Activity Recognition using Kinect and Comparison of Supervised Learning Models for Activity Classification

Tanakon Sawanglok
Department of Computer science
Thammasat university
Pathumthani,Thailand
tanakon.saw@gmail.com

Tananya Thampairoj
Department of Computer science
Thammasat university
Pathumthani,Thailand
tananyathampairoj@gmail.com

Pokpong Songmuang
Data Science and Innovation Program
Thammasat university
Pathumthani,Thailand
pokpong@cs.tu.ac.th

***Abstract*—This work presents a method to develop activity recognition using Kinect as a motion-sensing device and supervised learning for classification. Data from Kinect are continuous and independent frame representing in three dimensional axes from 20 human joints. The data then are trained for classify activities using supervised learning algorithms. The activities are 10 basic motion-gestures such as standing, waving, Thai-style greeting and walking. To compare supervised learning for classification in the task, four algorithms including neural networks, naive bayes, decision tree and support vector machine are applied to generate classification models. From experiment results, the best overall classification model was from neural network algorithm at about 75% accuracy while the second best was support vector machine with slightly lower accuracy. From analysis, the most incorrect activities were 'wai' (Thai greeting) and 'walking' in which were often misinterpreted to their similar activities as 'bowing' and 'running', respectively.**

Keywords— Activity Recognition, Kinect, Motion Detection, Data mining, Classification

I. INTRODUCTION

An action or activity is a part of human life. It becomes important information to be studied in several fields such medical science, social science, arts, etc. Thus, capturing human motions and classifying them into a specific understandable activity becomes one of required inputs for those studies. The information is then used in applications such as automatic falling detection for elders and handicapped persons [1], detection of certain diseases [2], or imitating gesture for robots and animation. The human motions and gestures are usually obtained via sensors or camera images. For real time detection, sensors are a more viable option since they require less processing [3]. In terms of classification, supervised data-mining is one of the frequent used approaches since its accuracy is significantly higher than other [4].

Among the sensors to detect human motions and gestures, Kinect [5] from Microsoft is a famous motion-sensing device with reasonable price and wide-coverage function. It has been used in several researches [6] to input human motions through a sensing webcam. In these researches, their classifying method is mostly supervised learning methods including support vector machine (SVM), neural network (NN), naïve bayes (NB) and decision tree (DT) with acceptable accuracy for their purposed task. Although those works provide insight discussion and limitation of the method in their publication, there is no clear specification on suitable methods in a certain tasks. Their results are unfortunately difficult to be compared since their settings, purposes and datasets are apparently different.

In this paper, we aim to study on classification methods used in gesture-based activity recognition with Kinect. This study is to compare the classification models from four algorithms including SVM, NN, NB and DT and expects to find more specifications of each method for further implementation reference. The study involves in ten basic motion-gestures used in our ordinary life such as greetings and sitting. We expect that the comparative results may show us insights on pros and cons of the classification algorithms in a task of gesture-based activity recognition. The rest of this paper is structured as follows. Section II provides background details of the related knowledge and existing works. Section III explains methodology for applying Kinect to gesture-based activity recognition. Experiment results in comparing the four classification algorithms are given in Section IV. Last, Section V provides conclusions of this paper.

II. LITERATURE REVIEW

A. *Classification Algorithms*

In this section, a brief summary on supervised classification algorithms frequently used in activity recognition is provided. The algorithms include neural network (NN), support vector machine (SVM), naïve bayes (NB) and decision tree (DT). These algorithms are then used for comparison in this work.

First, Neural Network (NN) is an analysis method to generate a classification model that imitates the network of the human brain [7]. NN analyzes data of each hierarchy. All hierarchies analyze the weight of the dataset and compare with a threshold to find the path to the next hierarchy. Multilayer Perceptron Neural Network uses a Back-propagation algorithm to train the dataset.

