm, a mobile app will be used to upload the classifier to it, and by using the Arduino with IMU we cathe gait data will be obtained and sent to the mobile app with the uploaded classifier to classify the pattern for different type of gaits. To send the data from the Arduino to the mobile app a Bluetooth or Wi-Fi module and contacted it to the Arduino.

## III. DATA ACQUISITION PROCEDURE .

as it had been mentioned in the previous section, the dataset should be collected by cratering an Arduino device which it should be wearable. in this part the focus will be on how the device create as well as the procedure followed to collected the data.

### *A. Circuit design*

The Arduino is used to obtain the data from the IMU sensor to do that the device should be portable, which mean the external power source is needed as well as a Bluetooth module to control and send the required commend to the Arduino when the serial compunction with the PC is not connected to the Arduino. Therefore, the circuit contain five main prats which are:

- the embedded system: in this case I used Arduino Nano version of the Arduino,
- 7.3V lithium battery or 9 volte battery as external power supply which is portable and it will give the necessary voltage to operate the circuit
- MPU sensor: which is the IMU sensor used to obtain the gait parameters, and it as 6 DOF and 2 sensors embedded to it, namely the gyroscope and accelerometer.
- HC-06 Bluetooth module: used to send the command to the Arduino to start taking the reading.

- Micro SD card adapter: since the device is not connected to any PC the data is saved on SD card to be imported later for the analysis, to do so, the SD card shied is used to write the sensor data on the SD card attached to it.

The schematic diagram of the circuit is shown in Fig 1

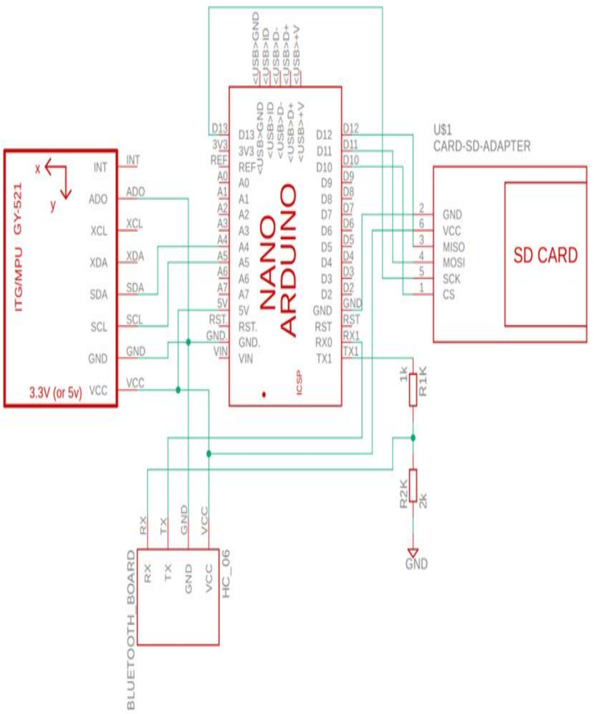


Fig 1

PCB seems like a good solution to create an efficient configuration as a wearable device and to ensure that there will not be any disconnectivity issues between the wires of the different components in the circuit., to create the PCB board the, these steps had been followed:

- The first step in creating the PCB is to create the layout of the circuit using Eagle software.
- Second step is preparing the board by removing the oxide layer of copper (board) using a rubber and clean the board to avoid any dust remains on it.
- The third step is to laminating the board for 180s to be ready for UV exposure which will print the circuit on the film layer.
- The next step is to use the developer solution to remove the unprinted film, after that the board will be ready for etching process.
- The etching process is to print the circuit on the copper layer by placing the board in the etching machine for 10-15 minutes and by using stripper solution we will remove all the film in the board.
- The following step is to drill the board in the pins locations for each component and place the components on the board.
- The last process is to create the cover for the device as well as the belt to make the device wearable, therefore to create the cover we had used 3D printer to print the cover in the specified demission for the board.

