World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

Orientation-based classification of sit-to-stand activity using Artificial Neural Network in real-time

Romy Budhi Widodo [1,*], Hanfrey Djaya Winataharto [1], Mochamad Subianto [1], Rudy Setiawan [1], Joseph Dedy Irawan [2] and Chikamune Wada [3]

[1] *Human-Machine Interaction Research Center, Informatics Engineering, Faculty of Science and Technology, Ma Chung University, Jl. Villa Puncak Tidar N-01, Malang 65151, Indonesia.*
[2] *Informatics Engineering, Faculty of Industrial Technology, National Institute of Technology Malang, Jl. Raya Karanglo km.2, Malang, Indonesia.*
[3] *Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino- Wakamatsu-ku-Kitakyushu, Fukuoka 808-0196, Japan.*

World Journal of Advanced Research and Reviews, 2023, 17(03), 307–317

Publication history: Received on 31 January 2023; revised on 03 March 2023; accepted on 06 March 2022

Article DOI: https://doi.org/10.30574/wjarr.2023.17.3.0376

Abstract

Human activity recognition is highly required to develop an assistive technology. This study proposes the use of Artificial Neural Network (ANN) to classify Sit to Stand (STS) activity based on the sensor's orientation angle. There are four main phases in this research which are Sit Phase, Flexion Phase, Extension Phase, and Stabilization Phase (Stand). Human activity recognition is highly required to develop an assistive technology. STS activity is an important movement for every human being despite the inability of certain age groups to perform this movement due to weakened muscle function. The limited information from previous on the difficult phases experienced by the subjects to perform STS causes the development process of assistive devices slower. Our solution can classify those phases in real-time using the angles on Korpus Sterni (chest) and Tibia (calf) to gather information on which phase is difficult to be performed. It manages to gather and process the sensor data on application with approximately 3 seconds delay, resulting in the extension phase being a difficult phase to classify. A dataset of 32,000 samples was obtained from 8 subjects consisting of 6 subjects aged 20-30 years and 2 subjects aged 40-50 years. After experimenting and testing the performance of the ANN architecture, the neural network architecture consisted of 4 input nodes, 4 hidden layers (93-69-89-76) with appropriate hyperparameters, and 4 output layers. The training accuracy and testing accuracy reached 86% and 72% respectively.

Keywords: ANN; HAR; IMU Sensor; STS activity

1. Introduction

As growing and aging are among the inevitable natures of human beings, the decline of muscle function as a skeleton mover is absolute in the elderly. A movement called Sit-to-Stand (STS), which requires a balance and muscle movement to raise and move forward the Center of Gravity (COG), has been difficult for the elderly due to a decreased muscle function. In fact, 8% of people aged 65 years and over experienced difficulties in performing STS whereas 3% of them required assistance to do it [1]. Along with the technological development in the 21st century, artificial intelligence has been applied to various fields of life including in the education and health sectors.

The STS activities undergo conscious and unconscious phases between the Sit and Stand position, namely Flexion phase (COG moves forward) and Extension phase (COG moves upward). The research employed inertia sensor and infrared

* Corresponding author: Romy Budhi Widodo

camera to estimate the angles of knee and ankle joints in order to provide the best parameter value during the Extension phase analysis [2]. Researchers in [3] has tried to develop an adjustable standing assistance tool prototype. Other researchers have classified human activity recognition (HAR) into three: dynamic (walking), static (lying), and transitional (bend over-to-stand).

The research applied ANN method with the combination of three activation functions (ReLU, ReLU6, and ELU) [4]. The HAR classification was carried out by using CNN with 1-D and 2-D block system inputs from raw data of accelerometer and gyroscope [5], [6]. The combining of raw data accelerometer from smartphone and IMU sensor on wristband with CNN algorithm was performed in [7]. Monitoring activity in real time to acquire information with great accuracy it takes a very long time, thus by using ANN researchers can process and acquire information much faster [8]. Most of the HAR classification by IMU sensor used the raw data sensor, and the STS phases were dominantly influenced by their angles. It is assumed that the IMU sensor raw data as the classification feature required complicated preprocessing such as eliminating jitter on the accelerometer and drift on gyrometer data. The main problem is the long process of data collection and lack of information on the difficult phases in STS.

