Basketball Activity Recognition Using Supervised Machine Learning Implemented on Tizen OS Smartwatch

Rosa Andrie Asmara [1], Nofrian Deny Hendrawan [2], Anik Nur Handayani [3], Kohei Arai [4]
[1] Politeknik Negeri Malang, Rosa.Andrie@polinema.ac.id, Malang and 65141, Indonesia
[2] Merdeka University of Malang, Nofrian.Hendrawan@unmer.ac.id, Malang and 65146, Indonesia
[3] State University of Malang, Malang and 65145, Indonesia
[4] Saga University, Sagashi Honjo Machi and 840-0027, Japan
Email: [3] aniknur.ft@um.ac.id, [4] arai@cc.saga-u.ac.jp

ARTICLE INFO

Article history:

Received March 24, 2022
Revised September 28, 2022
Accepted December 03, 2022

Keywords:

Smartwatch;
Basketball Activity Recognition;
Machine Learning;
Wearable Device

ABSTRACT

Basketball Activity Recognition (BAR) in sports teams, especially in basketball, to make statistical analysis of player activity data is currently a very important thing. BAR is one part of sports science that recognizes the movement of players in each activity, such as dribbling, passing, etc. Sport science in the sports business is used as one of the factors of coaches and management to determine strategy, starter line-up, check the condition of players after injury, etc. the current technology to recognize player activity only depends on the object detection method of players' through video recordings of players is considered lacking because it only sees the perspective of the coach to reduce players as starter line-up and there is no logical calculation of why players are not installed as starter line-up. One method for recognizing player activity is using a wearable device that has an accelerometer and gyroscope sensor with high accuracy. The values from those sensors will be classified and recognize their activity, i.e., Dribbling, Passing, and Shooting. Smartwatch is one of those wearable devices that meet those criteria. For the activity classification process, the use of the K-NN classification method is the most appropriate because it has a low computational level that is in accordance with the smartwatch specifications. The results of the classification using accelerometer sensor data and gyroscopes with K-NN as an activity recognition method have an accuracy of 81.62%, and player activity recognition applications using accelerometer and gyroscope sensors can also record the results of player movements for further analysis by management and coaches. This is the advantage of this BAR application compared to the recognition of player activity using object detection on video recordings.

Corresponding Author:

Rosa Andrie Asmara, Politeknik Negeri Malang, Jl. Soekarno Hatta No.09, Malang and 65141, Indonesia
Email: rosa.andrie@polinema.ac.id

1. INTRODUCTION

Currently, the overview of human activities in the field of sports is quite a trend for researchers engaged in the Internet of Things (IoT). Along with the development of wearable device technology that is increasingly sophisticated, it has a fairly high sensor sensitivity[1][2][3][4][5][6][7]. Apart from the factors mentioned above, it is also due to the need for current sports teams to use Artificial Intelligence as their management analysis material consideration from management before they buy new players, looking at the athlete's performance after an injury to determine whether the athlete could return to their original performance when they were not injured, etc. this is clearly a fundamental thing for sports team [8][9][10].

The majority of sports teams are aware of the importance of technology-based sports science for all sectors of their business analysis. One of the most important assets is their players, who are the spearhead to determine whether their business is going well [11]. The factor of sports science knowledge today is one of the main factors for sports teams for why teams are not optimal in matches. In this case, we will make highlight the sports team's weakness in terms of their analysis teams, where sports teams are not only in terms of players but also in terms of their analysis teams which will be used as a consideration for the coach to design what strategies are needed in the match. The analysis team on a sports team plays a very central role, yet, among the many objectives or problems they analyzed, quite a few analytical teams analyzed the percentage of movement of each player before the game or after the game dynamically [12][13][14].

Among the many sports that currently available, we chose basketball to be used as the research topic and what is the focus of the research. Also, we chose the method for introducing the activities of the players and making an activity recognition application from the method used. Referring to previous research, namely the use of homemade wearable devices, which have weaknesses in terms of system stability, sensor accuracy is quite low, etc. Another alternative is to identify video recordings of players and then use a neural network to identify player activities. However, in terms of accuracy or recognizing activities, it is not as sensitive as using a wearable device and also could not get the results of changes in player movement dynamically with time [15][16][17].

The use of wearable devices for activity recognition is indeed the best method among other methods [18]. One of the most widely used wearable devices today is the smartwatch. Smart-watches currently have 2 main sensors, namely an accelerometer and gyroscope, to detect user movement. These two sensors will later be used as attribute training data to the introduction of basketball activities, but embedding a neural network in smartwatches is currently quite difficult considering the limited computational capacity [19]. Therefore, the use of smart-watches in the introduction of human activities must be assisted by a third application to process sensor data, such as the cloud, so that sensor data processing is not too burdensome for the performance of the smart-watch. One of the smart-watch that currently has open-source features that can access JSON files (support firebase) is a smart-watch with Tizen operating system. With the feature to communicate with the cloud, it means that the smartwatch could also display recognition data from the cloud.

This paper makes the following arrangements on the structure of the article: Section 2: The main point of this research is to analyze the concept of introducing basketball activities to the x, y, and z angles of the accelerometer and gyroscope, the use of machine learning classification methods, application design application performance results evaluation with the activity recognition method using machine learning. At the same time, this section also analyzes the concept of introducing basketball activities to movement sensors.

Section 3: Based on the analysis in Section 2, the concept of this basketball activity recognition application is to use 2 movement sensors. At the same time, in this section, we make a method introduction in detail by explaining the use of machine learning methods, cloud delivery methods, data retrieval methods from the cloud, what cloud platform to use, etc. Section 4: Based on the analysis in Section 3, the application design on this smartwatch uses the backend and front end of the application. At the same time, this section also discusses the design of activity recognition applications. Section 5: To obtain a method of classifying activities with high accuracy, this section will also compare the classification method by taking sensor sampling data which is then trained. At the same time, this section will also evaluate the results of running the activity recognition application.

2. BASKETBALL ACTIVITY RECOGNITION

2.1. Movement Sensors

For the activity recognition method using a wearable device, the main thing to note down is the sensitivity of the position sensor embedded in the wearable device. The position sensors used here are an accelerometer and gyroscope, as shown in Fig. 1.