Second, Support Vector Machine (SVM) is an algorithm used for data analysis and classification [8]. SVM main method is to find hyper-plane that is able to optimally distinguish data. The method is to put the value of the data group into feature space. SVM attempts to find the optimal line base on a parameter such as, regularization (C), gamma, and kernel.

Third, Naïve Bayes (NB) is a classification algorithm that applies from Bayes' theorem [9]. NB predicts the types of data from the probabilities learned from datasets regarding given probability.

Last, Decision Tree is a classification algorithm that trains by a set of data in finding the most important attributes for classifying datasets [10]. The method is to generate a tree diagram model that can distinguish classes by considering attribute value.

These four algorithms require labelled data as a training dataset to create a classification model. The generated models though are apparently different regarding the training method, but they are all regarded as practical and usable classification model applied in many existing applications.

B. *Microsoft Kinect sensor*

Microsoft Kinect is a motion-sensor device manufactured by Microsoft. The device as shown in Fig. 1 uses the technology developed and researched with PrimeSensor as an accessory of Xbox gaming. It allows a user to control and interact with the game console or the computer through gesture and voice commands.

Fig 1. Microsoft Kinect sensor [15]

The concept of Kinect devices is that a user is a Joy Controller. For example, when playing a tennis game, a player originally controls motions through a joystick. When using Kinect, a player apparently moves in the style of playing tennis himself instead of using only fingers for controlling. Without holding a joystick, this increases a sense of realism and engagement with gaming. Kinect's main function is to capture the movement of the player from a camera into signals processed to the computer.

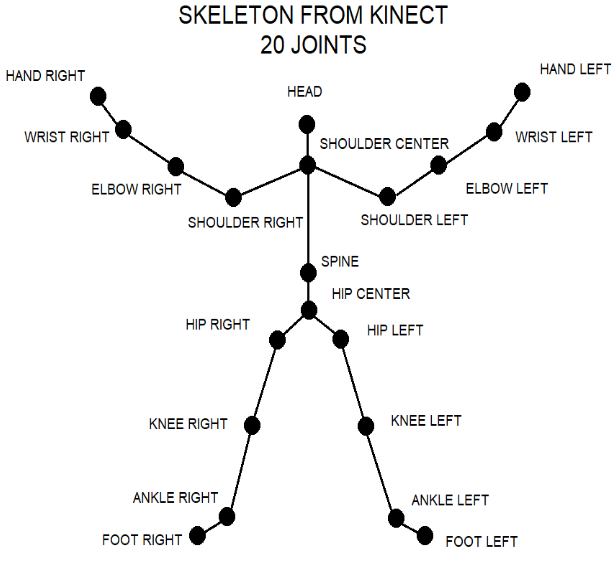


Fig 2. skeleton 20 joints

Kinect works with the infrared light from the camera in which cannot be seen with the human eyes. The projected light consists of a vertical dot of 480 points and horizontal dot of 640 points. Each point is 3 mm apart at a distance of two meters from the light source. Then, the depth sensor receives a picture of the brightness of infrared light falling on the object and sending back to Kinect measures axial depth Z (Axis-Z). If the brightness is high, the object is near. On the other hand, if the brightness is lower, the object is far away. This makes it possible to simulate the environment in three dimensions. Kinect also records player faces and can use voice in control. When the camera has a depth of image. Kinect sensors can separate users from the background environment such as wall, chair, or even the classification of the user's hand in front or back. Gesture information is processed as the appearance of the skeleton joint. Kinect analyzes the movement characteristics of the joints to realize user's motion. There are 20 joints as shown in Fig 2. The data from Kinect device are gather in form of vector (x,y,z).