Thus, this research aims to identify the difficult phases of STS to gather information and identify the use of ANN in the development of assistive tools application for the elderly. The STS covers four phases, namely Sit Phase, Flexion Phase, Extension Phase, and Stabilization (Stand) Phase.

2. Material and method

2.1. Dataset Retrieval

2.1.1. Apparatus

The preparatory stage of data retrieval involved sensor installment on the subject's body part and equipment synchronization. The data retrieval was carried out by using a sensor inertial measurement unit –IMU type MTw Awinda by Xsens Technologies B.V with a sampling rate of 100 Hz. *MTw Awinda* comes with a station that acts as a receiver and IMU battery charger. The retrieval used two laptops, two smartphones, a tripod with a minimum height of 94 cm, and a chair. The software used was MT Manager, Microsoft Excel, Visual Studio, and Google Colab.

The data recording for the training process and observation was done using a software called *MT Manager*. By using GUI *MT Manager software,* we could see the orientation angles of *roll*, *pitch*, and *yaw*—which were the result of Kalman filtering through the Euler parameter and also available from quaternion calculation. Besides, raw data from the *accelerometer, gyroscope,* and *magnetometer* sensors can be stored [9].

2.1.2. Subjects

Data were collected from eight subjects consisting of six subjects aged 20-30 years and two subjects aged 40-50 years.

2.1.3. Sensors Placement

Two IMUs were attached to the body parts of *Korpus Sterni* and *Tibia,* as illustrated in Figure 1.

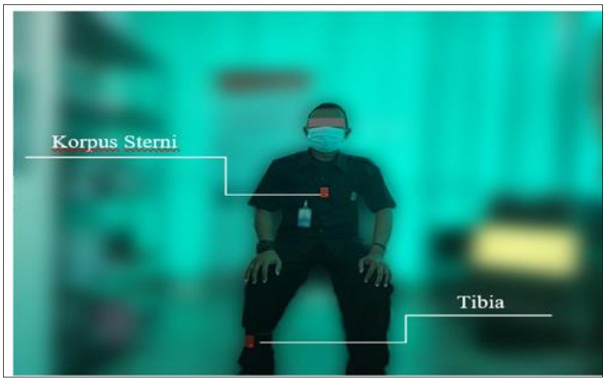
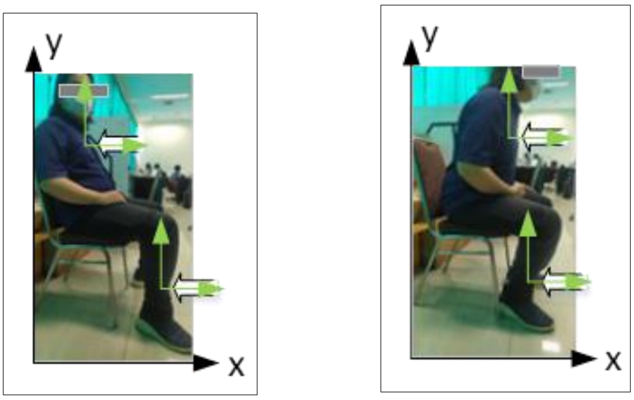


Figure 1 Sensor installment to the Korpus Sterni and Tibia

IMU sensor generally acts as a movement data recorder with *Roll*, *Pitch*, and *Yaw* angles. However, Yaw angles are affected by the magnetic field [10] and the subject heading is not a convenient feature. Therefore, Yaw could not be used in this research. The sensor installment with *Velcro strap* to the *Korpus Sterni* and *Tibia* was done under the consideration that both parts performed minimum muscle contraction, so they would ensure that the MTw was fastened tightly and robustly to the skin. According to [2], sensor installment on the chest would be the best because of the subject's body and tibia slope levels. Besides, the sensor installment to Korpus Sterni and Tibia, as illustrated in Figures 2a and 2b, caused the CoG to move forward during the shift from Sit to Flexion phase, which led to the change of *Korpus Sterni* and *Tibia* angles. At this posture, the *Korpus Sterni* and *Tibia* position leaned forward and formed an angle with the y-axis. The sensor placement in the *Korpus Sterni* was also performed by another researcher in [2].