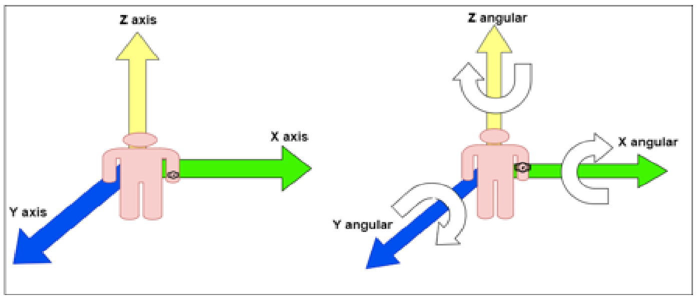


Fig. 1. Accelerometer vector and Gyroscope angular

In processing motion sensor of data (accelerometer), it is also found that roll, yaw, and pitch have a direct effect on the linear movement angle of the accelerometer. A yaw is a counterclockwise rotation of α about the $\mathcal{Z}$-axis [20][21][22]. The rotation matrix is given by

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

Note that the upper left entries of $R_z(\alpha)$ form a 2D rotation applied to x and y coordinates, whereas the $\mathcal{Z}$ coordinate remains constant. A pitch is a counterclockwise rotation of β about the $-y$ axis. The rotation matrix is given by

$$R_y(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \tag{2}$$

A roll is a counterclockwise rotation of γ about the $\mathcal{X}$-axis. The rotation matrix is given by

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix} \tag{3}$$

For example, the yaw matrix, $R_z(\alpha)$ essentially performs a 2D rotation with respect to the x and the y coordinates while leaving the $\mathcal{Z}$ coordinate unchanged [23]. Thus, the third row and third column of $R_z(\alpha)$ look like part of the identity matrix, while the upper right portion of $R_z(\alpha)$ looks like the 2D rotation matrix. The yaw, pitch, and roll rotations can be used to place a 3D body in any orientation. A single rotation matrix can be formed by multiplying the yaw, pitch, and roll rotation matrices to obtain

$$R(\alpha,\beta,\gamma) = R_z(\alpha)R_y(\beta)R_x(\gamma) \tag{4}$$

It is important to note that $R(\alpha,\beta,\gamma)$ performs the roll first, then the pitch, and finally, the yaw. If the order of these operations is changed, a different rotation matrix would be shown as a result [24]. So, we were carefully interpreting the rotations. Consider the final rotation, yaw by α.

2.2. Basketball Activity Recognition Concept

The basketball activities that we recognize here are passing, dribbling, and shooting activities. Based on the concept of calculating the movement sensor using the accelerometer and gyroscope, each activity is influenced by several features resulting from the sensor data above. The following is the effect of sensor features on the movement of basketball activities.

a. Dribble Activity

Among the other basketball activities, dribbling (Fig. 2) has its own characteristics because dribbling refers to the y-axis, whether the accelerometer sensor or gyroscope.

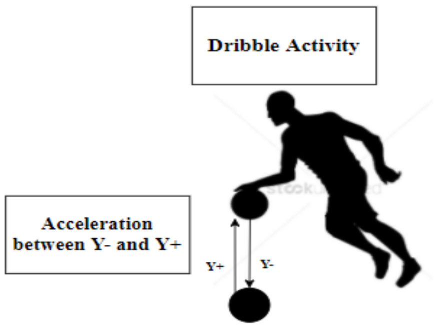


Fig. 2. Dribble Activity

Based on Fig. 2, the dribbling activity is strongly influenced by the timestamp between the y+ and y-vectors. Likewise, the pitch of the accelerometer comes from the y-axis equator angular of the gyroscope. For the timestamp itself, we use interpolation between dribbling activities with the highest timestamp as a reference, and the low timestamp is interpolated according to the highest timestamp with the highest timestamp as a reference and the lowest timestamp interpolated according to the highest timestamp.

b. Passing

Passing activities (Fig. 3) have similarities in the recognition process to shooting activities because passing and shooting activities are very much related to the x-axis on the accelerometer and angular x-gyroscope.

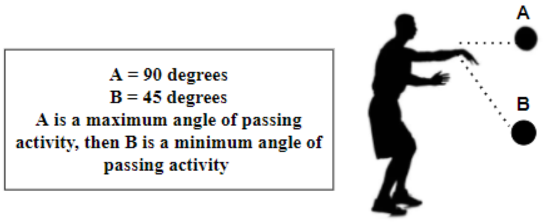

A = 90 degrees
B = 45 degrees
A is a maximum angle of passing activity, then B is a minimum angle of passing activity
A
B

Fig. 3. Passing Activity

Based on Fig. 3, passing activity is strongly influenced by the angular angle x on the gyroscope. In this study, we limit the maximum passing angle to around 90 degrees and the minimum passing angle to around 45 degrees.

c. Shooting

Shooting activities (Fig. 4) have similarities with passing activities, except that in this study, we used shooting activities > 90 degrees angle on the angular x gyroscope.

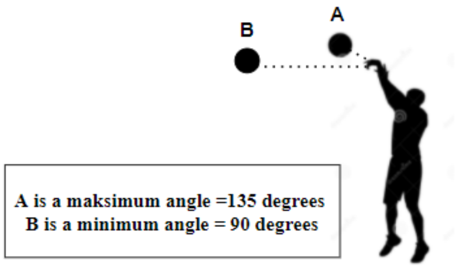

B
A
A is a maksimum angle =135 degrees
B is a minimum angle = 90 degrees

Fig. 4. Shooting Activity

Based on Fig. 4, shooting activity is also influenced by the angular angle x gyroscope, in which, in this study, we use a minimum angle of > 90 degrees and a maximum of 135 degrees. To add value to the validation of the dataset created, we take the data for shooting activities more than passing and can increase the accuracy of the dataset created for the introduction of activities later.

3. METHOD

3.1. Firebase Real-Time Database

Firebase Real-time Database is a database that is hosted through the cloud. Data is stored and executed in the form of JSON and synchronized in real-time to each connected user. These functions are to make it easier for you to manage a database on a fairly large scale [25][26]. When you create a cross-platform / multi-platform application using the Android SDK, IOS, or JS (JavaScript), all users will share an instance real-time database and receive data updates simultaneously and automatically. Another ability of the Firebase Real-time database is to remain responsive even when it is offline because the SDK Firebase Real-time database stores data directly to Disk Device or local memory. After the device is connected again to the internet, the user device (user) will receive any changes that occur.

3.2. Firebase Machine Learning

Firebase Machine Learning is a feature that creates an environment data training on communication devices in Firebase. In this case, the device sends sensor data to the real-time database and subsequently processes and detraining in Firebase Machine Learning to classify players' activities. Next, the results of the training data from Firebase Machine Learning gave feedback to the connected device.

3.3. K – Nearest Neighbors

Due to the low level of computing capability of the Smart Watch, the Machine Learning method was selected as a method for the introduction of activities and selecting classification methods. One of the most widely used classification methods is K-Nearest Neighbor (KNN). KNN is a supervised method which means it requires training data to classify objects that are closest to [27]. The working principle of K Nearest Neighbors is looking for the closest distance between the data to be evaluated to neighboring (Neighbor) in training data. In the training process, documents are grouped manually in accordance with the specified categories. After that, the document will go through the pre-processing stage, which will produce weight for each word in all training documents. Then, calculate the resemblance of the test document vector with each training document that has been classified. To find out the similarity of documents, used the Cosine Similarity method. This method can be used to interpret the distance of each document based on the similarity of documents. The calculation of distances with the Cosine Similarity method can be seen in Equation (5).

$$(d) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5}$$

Sorting the distance based on the smallest (closest) value to the largest (farthest). Then, determine the number of neighbors (values K) that we want to use as a reference for the classification process. From this K value, the output category is determined based on the nearest Euclidean value.