The final outcome is shown Fig 2

Fig 2

### *B. Experimental Procedure*

The dataset includes 5 subjects for each subject there will be 5 gait patterns, namely the normal walking, as either a ramp or stairs ascent or descent pattern, for each pattern the gait pattern will be obtain by using the IMU sensor 3 times.
The device during the collection of the dataset was located on the hand of the subject and fixed his rest position during the data reading. Table 2 shows the characteristic of the subjects

TABLE 2

| | weight (KG) | height (CM) | age | gender |
|---|---|---|---|---|
| subject 1 | 63 | 173 | 20 | male |
| subject 2 | 68 | 177 | 21 | male |
| subject 3 | 75 | 181 | 29 | male |
| subject 4 | 117 | 194 | 24 | male |
| subject 5 | 108 | 181 | 23 | male |

The dataset should be should be accurate and unbiased therefore, there are procedure to be followed to achieve those two objectives:
•Make sure the Y axis arrow in IMU sensor is pointing to the up of the subject and the X axis arrow is pointing to the right of the subject to ensure that all of the subject is having the same reading for the same axis. Fig3 Y and X arrows on the IMU sensor.

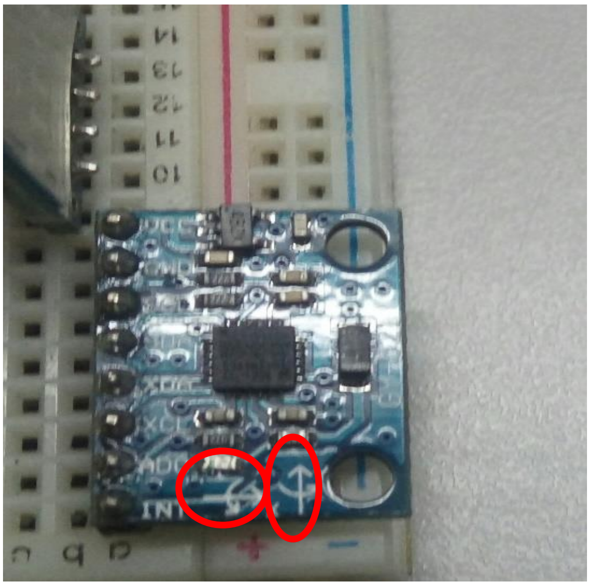

Fig 3

•The Arduino code allow the user to enter the number of reading from the sensor therefore, each gait pattern will have the same number of reading which is 1500 which it need 15 S to obtained.
•Each pattern will have 3 different reading set to increase the dataset size.
•Each of 1500 can be dived equally to 3 sets of 500 reading which it will be still capture the walking pattern phases.
•The final dataset contains 36 record (file) for normal walk, 45 record (file) for ramp ascent, 42 record (file) for ramp decent, 42 record (file) for stair ascent ,39 record (file) for stair descent, and the total number of files is 204 files.

Table 3 shows the header of the created dataset.

TABLE 3

| Time | Gyro (deg) X | Gyro (deg)Y | Gyro (deg)Z | Accel (g)X | Accel (g) Y | Accel (g) Z |
|---|---|---|---|---|---|---|
| 19:18:00.01 | 2.12 | 4.54 | -3.93 | -0.03 | -0.2 | 1.16 |
| 19:18:00.01 | -29.54 | -11.92 | 9.96 | 0.2 | -0.19 | 1.01 |
| 19:18:00.01 | 14.08 | 31.4 | 14.63 | 0.17 | -0.17 | 0.99 |
| 19:18:01.01 | 6.82 | 16.28 | 1.61 | 0.14 | -0.08 | 1.15 |

## IV. Result And Discussion

### A. *Classification process*

The dataset created will be feed to the ML classification algorithm, the target is to increase the accuracy of the algorithm to the highest possible level before the real-time implementation. the flow graph of the classifier shown in Fig4.

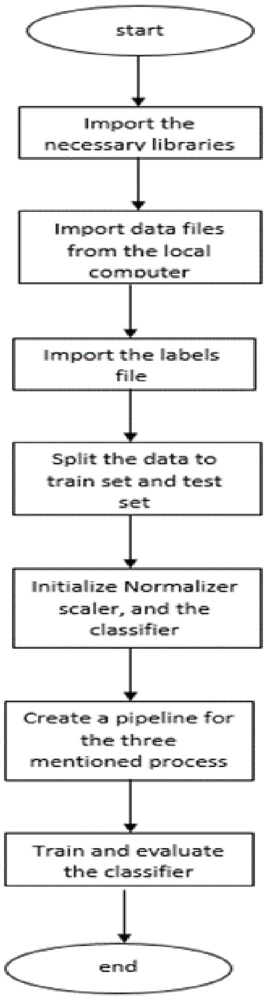


Fig.4

By fixing all the different parameters in the different steps such as the splitting process and the normalization and scaler process fixed and changing only the classifier, the following set of accuracy had been obtained