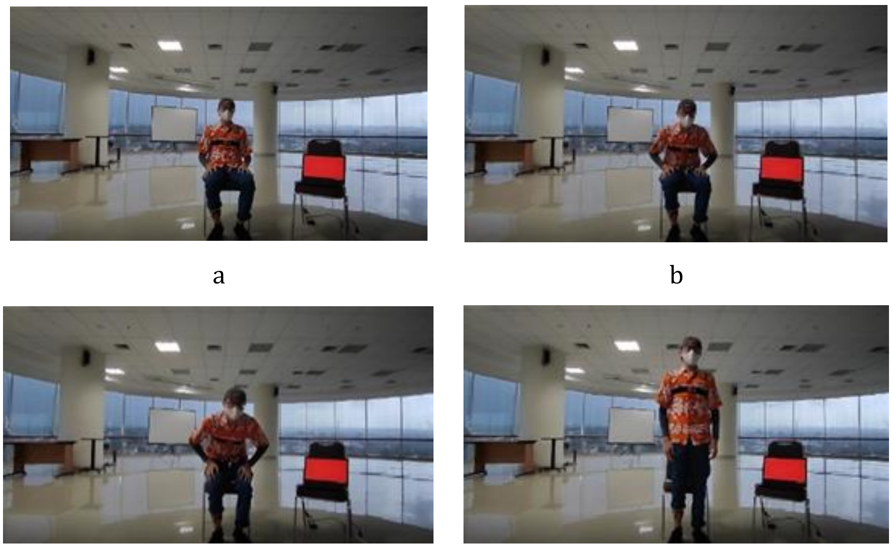


(a) (b)

Figure 2 (a) Sit phase – static CoG; (b) Flexion Phase – moving forward CoG

2.1.4. Experimental procedure

Once the sensor installment completed, subjects were instructed to perform sit-to-stand at a normal speed for 20 seconds. The experiment was repeated twice. During the experiment, not only the angle data from the two sensors were recorded using *MT Manager software*, but the video was also recorded –which later be used for verification in class labeling of the ANN dataset. A monitor screen as a synchronization indicator between data recording and data labeling was placed next to the subject. The green screen indicated that the data had not been recorded by *MT Manager software*, and the red screen indicated the data were being recorded by *MT Manager software.* Figures 3a to 3d were taken from the video recording, illustrating the four STS phases and the screen next to the subject functioned as the indicator of the ongoing recording process. The video used a sampling rate of 30 *fps* and the data recording by *MT* Manager software used a sampling rate of 100 Hz. The calibration of video sampling rate of 100 fps was carried out online via www.video2edit.com/convert-to-video.



a b

c d

Figure 3 (a) Sit phase; (b) Flexion phase; (c) Extension phase; (d) Stand phase

2.2. Inclination Reset Feature on Data Retrieval

The data were retrieved during the forming of ANN dataset with *MT Manager software* and real-time application with C#-based software. To avoid singularities in the Euler parameter, *quaternion* was used during the data retrieval. The use of Euler angles led to singularity problems when the pitch angles reached 90 degrees or -90 degrees. This situation is called Gimbal Lock, which is when the roll and yaw angles are indistinguishable [11], [12]. In this research, the *inclination reset* feature zeroed the roll and pitch angles at the beginning of data retrieval, so the initial sit movement had the pitch and roll angle of 0 degrees on both sensors [10].

2.3. ANN Model Development Process

After completing the dataset retrieval and labeling process, the ANN model was created by using *Python*. The dataset used to perform training is illustrated in Table 1.