3.4. Naïve Bayes

Naïve Bayes Classifier is a classification method that is rooted in the Bayes theorem. The main characteristic of the Naïve Bayes Classifier is a very strong assumption (Naïve) will be the independence of each condition/incidence [28]. Before explaining the Naïve Bayes Classifier, it will be explained in advance the Bayes theorem, which is the basis of the method. In the Bayes theorem, if there are two separate events (for example, A and B), the Bayes theorem is formulated as follows:

$$P(A|B) = \frac{P(A)}{P(B)} P(B|A) \tag{6}$$

The formula above explains that the probability of entering a sample of certain characteristics in class C (Posterior) is the probability of the appearance of class C (before the inclusion of the sample, often called a prior) multiplied by the probability of the occurrence of the sample characteristics in class C (also called likelihood), divided by the probability emergence of sample characteristics globally (also called evidence). Therefore, the above formula can also be written as follows:

$$e = \frac{i \times u}{v} \tag{7}$$

Evidence value is always fixed for each class in one sample. The value of the posterior will be compared with the value of the posterior value of other classes to determine to what class a sample will be classified. Further elaboration of the Bayes equation describes (A|B1,…,Bn) use the multiplication rule as follows:

$$P(A|B1, \dots, B_n) = P(A)P(B1, \dots, B_n|A) = P(A)P(B1|A)(B2, \dots, B_n|A, B1) \tag{8}$$

It can be seen that the result of the elaboration causes more and more complex conditional factors that affect the probability value, which is almost impossible to analyze one by one. As a result, the calculation becomes difficult to do. This is where the assumption of very high (naive) independence is used, that each clue is independent of the others. Equation (9) is a model of Naive Bayes Theorema, which will then be used in the classification process. For classification with continuous data, the Gaussian Density formula is used as follows:

$$P = (X_i = x_i | Y_i = y_i) = \frac{1}{\sqrt{2\pi\sigma ij}} e^{-\frac{(x_i - u_{ij})^2}{2\sigma^2 ij}} \tag{9}$$

$$\mu = \frac{1}{n} \sum_{i=0}^{n} x_i \tag{10}$$

$$\sigma = \left[\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)^2 \right]^{0.5} \tag{11}$$

3.5. Support Vector Machine

Fig. 5. shows some patterns that are members of two classes: +1 and –1. The patterns belonging to class -1 are symbolized by red (squares), while patterns in class +1 are symbolized by yellow (circles). Classification problems can be translated by trying to find a line (hyperplane) that separates the two groups. The best separating hyperplane between the two classes can be found by measuring the hyperplane margin. And find the maximum point. Margin is the distance between the hyperplane and the closest pattern of each class. This closest pattern is called a support vector. The solid line in the image shows the best hyperplane, which is located right in the middle of the two classes, while the red and yellow dots in the black circle are support vectors [29]. The efforts to find the location of this hyperplane are the core of the learning process in Support Vector Machine (SVM).

The hyperplane (decision boundary), like in Fig. 6, is the best separating way between the two classes that can be found by measuring the margin of the hyperplane and finding its maximum point. Margin is the distance between the hyperplane and the closest data from each class. This closest data is called the support vector. Since there are so many decision boundaries that can not be modeled by a linear form or equation, we must model the decision boundary as a non-linear function. The support vector classifier extension is a support vector machine that uses kernel techniques. A kernel function transforms data into another space (usually a higher dimension). This transformed data (at higher dimensions), we expect to be separated by a linear function. Suppose we look back at the original dimension, the decision boundary in the new dimension models a non-linear decision boundary in the original dimension. This is illustrated in Fig. 7.

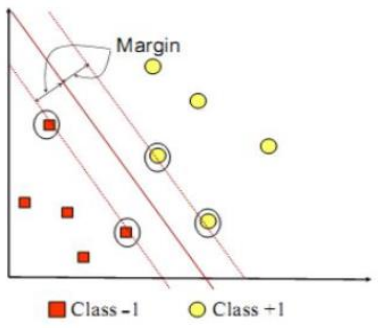


Fig. 5. Hyperplane Between 2 Class for Support Vector Classifier

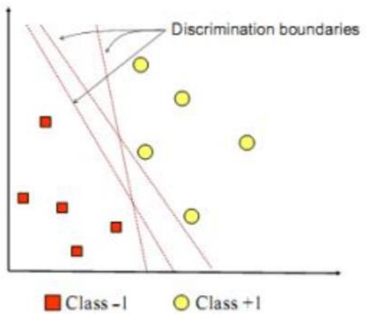


Fig. 6. Perfect Hyperplane for Support Vector Classifier

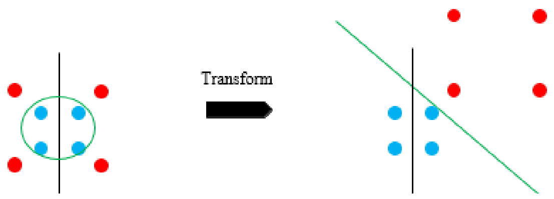


Fig. 7. Transform Illustration data of Support Vector Machine

There are many functions of this kernel, some of which are well-known, such as the Polynomial Kernel is

$$k(x_i, x_j) = (x_i x_j + 1)^d \quad (12)$$

The Radial Basis Function Kernel is

$$k(x, y) = exp - \frac{|x - y|^2}{2\sigma^2} \tag{13}$$

The difference between a support vector machine (SVM) and a support vector classifier is the integration of kernel functions into the model. Before discussing how kernel functions are integrated with SVM, let's discuss a little more about the support vector classifier. A support vector classifier that fulfills all these constraints can be represented as Equation (13), where the x' represents a new data, x_i is an instance of the training data, and N is the number of training data. The operation (x', xi) represents the inner product. How to calculate the inner product of two vectors is given in Equation (14), where the F represents the length of the vector. The inner product can be interpreted as a calculation of the similarity of two vectors.

$$f(x') = \beta_0 + \sum_{i=0}^{n} \alpha_i (x', x_i) \tag{14}$$

$$(a, b) = \sum_{j=1}^{F} \alpha_j b_j \tag{15}$$

To calculate β parameter and α in Equation (15), we need the (N/2) combination of instance pairs that exist in the training data. However, when it passes a new input to the equation, we are actually simply calculating how close (similar) the inputs are to the support vectors. This is because 0 for instances other than support vectors, etc., is set to only nonzero for support vectors. That is, the classification decision depends on the choice of support vectors.