C. *Related work*

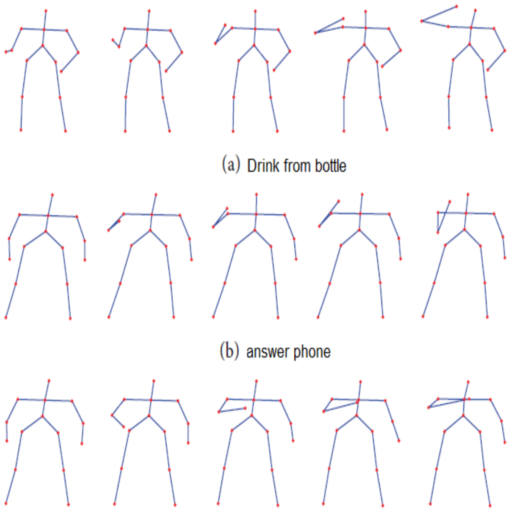
In this part, existing works on motion-based activity recognition are reviewed and presented in brief. The review focuses on their classification method and motion capturing method.

Simon Fong et al. [11] propose method to recognize human activities using Shadow features. The technique is used to improve the supervised learning efficacy of the classifier. Shadow features are inferred from the dynamics of body movements, and thereby modelling the underlying momentum of the performed activities. The system was tested by six algorithms as follow, decision tree, support vector machine, neural network, hoeffding tree, naïve Bayes, and k-nearest neighbor. The most accurate is neural network and support vector machine, decision tree, K-nearest neighbor, naïve bays and hoeffding tree respectively.

Rendy Alfuadi and Kusprasapta Mutijarsa [12] propose method in tracking three basic positions including ‘standing’, ‘sit down’ and ‘lie down’. The study uses three-machine learning with the classification method as Support vector machine, multi-layer perceptron and Naive Bayes. The most accurate is Support vector machine, multi-layer perceptron and naive Bayes, respectively.

Alessandro Manzi et al. [13] present a two-person activity recognition system. The human actions are encoded using a set of a few basic postures obtained with an unsupervised clustering approach. Multiclass support vector machines are used to build models on the training set. The system is evaluated on the Institute of Systems and Robotics (ISR) - University of Lincoln (UoL) and Stony Brook University (SBU) datasets, reaching overall accuracies of 0.87 and 0.88, respectively.

Enea Cippitelli et al. [14] studies in using SVM algorithm to recognize human activities. Activity events are split into the posture feature into a vector and slice vector to a cluster, then separate each activity with the k-means algorithm. It was tested with five public data sets and compared the experimental results of public data in the past. The testing dataset is based on open activity data. There are from five public sources as KARD, The Cornell Activity, The UTKinect, The Florence3D and MSR Action3D. Although each public data set consists of different set of activities, the gesture motions are similar as shown in Fig. 3. The results showed that their proposed method obtained better accuracy from the KARD public and the Cornell Activity data series. The results of the experiment were more than 77 percent accuracy in all datasets.



(a) Drink from bottle

(b) answer phone

(c) Read watch

Fig 3 Similar activity

III. Methodology

This study aims to compare classification algorithms for activity recognition with Kinect. The classification method for modeling activity recognition is mainly adopted from Enea Cippitelli et al [14]. Kinect is selected as our motion sensor to capture human actions. Data are split into clusters and trained for models using four different algorithms including NB, NN, SVM and DT. The activities in this study however are different from the original work. The processes in preparing data for activity recognition are as follows.

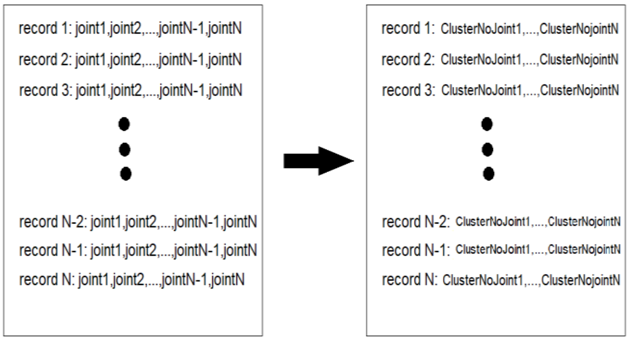
A. feature extraction

The data from the Kinect camera consist of 20 skeleton points in three axes as *x*, *y* and *z*. The data from the user's gesture are converted to the Arff file, a file format of Weka, for modeling. The motion data is non-independent and continuous. Each activity data are a group of motions from the beginning and end of the event to be labelled. The scope in this work is 10 activities as follow.