Table 1 contains 32,000 rows of data taken from eight subjects.

Table 1 Illustration of training data

roll_korpus	pitch_korpus	roll_tibia	pitch_tibia	phase-class
0.79	-12.27	-0.46	6.31	Sit
0.86	-12.76	-0.48	6.66	Sit
0.95	-13.27	-0.50	6.99	Sit
1.04	-13.78	-0.52	7.29	Sit
1.12	-14.28	-0.53	7.57	Flexion
1.21	-14.76	-0.55	7.84	Flexion
1.29	-15.20	-0.56	8.08	Flexion
:	:	:	:	:
1.38	-15.69	-0.58	8.31	Flexion

Table 1 shows that the dataset had four main features, namely *roll_korpus, pitch_korpus, roll_tibia,* and *pitch_tibia.* The dataset cleaning was done with 10%-15% winsorizing dataset to remove the outliers.

Figure 4 illustrates the data distribution in the box plot after Winsorizing the four classes/phases

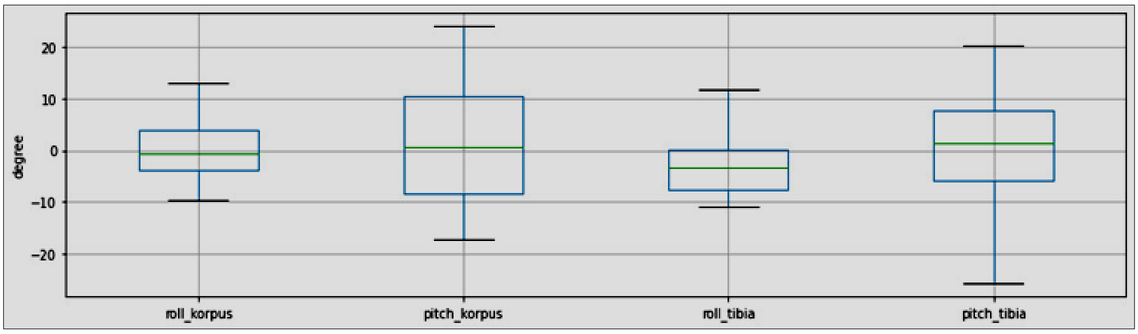


Figure 4 Distribution of training data in the four classes

After reaching the cleaning process, the dataset would be employed as the training data. ANN would be trained for 400 epochs, 95 batch size, with loss function categorical crossentropy. The used ANN architecture was 4 input nodes (roll_korpus, pitch_korpus, roll_tibia, and pitch_tibia), and followed by learning through 4 hidden layers (each layer comprised 93 nodes, 69 nodes 89 nodes, dan 76 nodes) using ReLU activation function. Dropout was also used, as much as 0.15, activated using the softmax function. The output layer comprised 4 nodes that represented 4 classes, namely Sit, Flexion, Extension, and Stand. The model architecture used a multi layer perceptron as illustrated in Figure 5.

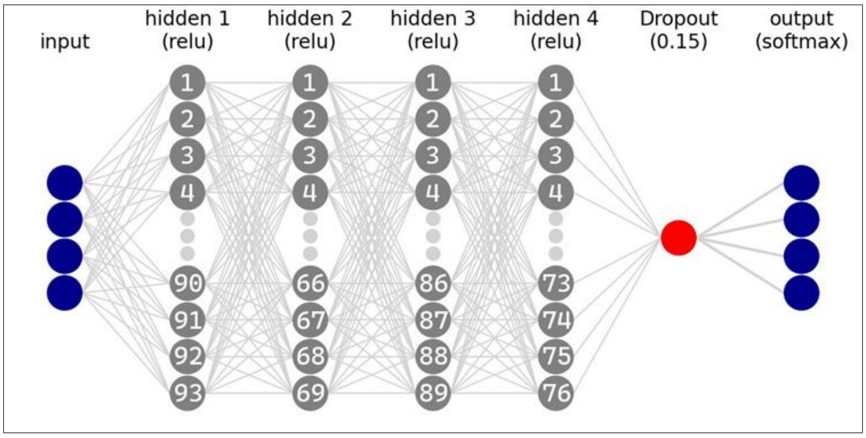


Figure 5 ANN architecture

The categorical output conversion into binary data in the ANN output was done by using a one-hot encoding technique. The subsequent process was the evaluation on the model by using K-Fold Cross Validation and confusion matrix to observe the ANN model performance.