3.6. Decision Tree

Fig. 8 is the model of the decision tree is the most widely used algorithm for classification problems. A decision tree consists of several nodes, namely the tree's root, internal nodes, and leaves. The concept of entropy is used to determine which attributes a tree will split. In the decision tree, each internal node divides the space into two or more according to a discrete function of the input value attribute. In the simplest and most frequent cases, each test assumes it has a single attribute, so the empty partitioned space is matched to the attribute value [30]. Classification using decision trees is carried out by routing from the root node to the leaf node. Decision tree algorithms include ID3, C4.5, C5.0, and CART. After the Decision Tree is built, each case is assigned to each of j leaf, which is $j = 1...N$ with wij's weight is 0 or 1 if each test attribute is known for i. At first, the entire population forms the roots of different trees, so in order to produce various tree branches, the selected features are distinguished from the population characteristics. This feature is called a test which generates a new child node. The power of discrimination can be measured by the Shannon entropy gains G. The higher the entropy of a sample, the more impure the sample is when the algorithm in this method uses the concept of entropy. The concept of Entropy is used to measure "how informative" a node is (which is usually called how good it is). Entropy (S) = 0, if all examples of S are in the same class. Entropi(S) = 1, if the number of positive samples and the number of negative samples in S are the same. 0 < Entropy (S) < 1, if the number of positive and negative samples in S are not the same.

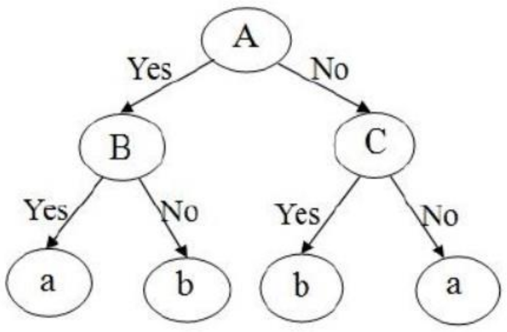


Fig. 8. The Model of Decision Tree

The formula used to calculate the entropy of sample S is

$$S = \sum_{j=1}^{n} -p_j \, log_2 \, p_j \tag{16}$$

After getting the entropy value, the attribute selection is made with the largest information gain value.

$$A = S - \sum_{i=1}^{n} \frac{|S_i|}{|S|} x \, (S_i) \tag{17}$$

4. APPLICATION AND SYSTEM DESIGN

In designing the system and application activity recognition, we use Tizen Studio Web Apps, which consists of a front end and a back end. Also, the use of other features of the cloud platform that used is machine learning firebase which is very helpful for shortening the test time without a third application, as in Fig. 9.

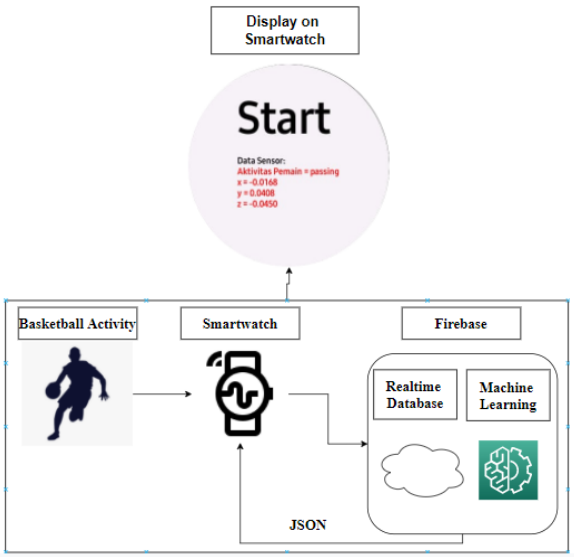


Fig. 9. Recognition Method of Basketball Activity Recognition

4.1. Back end

The Back End of Tizen studio uses the Javascript programming language. To open access from software on the smartwatch to get sensor API data, we have to register privileges on Tizen Studio by accessing the sensors that we will use, namely the accelerometer and gyroscope [31][32][33][34]. For the back end itself, we focus on sending sensor data to the firebase real-time database, and vice versa smartwatches can retrieve data from firebase to be displayed on the smartwatch display. After gaining access from the Tizen Studio sensor API from the Tizen Studio privilege system, the smart-watch can then be used to send sensor data to the Firebase Real-time Database and vice versa to download sensors from Firebase, as shown in Fig. 10.

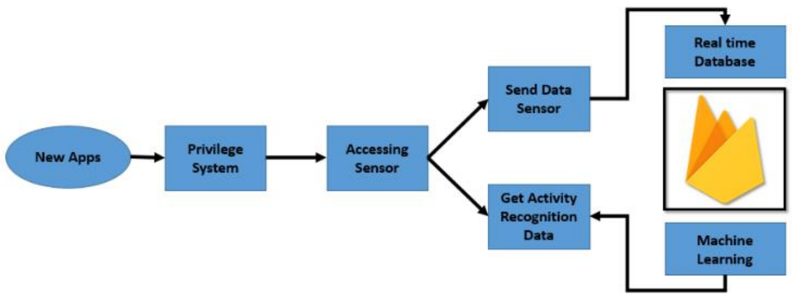


Fig. 10. Backend Framework of Basketball Activity Recognition

4.2. Front End

The Front End in designing the Basketball activity recognition application is needed to see the response time for sending data and downloading data. Therefore, in designing the front end of this application, we not only display the data from the introduction on Firebase Machine Learning, but we also display sensor data to facilitate the validation process. The main task of the front end here is to display the sensor data sent and the classification data downloaded from the cloud, as shown in Fig. 11.

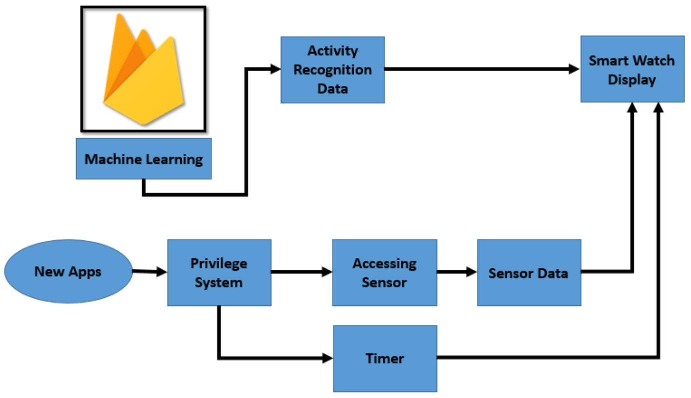


Fig. 11. Frontend Framework of Basketball Activity Recognition

5. RESULT AND DISCUSSION

In the test results in this section, we will divide the analysis into the analysis of the results of running the application and the analysis of the results of the introduction using machine learning.

5.1. Classifier Result

Before training the censor data to find which kind of classifications method that has the highest accuracy, we must first go through the design of the dataset (Fig. 12). The design of the dataset was carried out after the censor data obtained from the smart-watch with a total number of 921 iterations which were divided into 307 iterations/activities for 3.07 seconds with the specifications being made to 20 Hz for each recognized activity (passing, shooting, dribble) resulting from data selection with a high-pass filter and data interpolation to balance data for high validation values.