TABLE 4

| Classifier | Accuracy |
|---|---|
| Random Forest Classifier | 33.30% |
| Logistic Regression | 33.30% |
| SVC with C parameter <1 ".6" | 14.28% |
| SVC with C parameter >1 "5" | 33.30% |
| K Nearest Neighbors Classifier | 66.60% |
| Gaussian Process Classifier | 23.80% |
| Naïve Bayes | 42.80% |
| **Average accuracy** | **35.34%** |

*B. Analyzing the result for the classifier*

As can observed form the result, the highest accuracy occurred when we used KNN classifier since it is the most suitable classifier based on the size of the dataset The possible explanation as to why KNN is superior has been explained in Chapter 2 (The Literature Review) , However the average accuracy for all the classifiers is relatively low (35.34%), and can be explained by the following justification:

1. The size of the dataset itself is not that lager the total number of samples for all the five classes is 204 sample which it can be consider low-to-medium dataset therefore the classifier does not have the enough amount of data to train the classifier on it and the accuracy of the classifier will be affected negatively. Table 5 shows the scored accuracy for changing the size of the testing and training dataset and how it effects the accuracy for KNN classifier

TBALE 5

| the training dataset size to total dataset size | the testing dataset size to total dataset size | scored accuracy |
|---|---|---|
| 30.0% | 70.0% | 36.36% |
| 40.0% | 60.0% | 37.39% |
| 50.0% | 50.0% | 38.25% |
| 60.0% | 40.0% | 39.02% |
| 70.0% | 30.0% | 48.38% |
| 80.0% | 20.0% | 46.30% |
| 90.0% | 10.0% | 66.66% |

Fig .5 shows the positive increasing trend for the accuracy by increasing the size of training dataset and decreasing the size for testing dataset. This indicates how the accuracy effected by the size of the trained data.

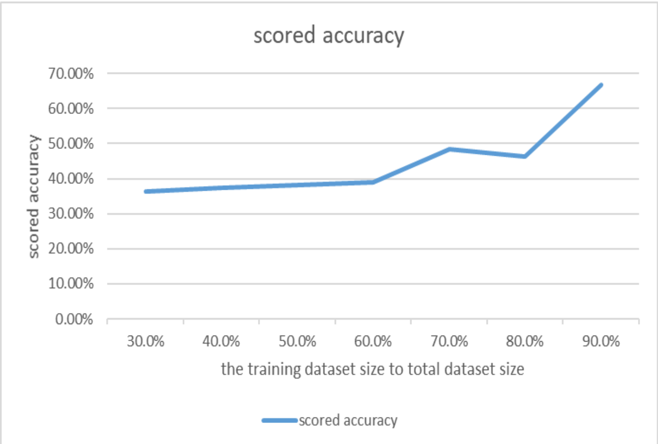


Fig 5

2. By looking into the confusion matrix for the KNN classifier shown in table 6:

TABLE 6

| | normal_walk | ramp_ascent | stair_ascent | ramp_descnt | stair_descnt |
|---|---|---|---|---|---|
| normal_walk | 4 | 1 | 0 | 0 | 0 |
| ramp_ascent | 3 | 4 | 1 | 1 | 2 |
| stair_ascent | 2 | 0 | 7 | 1 | 2 |
| ramp_descnt | 5 | 2 | 0 | 0 | 0 |
| stair_descnt | 1 | 0 | 1 | 0 | 4 |

it can be seen that the classifier has a very low accuracy for both ramp ascent and descent which is in ramp decent 0% and the reasons for that are:

- The ramp unique nature which is a combination between the normal walk and stairs in altitude changing.
- It was a difficult to fix the ramp parameters such as the angle of the ramp and the length of it for all the subject which caused this behavior

Therefore, to ensure that the problem is raised because the dataset quality and not the methodology itself, another classifier has been created for only normal walk and stair ascent to observer the quality of the classifier and the rest of the dataset. By using the same steps and same classifiers we yield the following set of accuracy as shown in Table 7.

TABLE 7

| Classifier | Accuracy |
|---|---|
| Random Forest Classifier | 75.00% |
| Logistic Regression | 56.25% |
| SVC with C parameter <1 “.6” | 62.50% |
| SVC with C parameter >1 „5” | 68.75% |
| K Nearest Neighbors Classifier | 82.00% |
| Gaussian Process Classifier | 43.75% |
| Naïve Bayes | 75.00% |
| **Average accuracy** | **66.18%** |

We can see clearly that the accuracy is doubled in most of the classifier and the average accuracy for all the classifiers increased by 31%, also by changing the other parameter such as the test set percentage to be 10% and the K parameter in the KNN classifier to 3 the obtained accuracy was almost 90% which is relatively high accuracy.