1. Right-hand waving
2. left-hand waving
3. Walking
4. Eating
5. Standing
6. Running
7. Jumping
8. Sitting
9. Bowing
10. Wai (Thai greeting gesture)

Constructing a motion data set starts with a 20-joint skeleton joint from Kinect to create a record. The column is a 20-point skeleton including target class and row is the amount of data stored in a frame. Since the motion data can not be used directly, the algorithm is calculated in a consideration that each row is independent.

However, this leads to be unable for separation between the left hand up and left handwave. So, we need to convert the posture into a sequence of posture in a row. The data sequence is generated from the k-mean model. K-mean details by replacing the posture with the cluster, respectively.



record 1: joint1,joint2,....,jointN-1,jointN
record 2: joint1,joint2,....,jointN-1,jointN
record 3: joint1,joint2,....,jointN-1,jointN

record N-2: joint1,joint2,....,jointN-1,jointN
record N-1: joint1,joint2,....,jointN-1,jointN
record N: joint1,joint2,....,jointN-1,jointN

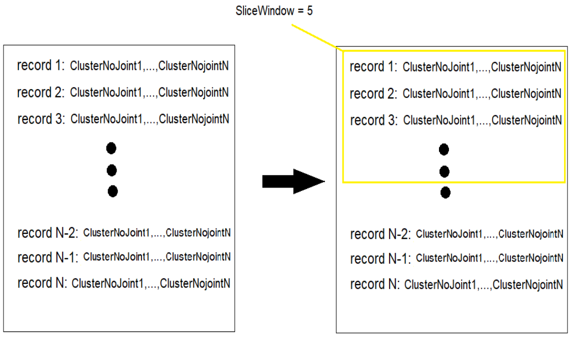
record 1: ClusterNoJoint1,....,ClusterNojointN
record 2: ClusterNoJoint1,....,ClusterNojointN
record 3: ClusterNoJoint1,....,ClusterNojointN

record N-2: ClusterNoJoint1,....,ClusterNojointN
record N-1: ClusterNoJoint1,....,ClusterNojointN
record N: ClusterNoJoint1,....,ClusterNojointN

ClusterNo = ClusterNumber.

Fig 4. transform record from raw data to cluster data

Next, removal of the duplicate cluster is performed in only stored transition cluster. For example, we modify a sequence of cluster [C1 C1 C2 C2 C3 C2 C2 C1] to [C1 C2 C3 C2 C1].



record 1: ClusterNoJoint1,....,ClusterNojointN
record 2: ClusterNoJoint1,....,ClusterNojointN
record 3: ClusterNoJoint1,....,ClusterNojointN

record N-2: ClusterNoJoint1,....,ClusterNojointN
record N-1: ClusterNoJoint1,....,ClusterNojointN
record N: ClusterNoJoint1,....,ClusterNojointN

record 1: ClusterNoJoint1,....,ClusterNojointN
record 2: ClusterNoJoint1,....,ClusterNojointN
record 3: ClusterNoJoint1,....,ClusterNojointN

record N-2: ClusterNoJoint1,....,ClusterNojointN
record N-1: ClusterNoJoint1,....,ClusterNojointN
record N: ClusterNoJoint1,....,ClusterNojointN

ClusterNo = ClusterNumber.

Fig 5. slice cluster data with slide window (5)

Then, we divide the sequence into 5 sequences. For example, the left hand motion originally is defined as [C1 C2 C3 C2 C1 C2 C1 C3 C2], and we divide it into 3 phases (A_1, A_2 and A_3, respectively) as shown in Fig 6.