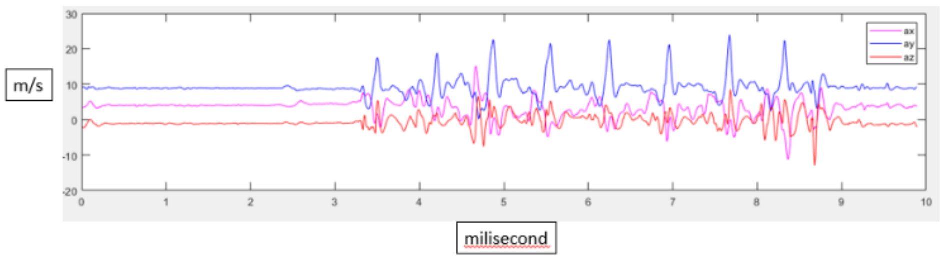


Fig. 12. Dataset in Time Domain

Referring to Fig. 12, the sensor data sent from the smartwatch to the firebase is still in the form of a time domain signal. Therefore we need to process the data from the sensor into a frequency domain form and apply high pass filtering to make it easier to select flat signals (activities that are not detected), see Fig. 13.

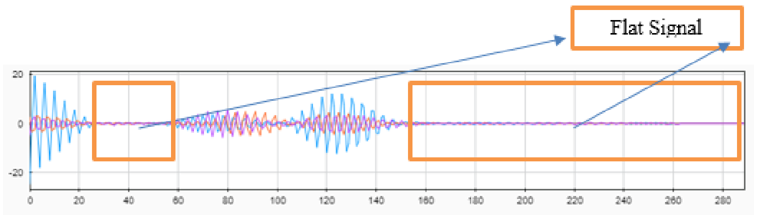


Fig. 13. Dataset after filtering using high pass

After the Flat Signal has been deleted, then we use the selected dataset as material for training data. To determine the classification method as a material for comparison in this study, we first conduct a data training process on 4 classification methods. With the training process, we can see the accuracy of the types of classification methods, in this case, KNN, Naïve Bayes, Decision Tree, and SVM. K – Nearest Neighbors result is shown in Fig. 14.

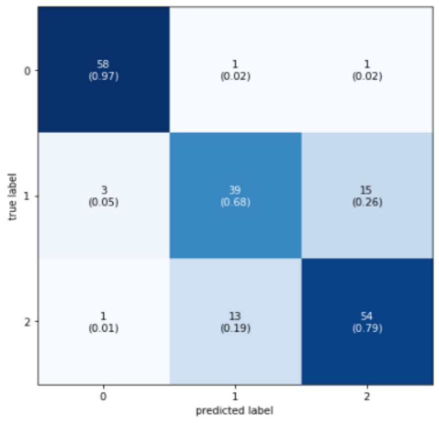


Fig. 14. Confusion Matrix of KNN

The results of the introduction of activities using the KNN method, the dribble activity, has a single accuracy better than the other two activities. For more detailed activity classification results, see Table 1.

Table 1. Result of KNN

Activity	**Recognition Result**		
	Accuracy	**Precision**	**Recall**
Dribble	95%	94%	97%
Passing	71%	74%	68%
Shooting	78%	77%	79%

For a further and complete analysis, we tested 2 features that were used for classification and the most affected by the classification process, namely the $x - axis$ to the $y - axis$ accelerometer sensor, which can be seen in Fig. 15.

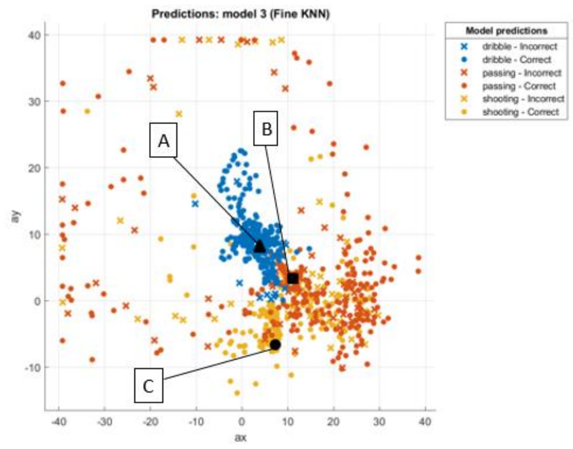


Fig. 15. Scatter Plot of KNN

From the results of the tests that have been carried out in Table 1 and the centroid analysis in Fig. 14, the dribble shooting and passing activities have low similarity values, and the incorrect data test values for passing and shooting activities are quite low. It has shown in Fig. 15. A value which was a dataset for dribbles that have a more solid circle/centroid, while B and C have centroids that were not as good as dribbles, but the total accuracy of the K – NN method was 81.62%.

Naïve Bayes result is shown in Fig. 16. The results of activity recognition using the Naïve Bayes method, dribble activity, has a single better accuracy than the other two activities. For more detailed activity classification results, see Table 2. For a further and complete analysis, we tested 2 features that were used for classification the most influenced by the classification process, namely the $x-axis$ to the $y-axis$ accelerometer sensor which can be seen in Fig. 17.

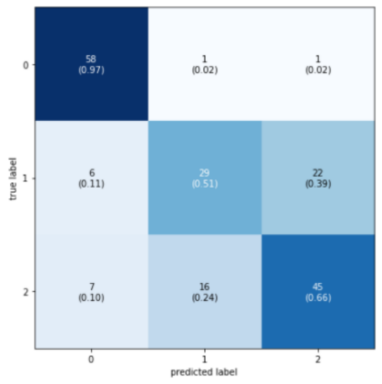


Fig. 16. ConfusionMatrix of Naïve Bayes

Table 2. Result of Naïve Bayes

Activity	Recognition Result		
	Accuracy	Precision	Recall
Dribble	89%	82%	97%
Passing	56%	63%	51%
Shooting	66%	66%	66%

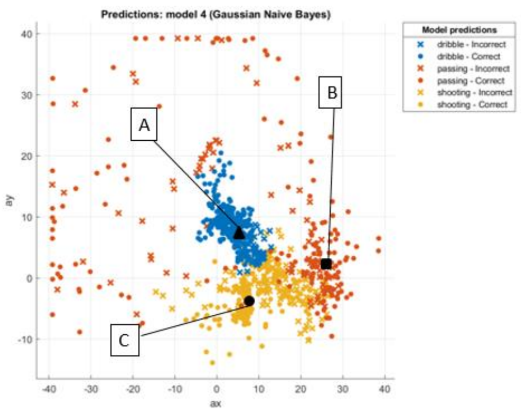


Fig. 17. Sactter Plot of Naïve Bayes

From the results of the tests that have been carried out in Table 2 and the centroid analysis in Fig. 16, the value of dribble shooting and passing activities has low similarity, but the value of incorrect data on passing and shooting activity tests was quite high. It has shown in Fig. 17. A value which was a dataset for dribbles that have a more solid circle/centroid, while B and C have centroids that were not as good as dribbles, and the total accuracy of this Naïve Bayes recognition method was 71.35%.

Table 3. Result of Support Vector Machine

Activity	Recognition Result		
	Accuracy	Precision	Recall
Dribble	94%	90%	100%
Passing	62%	60%	63%
Shooting	68%	74%	63%

Support Vector Machine result is shown in Fig. 18. The results of the introduction of activities using the Support Vector Machine method, the dribbling activities, have a single better accuracy than the other two activities. More detailed activity classification results can be seen in Table 3.