3. Another reason for this low accuracy is the demission of the dataset, the input of the classifiers is 3D input which is the number of rows for each subject, the number of columns of reading for the two sensors, and the third demission is the different walking pattern. Therefore, third demission added a

complexity to the model and effect the accuracy of the classifier.

4. The final reason for the low accuracy is the position of the device during data collection experiment which did not result in much change in the gyroscopes sensor reading, However, the variance in gyroscopes sensor reading is not that much since removing the gyroscopes sensor reading from the data set causes the accuracy recused from 66.66% to 42.45%. So, it is expected that by changing the position of the device to the knee or the ankle the accuracy could increase

*C. Improving the Classifier Accuracy*

As it had been mentioned in the previous section there are four reasons can justify the low accuracy ,However, 3 of them are caused by the way the dataset is collected ,therefore it will not be easy to deal with them ,yet ,by using some features engineering on the dataset ,the complexity of the dataset can be simplified which will lead to a better accuracy .

The first method used to simplify the dataset was (principle component analysis) PCA, The basic idea for PCA is to compressing a lot of data into something that captures the essence of the original data, the PCA takes dataset with a lot of features and represented in a smaller number of features without losing too much of the important data. However, by applying the PCA concept on the dataset, the accuracy had been affected negatively, for example the KNN classifier achieve only 42,8% for the accuracy and some classifier such as SVC scored less than 10% in the accuracy.

The second approach was to represent the dataset itself in a new form and apply same statistical operation on it to create a new dataset to be feed to the classifier. Therefore, rather the dataset will have 1500 row and six columns, it can be represented in only one row and 12 columns which contains the mean and the variance for each axis for both the gyroscope and accelerometer. Table 8 shows the header of the new data set.

TABLE 8.1

| | Gyro **X mean** | Gyro **y mean** | Gyro **z mean** | **acc x mean** | **acc y mean** | **acc z mean** |
|---|---|---|---|---|---|---|
| **0** | -2.5 | 3.1 | -1.0 | 0.0 | 0.1 | 1.0 |
| **1** | -2.9 | 2.4 | -1.3 | 0.0 | 0.1 | 1.0 |
| **2** | -1.8 | 2.4 | -0.2 | 0.0 | 0.1 | 1.0 |
| **3** | -1.8 | 1.6 | -12.1 | 0.0 | 0.2 | 1.0 |
| **4** | -3.7 | 2.2 | 0.1 | 0.0 | 0.2 | 1.0 |
| **5** | -2.8 | 2.4 | -1.6 | 0.1 | 0.2 | 1.0 |
| **6** | -3.2 | 3.6 | -0.6 | 0.1 | 0.1 | 1.0 |

TABLE8.2

| | Gyro **X var** | Gyro **y var** | Gyro **z var** | **acc x var** | **acc y var** | **acc z var** |
|---|---|---|---|---|---|---|
| **0** | 53.499 | 81.416 | 44.430 | 0.003 | 0.002 | 0.004 |
| **1** | 124.507 | 105.426 | 74.652 | 0.005 | 0.003 | 0.009 |
| **2** | 144.778 | 160.422 | 105.747 | 0.005 | 0.005 | 0.019 |
| **3** | 59.967 | 225.098 | 76.404 | 0.004 | 0.005 | 0.006 |
| **4** | 136.860 | 134.603 | 134.600 | 0.004 | 0.005 | 0.013 |
| **5** | 126.737 | 114.807 | 101.269 | 0.005 | 0.007 | 0.014 |
| **6** | 65.512 | 269.443 | 41.516 | 0.003 | 0.004 | 0.002 |

**Gyro: gyroscope, acc: accelerometer, var: variance**

The new dataset can be feed directly to the classier by using the same approach induced in the classifier flow graph table 9, the accuracy was the following:

TABLE9

| Classifier | Accuracy |
|---|---|
| Random Forest Classifier | 54.54% |
| Logistic Regression | 42.82% |
| SVC with C parameter <1 ".6" | 26.8% |
| SVC with C parameter >1 "5" | 38.09% |
| K Nearest Neighbors Classifier | 72.72% |
| Gaussian Process Classifier | 76.6% |
| Naïve Bayes | 47.6% |
| **Average accuracy** | **51.31%** |

The accuracy had been improved significantly comparing to the accuracy obtained before demission reduction prosses.