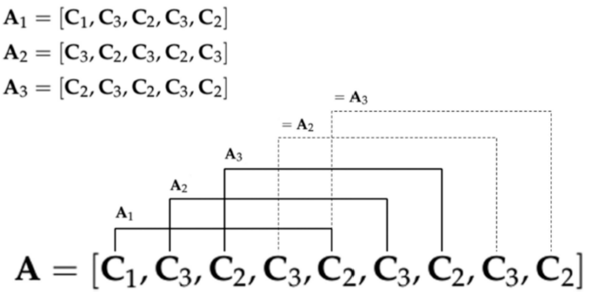

$A_1 = [C_1, C_3, C_2, C_3, C_2]$
$A_2 = [C_3, C_2, C_3, C_2, C_3]$
$A_3 = [C_2, C_3, C_2, C_3, C_2]$
$= A_3$
$= A_2$
A_3
A_2
A_1
$A = [C_1, C_3, C_2, C_3, C_2, C_3, C_2, C_3, C_2]$

Fig 6. Activity record [4]

In Fig. 6, the notation ‘A’ represents the type of motion data files that are replaced by the cluster and the slice window is set to 5. The divided motions (namely A_1, A_2 and A_3 in Fig. 6, for instance) thus are the data in each record to be trained in the model.

B. *training model*

The prepared dataset according to previously mentioned method is then used to generate a classification model of the motion recognition. To compare the performance of different classification algorithms, there are four selected algorithms including artificial neural network, Naive Bayes, Decision tree and support vector machine.

IV. Experiment

A. *Experimental method*

In this experiment, we aim to compare an accuracy of the activity recognition using different classification algorithms. The motions were captured from 6 human samples in which were 3 males and 3 females. Their heights were in between 150-180 cm. The samples were asked to perform 10 activities (mentioned in Section III.A) in front of the setting Kinect. The data were then processed following the methods.

In evaluation, 10-fold cross validation was applied to separate training data and testing data. The classification result was the activity of the user. Measurements in this experiment were precision, recall and f-measure. The models used for classification were created by Weka. Parameters of each algorithm were configured as given in Table I. Details of motion sensing setting for Kinect and K-mean parameter of features were set as shown in Table II.

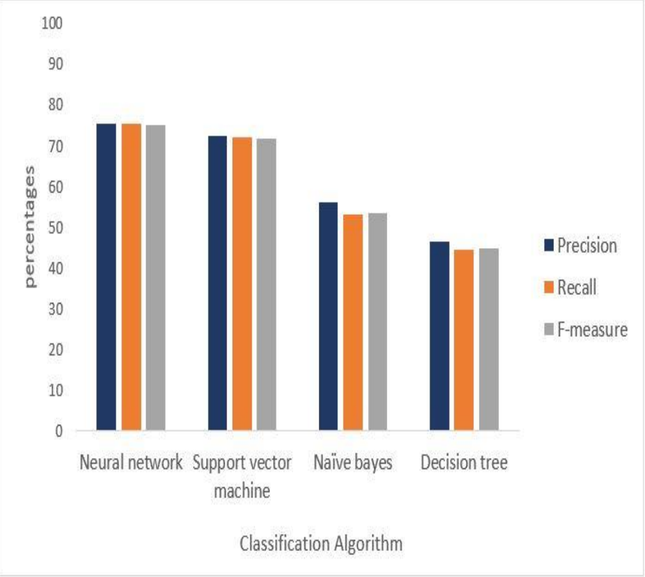
TABLE II. Motion feature

Parameter	value
X joints (Horizontal)	20
Y joints (Vertical)	20
Z joints (Depth)	20
Number of cluster	5
slice window	5
Feature	(20+20+20) *5 = 300

B. *Experimental results*

Results of classification based on precision, recall and f-measure are given in Fig 7. In a comparison, NN performed the best among four selected algorithms in both precision and recall measurements as around 75% while the second best was SVM with slightly lower results.