For a further and complete analysis, we tested 2 features that were used for classification the most influenced by the classification process, namely the $x - axis$ to the $y - axis$ accelerometer sensor which can be seen in Fig. 19. From the results of the tests has been carried out in Table 3 and the centroid analysis in Fig. 18, the value of dribble shooting and passing activities has low similarity, but the value of incorrect data on passing and shooting activity tests was quite high. It has shown in Fig. 19. The value of A was the dataset for dribbles that have a more solid circle/centroid, while B and C have centroids that were not as good as dribbles, and the total accuracy of the Support Vector Machine recognition method was 75.13% which was better than Naïve Bayes.

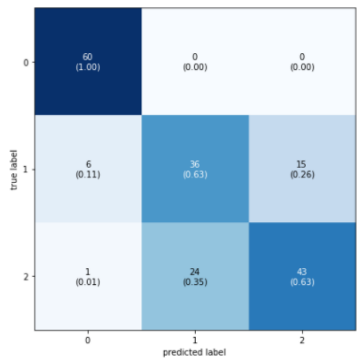


Fig. 18. Confusion Matrix of Support Vector Machine

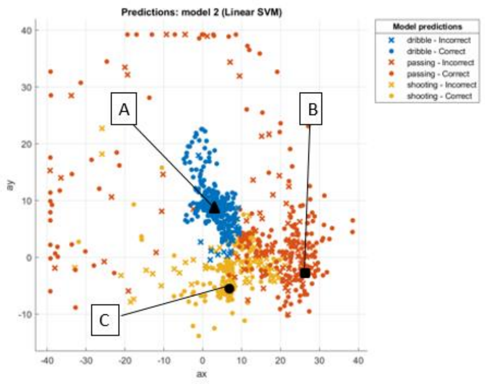


Fig. 19. Scatter Plot of Support Vector Machine

The decision tree result is shown in Fig. 20. The results of the introduction of activities using the Decision Tree method, dribbling activities, have a single accuracy better than the other two activities. More detailed activity classification results can be seen in Table 4.

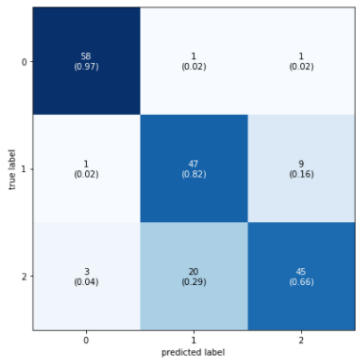


Fig. 20. Confusion Matrix of Decision Tree

Table 4. Result of Decision Tree

Activity	**Recognition Result**		
	Accuracy	**Precision**	**Recall**
Dribble	95%	94%	97%
Passing	75%	69%	82%
Shooting	73%	82%	66%

For a further and complete analysis, we tested 2 features that were used for classification and the most influenced by the classification process, namely the $x-axis$ to the $y-axis$ accelerometer sensor which can be seen in Fig. 21.

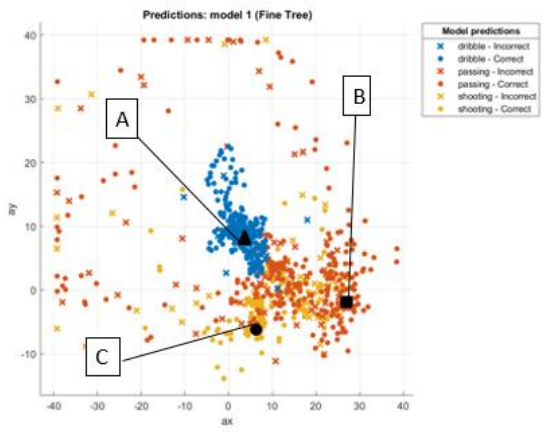


Fig. 21. Scatter Plot of Decision Tree

From the results of the tests has been carried out in Table 4 and the centroid analysis in Fig. 20, the value of dribbling shooting and passing activities has low similarity, but the value of incorrect data on passing and shooting activity tests was quite high. It has shown in Fig. 21. A value which was a dataset for dribbles that have a more solid circle/centroid, while B and C had centroids that were not as good as dribbles, and the total accuracy of this Decision Tree recognition method was 81.08% which was better than Naïve Bayes and Support Vector Machines. From the 4 methods that have been tried, we got the conclusion that K Nearest Neighbors is the highest method, which was 81.62%, and will be implemented as an activity classification method in basketball activity recognition applications.

5.2. Implementation Classifier Method to Apps

After getting the results of the classification method with the highest accuracy, namely KNN, then we try to implement it into the application that we designed earlier [35][36][37]. The Starting Display of the Apps is shown in Fig. 22.

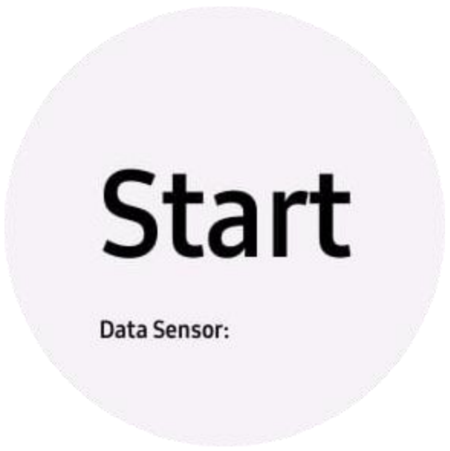
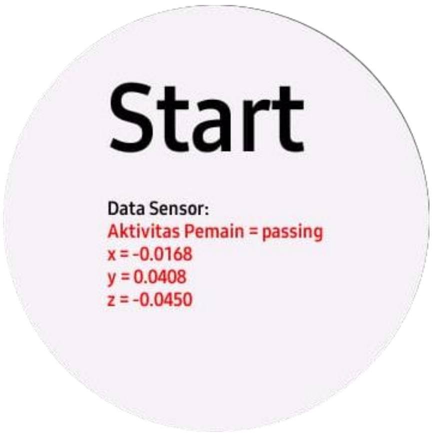


Fig. 22. Starting Display of the Apps

Here we use a simple layout to validate the recognition results from the Firebase real-time database with sensor data results. Therefore, only the recognition results, timer, and sensor data will be displayed, as shown in Fig. 23.