*D. Real -Time Implmention Overview*

There are too many options on how the real time implementation can be achieved however as the flow graph shows in this paper the processing of the data as well as the classification process will occur by developing mobile application, and there are a several reasons way the method had been adopted, the reasons as the following:

- The mobile app is an easy be developed and allow a friendly user experience.
- The classification model and the preprocessing of the data before it can feed to the classifier are computationally cheap, since the classifier had been trained before it can be deployed as a mobile app.

- Comparing to the embed system namely, Arduino and raspberry pi, the mobile phones are easier to work with it.
- This method opens an opportunity, to use the created device in stage one to send the data as well as recording it in the same time, which will be very useful to increase the dataset size in the future.
- Xamarin tool allow the user to create a mobile app which can work on an Android based system as well as IOS systems.

### *E. The Steps To Implement The Real Time Solution*

The first sept to implement the real time solution is to create the classifier model which it had been done in stage two and the classifier had the following catachrestic

- The type: KNN classifier with K parameter = 5
- The preprocessing: normalization and z-score transformation (Standard Scaler).
- Scored accuracy: 72.72% with 10% of the dataset used as testing set.
- The number of the features are 12 as shown in tables 8 and the number of classes are five namely "normal walk", "ramp ascent", "stair ascent", "ramp descant", "stair descant".

After the classifier had been crated, trained and evaluated, it will be converted to ONXX format and saved in this format, ONNX is an open format to represent, machine leering and deep learning models. The reason why the ONNX format is used because the classifier created by python programing language, However, C# programing language will be used to create the functionality of the Xamarin application ,therefore , the ONNX is a good choose since it is readable by both python and C# also it will save the time to create and train the classifier again by using C# . The next step is to create the Xamarin application, Xamarin is a framework which is used to develop a cross platform mobile application using C# to intrudes the functionality to the application. the functionality in this app are quit sample which are the following:

1. Load the classifier as OXXN format to the app
2. Allow the user to connect to the device created in stage one to receiver the reading from the sensors.
3. Apply the same statistical operation used to reduce the demission in section *C*
4. Feed the processed data to the classifier
5. Show the predicted class as well as the confident probability.

The final step after the app is created is to load it to the mobile phone to test it. The fig.6 shows the app after it scan for near Bluetooth devices to connect with it

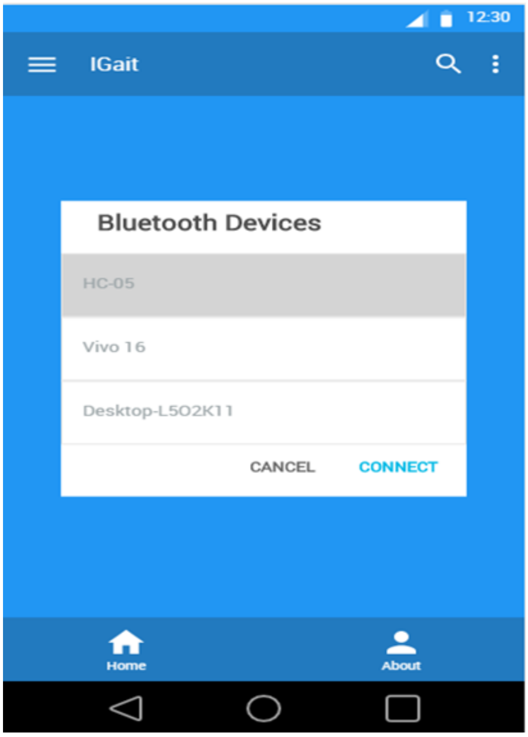


Fig.6

The fig. 7 shows the app after it had predicted the class as well as the confident probability for the data received from the Arduino device which is connected to the IMU sensor via the Bluetooth module.