Fig 7. Results of Activity classification based on precision, recall and f-measure



C. *Discussion*

The experiment results show that the best algorithm was NN and SVM, respectively. However, there were more than 20% of the classification results that were incorrect. To

TABLE I. Parameter Setting for Classification Algorithms

Parameter	**Neural network**	**Support vector machine**	**Naïve Bayes**	**Decision tree**
Batch size	100			
doNotCheckCapabilities	FALSE			
Specific parameter setting	- Decay: FALSE - Hidden layer: (attribs + classes)/2 - Learning rate: 0.3 - Momentum: 0.2 - Iteration: 500	- C: 1 - Epsilon: 1.0 * 10-12 - Kernel: PolyKernel - Calibrator: Logistic - ToleranceParameter: 0.001	- NumDecimalPlaces: 2 - UseKernelEstimator: FALSE - UseSupervisedDiscretization: FALSE - displayModelInOldFormat: FALSE	- ConfidenceFactor: 0.25 - MinNumObj: 2 - NumFolds: 2 - Unpruned: FALSE - UseLaplace: FALSE

analyze the incorrect, we then observed the classification results from the 2-best models and found that there are some different incorrect results from the two models as provided in Table III. As a note, other activities that do not appear in the table are all correct.

TABLE III. STATISTICS OF THE INCORRECT RESULTS FROM NN AND SVM

Model	Activity	Incorrect Percentage	Classified Results
NN	walk	62.50%	stand,run,jump,wai
	jump	33.33%	walk,bow
	wai	66.67%	lefthandwave,stand,eat,bow
	run	33.33%	walk,stand,jump,wai
	eat	28.57%	wai,drink
	drink	28.57%	righthandwave,eat
	stand	11.11%	jump
SVM	walk	62.50%	stand,run,jump
	jump	33.33%	walk,eat,run
	wai	66.67%	stand,eat,run,bow
	run	33.33%	stand,jump,wai
	eat	42.86%	wai,drink
	bow	25.00%	stand,jump
	drink	28.57%	righthandwave,wai
	stand	11.11%	jump

The incorrect results indicate that the top incorrect activities were ‘wai’ and ‘walk’ for both models. They were more than 60% of these activities to be incorrect. The ‘wai’ gesture was often misinterpreted as ‘bow and ‘standing’, respectively. This may happen because these three gestures are similar from the standing pose but differentiate from hand movement and top body movement. Since Kinect captures these activities from the front of the samples, it can be confused among them from unclear hand motions. Similarly, the most incorrect ‘walk’ activities were also confused with ‘run’ since the motions are similar as moving actions. These issues may require extra sensing device or extra parameters to discriminate these similar activities.

Moreover, the incorrect results also reveal that NN was the only one that did not fail in classifying ‘bow’ activity while other models occasionally mistook it for other activities such as ‘stand’ and ‘jump’. Thus, we looked into the model and found that the model from NN considered the axes for ‘shoulder-center’ and ‘head’ in the case more than other models. This made the difference in their classification results especially for ‘bow’. For ‘eating’ and ‘drinking’, the samples providing the gesture data were all right-handed; thus, they may easily be similar with right-hand waving gestures from the most alike hand motion around a head. Furthermore, ‘eating’ and ‘drinking’ could be overlapped in motion, but there were different objects in hand in which are not detected by the sensing device. Hence, this pair activity may require another device checking on the held object to clearly classify apart.

V. CONCLUSION

This paper studies and reports on developing activity recognition using Microsoft Kinect as sensing device on four classification models. The method focuses on defining continuous yet independent data from Kinect device into clusters for training an activity classification model. The activities in this works are basic motion gestures such as hand-waving, walking, Thai-style greeting and eating. The experiment was conducted to compare an effectiveness of four different classification algorithms including neural networks, Naive Bayes, Decision tree and support vector machine for precision, recall and f-measure. The results revealed that the best overall classification model was from neural network algorithm while the second best was support vector machine with slightly lower accuracy. From analysis, the most incorrect activities were ‘wai’ (Thai greeting) and ‘walking’.