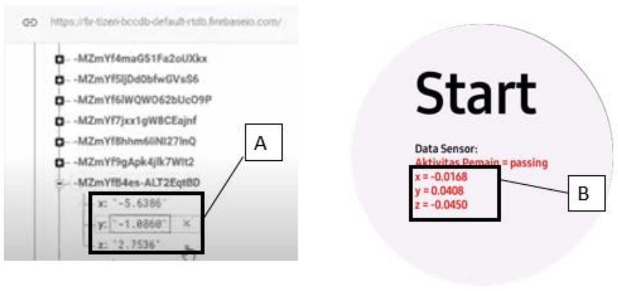


Start

Data Sensor:
Aktivitas Pemain = passing
x = -0.0168
y = 0.0408
z = -0.0450

Fig. 23. Running display of the Apps

From Fig. 23, it is clear that we got the data from the introduction of activities from the Firebase Real-Time Database and sensor data from the smart-watch directly without a firebase intermediary [38][39][40]. For details regarding the validation of sensor data results that were sent to the Real-Time Database with those downloaded from the smart-watch, we will show them in Fig. 24.



```
https://fir-tizen-bccdb-default-rtdb.firebaseio.com/
-MZmYf4maG51Fa2oUXkx
-MZmYf5IjD0d0bfwGVsS6
-MZmYf6lWQWO62bUcO9P
-MZmYf7jxx1gW8CEajnf
-MZmYf8hhm6liNt27lnQ
-MZmYf9gApk4jlk7WIt2
-MZmYfB4es-ALT2EqtBD
  x: "-5.6386"
  y: "-1.0860"
  z: "2.7536"
```

Start

Data Sensor:
Aktivitas Pemain = passing
x = -0.0168
y = 0.0408
z = -0.0450

Fig. 24. Display Data Sensor in Firebase and Smartwatch

From Fig. 24, we can compare the difference in the response time of sending data to the firebase, indicated by A, and censoring the data from the smartwatch, which was indicated by B. The difference in delay between sending data on the smart-watch and the data from the smart-watch directly has a delay of 1 iteration of data/10 Ms. After we got the delay to respond censor data value which was already displayed. Next, we can add a pop-up feature to find out that we do the activity for seconds, as shown in Fig. 25.

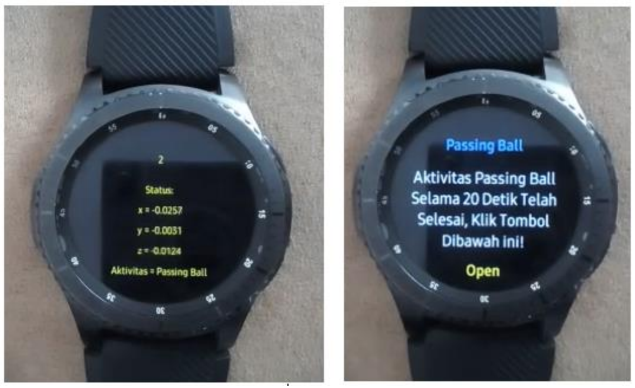


Fig. 25. Final Running Apps

6. CONCLUSION

The conclusion of this research that has been obtained from the results and discussion of data processing on smartwatches is for recognizing activities in basketball and classification methods with the highest accuracy for each activity. K-NN has the best accuracy results among others, with the average classification results for each activity equal to 81.62%.

REFERENCES

[1] R. A. Asmara, M. Ridwan, and G. Budiprasetyo, "Haar Cascade and Convolutional Neural Network Face Detection in Client-Side for Cloud Computing Face Recognition," *2021 International Conference on Electrical and Information Technology (IEIT)*, pp. 1-5, 2021. https://doi.org/10.1109/IEIT53149.2021.9587388.

[2] D. Puspitasari, Noprianto, M. A. Hendrawan, and R. A. Asmara, "Development of Smart Parking System using Internet of Things Concept," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 611-620, 2021, https://doi.org/10.11591/ijeecs.v24.i1.pp611-620.

[3] R. A. Asmara, I. Siradjuddin and N. Hendrawan, "Improving Basketball Recognition Accuracy in Samsung Gear S3 Smartwatch using Three Combination Sensors," *2020 4th International Conference on Vocational Education and Training (ICOVET)*, pp. 386-390, 2020, https://doi.org/10.1109/ICOVET50258.2020.9230342.

[4] R. A. Asmara, B. Syahputro, D. Supriyanto, and A. N. Handayani, "Prediction of Traffic Density Using YOLO Object Detection and Implemented in Raspberry Pi 3b + and Intel NCS 2," *2020 4th International Conference on Vocational Education and Training (ICOVET)*, pp. 391-395, 2020, https://doi.org/10.1109/ICOVET50258.2020.9230145.

[5] R. A. Asmara, F. Al Huda, B. S. Andoko and A. N. Handayani, "Optimized walking straight guidance system for visually impaired person that use Android smartphone," *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, pp. 332-336, 2017, https://doi.org/10.1109/SIET.2017.8304159.

[6] K. L. -M. Ang and J. K. P. Seng, "Application Specific Internet of Things (ASIoTs): Taxonomy, Applications, Use Case and Future Directions," in *IEEE Access*, vol. 7, pp. 56577-56590, 2019, https://doi.org/10.1109/ACCESS.2019.2907793.

[7] H. Lu, X. He, M. Du, X. Ruan, Y. Sun and K. Wang, "Edge QoE: Computation Offloading With Deep Reinforcement Learning for Internet of Things," in *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9255-9265, 2020, https://doi.org/10.1109/JIOT.2020.2981557.

[8] Y. Tang and D. Wang, "Optimization of Sports Fitness Management System Based on Internet of Health Things," in *IEEE Access*, vol. 8, pp. 209556-209569, 2020, https://doi.org/10.1109/ACCESS.2020.3039508.

[9] F. Al Huda, H. Tolle, and R. A. Asmara, "Realtime Online Daily Living Activity Recognition Using Head-Mounted Display," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. *11*, no. 3, pp. 67–77, 2017, https://doi.org/10.3991/ijim.v11i3.6469.

[10] R. A. Asmara, F. Al Huda and A. N. Handayani, "Design and Implementation of Blind Runner Guide Android Mobile Application for the Visual Impairment User Experience," *International Journal of Engineering & Technology*, vol. 7, no. 4.36, pp. 1295-1300, 2018, https://www.sciencepubco.com/index.php/ijet/article/view/28975.

[11] R. M. Musa *et al.*, "The application of Artificial Neural Network and k-Nearest Neighbour classification models in the scouting of high-performance archers from a selected fitness and motor skill performance parameters," *Science & Sports*, vol. 34, no. 4, pp. e241-e249, 2019, https://doi.org/10.1016/j.scispo.2019.02.006.

[12] Y. -L. Hsu, H. -C. Chang and Y. -J. Chiu, "Wearable Sport Activity Classification Based on Deep Convolutional Neural Network," in *IEEE Access*, vol. 7, pp. 170199-170212, 2019, https://doi.org/10.1109/ACCESS.2019.2955545.

[13] M. Yu, J. Jin, X. Wang, X. Yu, D. Zhan, and J. Gao, "Development and Design of Flexible Sensors Used in Pressure-Monitoring Sports Pants for Human Knee Joints," in *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25400-25408, 2021, https://doi.org/10.1109/JSEN.2021.3087005.

[14] P. Cao, G. Zhu, Q. Zhang, F. Wang, Y. Liu, and R. Mo, "Blockchain-Enabled HMM Model for Sports Performance Prediction," in *IEEE Access*, vol. 9, pp. 40255-40262, 2021, https://doi.org/10.1109/ACCESS.2021.3064261.

[15] R. Ji, "Research on Basketball Shooting Action Based on Image Feature Extraction and Machine Learning," in *IEEE Access*, vol. 8, pp. 138743-138751, 2020, https://doi.org/10.1109/ACCESS.2020.3012456.