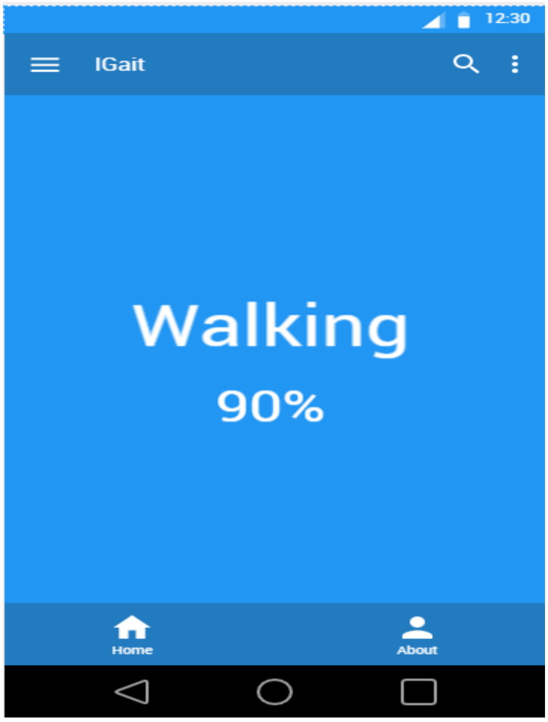


Fig 7

## V. CONCLUSION

### A. summary

The gait analysis is wildly used in many different filed such as the clinical field to identify the psychical problem for the patient as well as the sport field, however by creating an intelligent system to diagnose the gait and analyses it will be more efferent to detect an unusual gait pattern and report it, and it also can be integrated with controller systems in the smart artificial legs to increase the efficiency of it and many other applications which are out of the scope of this paper however it can be easily integrated with it in a way or another .

This paper is a new way to look how to analyses the gait parameter by using machine learning for the WS such as IMU sensor.

There are two paths this paper introduces the first one is to continue working on same project therefore the target will be to increase the accuracy of the classifier as well as to introduce a new walking patterns or new activity to the classifier.

The second path is to work on the same applied concept (time series classification) with ML or deep learning algorithm based with different application beside the gait analysis.

### B. Recombination And Suggestion

The accuracy of the used classifier in real -time implementation was relatively good (72.72%), the accuracy is based on the behavior of the classifier as well as the quality of created dataset which it will be feed to the classifier. Therefore, there are a space for further achieved be different approaches such as the following:

1. Improve the dataset: the dataset can be improved by increasing the size of the dataset which will allow the classifier to increase the training dataset which it will lead to higher accuracy , also the parameter of the ramp can be enhanced in a way such that the overfitting of the model can be avoided yet it can be easy to classify and finally the location of the sensors and be changed so that the gyroscope reading have much variance .
2. Changing the classification algorithm: the behavior of the classifier is very different from one algorithm to another, even though, the accuracy and the quality of the classifier deepening on the quality of the dataset itself, the classification algorithm play an importin in the accuracy of the classifier. Therefore, different ML classification algorithm can score higher accuracy. Also, a deep learning algorithm can be very useful in this case, yet the dataset should be lager to obtain good performance.
3. Preprocessing and simplification of the data: as it had been shown the good preprocessing of the data and simplification of the data without losing any necessary information will result in high accuracy.

Therefore, it is highly recommended that any one will continue the work on this topic or another topic with similar concept (time series classification) to the mention methods to improve the accuracy of the classifier used if it based on ML or deep learning algorithm.

## REFERENCES


[1] Bell, J. (2016). Machine Learning Hands-On For Developers and Technicalrofessionals.SantePubliqu(Vol.28). https://doi.org/10.1017/CBO9781107415324.004

[2] Fielding, A. H. (1999). An introduction to machine learning methods. Machine Learning Methods for Ecological Applications. https://doi.org/10.1007/978-1-4615-5289-5_1

[3] H.F., M., M.A.B., H., M.I., A., A., A., N., I., & http://orcid.org/0000-0003-3193-4984, D.-S. A. A. A. O.-M. H. F. . O. (2017). A Real-Time Gait Event Detection for Lower Limb Prosthesis Control and Evaluation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*(9), 1500–1509. https://doi.org/http://dx.doi.org/10.1109/TNSRE.2016.2636367

[4] Hoflinger, F., Muller, J., Zhang, R., Reindl, L. M., & Burgard, W. (2013). A wireless micro inertial measurement unit (IMU). *IEEE Transactions on Instrumentation and Measurement*, *62*(9), 2583–2595. https://doi.org/10.1109/TIM.2013.2255977

[5] Muro-de-la-Herran, A., García-Zapirain, B., & Méndez-Zorrilla, A. (2014). Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications. Sensors (Switzerland), 14(2), 3362–3394. https://doi.org/10.3390/s140203362

[6] Zhou, H., Ji, N., Samuel, O. W., Cao, Y., Zhao, Z., Chen, S., & Li, G. (2016). Towards real-time detection of gait events on different terrains using time-frequency analysis and peak heuristics algorithm. *Sensors (Switzerland)*, *16*(10), 1–17. https://doi.org/10.3390/s16101634