For improving, we plan to find additional methods to help distinguishing the activities with similar motion gestures such as ‘wai’ and ‘bow’. Moreover, we plan to study on placing a sensing device in different location such as ceiling or using other sensing devices for activity recognition.

REFERENCES

[1] S. Khawandi, B. Daya and P. Chauvet, "Implementation of a monitoring system for fall detection in elderly healthcare", *Procedia Computer Science*, vol. 3, pp. 216-220, 2011.

[2] P. Sajda, "MACHINE LEARNING FOR DETECTION AND DIAGNOSIS OF DISEASE", *Annual Review of Biomedical Engineering*, vol. 8, no. 1, pp. 537-565, 2006.

[3] M. Naeemabadi, B. Dinesen, O. Andersen, S. Najafi and J. Hansen, "Evaluating Accuracy and Usability of Microsoft Kinect Sensors and Wearable Sensor for Tele Knee Rehabilitation after Knee Operation", *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 1: BIODEVICES*, vol. 1, pp. 128-135, 2018.

[4] X. Fan, H. Zhang, C. Leung and C. Miao, "Comparative study of machine learning algorithms for activity recognition with data sequence in home-like environment," *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Baden-Baden, 2016, pp. 168-173. doi: 10.1109

[5] M. Pedraza-Hueso, S. Martín-Calzón, F. Díaz-Pernas and M. Martínez-Zarzuela, "Rehabilitation Using Kinect-based Games and Virtual Reality", *Procedia Computer Science*, vol. 75, pp. 161-168, 2015.

[6] V. C. M. and A. Noel Joseph Raj, "ADVANCES IN THE ANALYSIS OF HUMAN GESTURE RECOGNITION USING KINECT SENSOR: A REVIEW", *ARPN Journal of Engineering and Applied Sciences*, vol. 11, pp. 7147-7154, 2016.

[7] Madelineschiappa, 'Man vs machine: comparing artificial and biological neural networks', 2017. [Online]. Available: https://news.sophos.com/en-us/2017/09/21/man-vs-machine-comparing-artificial-and-biological-neural-networks/. [Accessed: 14-Oct- 2018].

[8] Wikipedia, 'Support vector machine', 2018. [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine. [Accessed: 14- Oct- 2018].

[9] Sunil Ray, '6 Easy Steps to Learn Naive Bayes Algorithm (with codes in Python and R)', 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/. [Accessed: 14- Oct- 2018].

[10] Jinde Shubham, 'Decision Trees in Machine Learning', 2017. [Online]. Available: https://becominghuman.ai/decision-trees-in-machine-learning-f362b296594a. [Accessed: 14- Oct- 2018].

[11] S. Fong, W. Song, K. Cho, R. Wong and K. Wong, "Training Classifiers with Shadow Features for Sensor-Based Human Activity Recognition", *Sensors*, vol. 17, no. 3, p. 476, 2017.

[12] R. Alfuadi and K. Mutijarsa, "Classification method for prediction of human activity using stereo camera," *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, Semarang, 2016, pp. 51-57. doi: 10.1109/ISEMANTIC.2016.7873809

[13] A. Manzi, L. Fiorini, R. Limosani, P. Dario and F. Cavallo, "Two-person activity recognition using skeleton data," in IET Computer Vision, vol. 12, no. 1, pp. 27-35, 2 2018.doi: 10.1049/iet-cvi.2017.0118

[14] E. Cippitelli, S. Gasparrini, E. Gambi and S. Spinsante, "A Human Activity Recognition System Using Skeleton Data from RGBD Sensors", *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1-14, 2016.

[15] Microsoft,Refurbished Kinect Sensor for Xbox One.2018