[16] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, "TSE-CNN: A Two-Stage End-to-End CNN for Human Activity Recognition," in *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 1, pp. 292-299, 2020, https://doi.org/10.1109/JBHI.2019.2909688.

[17] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A Semisupervised Recurrent Convolutional Attention Model for Human Activity Recognition," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1747-1756, 2020, https://doi.org/10.1109/TNNLS.2019.2927224

[18] Z. Song, Z. Cao, Z. Li, J. Wang, and Y. Liu, "Inertial motion tracking on mobile and wearable devices: Recent advancements and challenges," in *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 692-705, 2021, https://doi.org/10.26599/TST.2021.9010017.

[19] H. Aksu, A. S. Uluagac and E. S. Bentley, "Identification of Wearable Devices with Bluetooth," in *IEEE Transactions on Sustainable Computing*, vol. 6, no. 2, pp. 221-230, 2021, https://doi.org/10.1109/TSUSC.2018.2808455.

[20] W. Cho, J. Suh, and S. -H. You, “Integrated Motion Control Using a Semi-Active Damper System to Improve Yaw-Roll-Pitch Motion of a Vehicle,” in *IEEE Access*, vol. 9, pp. 52464-52473, 2021, https://doi.org/10.1109/ACCESS.2021.3070366.

[21] S. -H. Mok, H. Bang, B. Koh, K. -M. Park and H. Leeghim, “Roll Steering of Yaw–Pitch Steered SAR for Reducing Ground–Target Pointing Error,” in *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 1, pp. 38-42, 2018, https://doi.org/10.1109/LGRS.2017.2772245.

[22] X. Xu, Y. Sun, X. Tian, L. Zhou and Y. Li, “A Double-EKF Orientation Estimator Decoupling Magnetometer Effects on Pitch and Roll Angles,” in *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 2055-2066, 2022, https://doi.org/10.1109/TIE.2021.3060652.

[23] H. Hsiao and J. J. Chang, “Characterization of Signal Integrity Due to Pitch–Roll–Yaw Rotation Tolerance in Magnetic Position Sensing Systems,” in *IEEE Transactions on Magnetics*, vol. 53, no. 3, pp. 1-7, 2017, https://doi.org/10.1109/TMAG.2016.2624270.

[24] Y. Sun, X. Xu, X. Tian, L. Zhou and Y. Li, “A Decoupled Orientation Estimation Approach for Robust Roll and Pitch Measurements in Magnetically Disturbed Environment,” in *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-11, 2022, https://doi.org/10.1109/TIM.2021.3135555.

[25] B. P. Prathaban, R. Balasubramanian and R. Kalpana, “A Wearable ForeSeiz Headband for Forecasting Real-Time Epileptic Seizures,” in *IEEE Sensors Journal*, vol. 21, no. 23, pp. 26892-26901, 2021, https://doi.org/10.1109/JSEN.2021.3120307.

[26] M. T. Ahammed *et al*., “Real-Time Non-Intrusive Electrical Load Classification Over IoT Using Machine Learning,” in *IEEE Access*, vol. 9, pp. 115053-115067, 2021, https://doi.org/10.1109/ACCESS.2021.3104263.

[27] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, “Efficient kNN Classification With Different Numbers of Nearest Neighbors,” in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774-1785, 2018, https://doi.org/10.1109/TNNLS.2017.2673241.

[28] Y. Shi, X. Lu, Y. Niu, and Y. Li, “Efficient Jamming Identification in Wireless Communication: Using Small Sample Data Driven Naive Bayes Classifier,” in *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1375-1379, 2021, https://doi.org/10.1109/LWC.2021.3064843.

[29] M. Bernardini, L. Romeo, P. Misericordia, and E. Frontoni, “Discovering the Type 2 Diabetes in Electronic Health Records Using the Sparse Balanced Support Vector Machine,” in *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 1, pp. 235-246, 2020, https://doi.org/10.1109/JBHI.2019.2899218.

[30] R. Rivera-Lopez and J. Canul-Reich, “Construction of Near-Optimal Axis-Parallel Decision Trees Using a Differential-Evolution-Based Approach,” in *IEEE Access*, vol. 6, pp. 5548-5563, 2018, https://doi.org/10.1109/ACCESS.2017.2788700.

[31] A. Barros, C. Ouyang and F. Wei, “Static Analysis for Improved Modularity of Procedural Web Application Programming Interfaces,” in *IEEE Access*, vol. 8, pp. 128182-128199, 2020, https://doi.org/10.1109/ACCESS.2020.3008904.

[32] B. Liu, W. Dong and Y. Zhang, “Accelerating API-Based Program Synthesis via API Usage Pattern Mining,” in *IEEE Access*, vol. 7, pp. 159162-159176, 2019, https://doi.org/10.1109/ACCESS.2019.2950232.

[33] X. Sun, C. Xu, B. Li, Y. Duan and X. Lu, “Enabling Feature Location for API Method Recommendation and Usage Location,” in *IEEE Access*, vol. 7, pp. 49872-49881, 2019, https://doi.org/10.1109/ACCESS.2019.2910732.

[34] Y. Zhou, C. Wang, X. Yan, T. Chen, S. Panichella and H. Gall, “Automatic Detection and Repair Recommendation of Directive Defects in Java API Documentation,” in *IEEE Transactions on Software Engineering*, vol. 46, no. 9, pp. 1004-1023, 2020, https://doi.org/10.1109/TSE.2018.2872971.

[35] Y. Lee and M. Song, “Using a Smartwatch to Detect Stereotyped Movements in Children With Developmental Disabilities,” in *IEEE Access*, vol. 5, pp. 5506-5514, 2017, https://doi.org/10.1109/ACCESS.2017.2689067.

[36] V. Genovese, A. Mannini and A. M. Sabatini, “A Smartwatch Step Counter for Slow and Intermittent Ambulation,” in *IEEE Access*, vol. 5, pp. 13028-13037, 2017, https://doi.org/10.1109/ACCESS.2017.2702066.

[37] C. Shen, B. -J. Ho and M. Srivastava, “MiLift: Efficient Smartwatch-Based Workout Tracking Using Automatic Segmentation,” in *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1609-1622, 1 July 2018, https://doi.org/10.1109/TMC.2017.2775641.

[38] K. Kyritsis, C. Diou, and A. Delopoulos, “Modeling Wrist Micromovements to Measure In-Meal Eating Behavior From Inertial Sensor Data,” in *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 6, pp. 2325-2334, 2019, https://doi.org/10.1109/JBHI.2019.2892011.

[39] Y. Gao *et al*., “Heart Monitor Using Flexible Capacitive ECG Electrodes,” in *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4314-4323, 2020, https://doi.org/10.1109/TIM.2019.2949320.

[40] W. -J. Li, C. Yen, Y. -S. Lin, S. -C. Tung and S. Huang, “JustIoT Internet of Things based on the Firebase real-time database,” *2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)*, pp. 43-47, 2018, https://doi.org/10.1109/SMILE.2018.8353979.