2.4. Real-time Application Design

The classifying real-time application was developed with C# and API IMU of *MTw Awinda*. The designed ANN model was connected to the C#-based application using *Keras.NET*. After being connected to the model, subjects tested the application by performing a sit-stand activity for 6 seconds without repetition. The classification results were immediately displayed in the GUI application.

To evaluate the accuracy, two outputs were provided. The first output was the movement recording file from the two sensors with .mtb extension which could be opened with *MT Manager software*. The second output was a text file comprising the results of phase classification, recorded angle data, and model certainty level in classifying STS phases. The coefficient of model certainty was obtained from the use of the softmax activation function. Likewise, the testing process was recorded on video.

The true-false calculation of classification results was performed manually by comparing videos, sensor movement recordings, and classification text files. The results were recorded in the confusion matrix for further performance analysis. The GUI is shown in Figure 6.

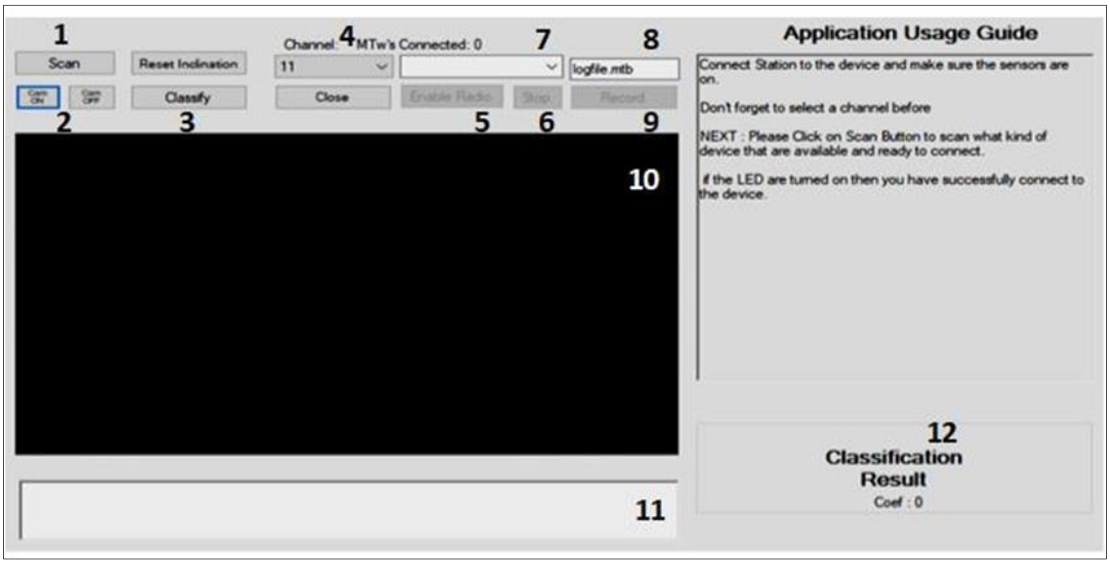


Figure 6 GUI application classifies STS movement in real time

The function of each widget in the GUI screen in Figure 10 based on each number is as follows:

1. "Scan" button to scan the active sensor, which is ready to use.
2. "Cam ON and Cam OFF" buttons to activate and deactivate the webcam.
3. "Classify" button to perform the STS movement classification.
4. "Channel" dropdown button to select channels that have been connected to MT Manager software and API MTw Awinda.
5. "Enable Radio" button to activate Awinda station or USB Dongle, and ready to connect to IMU MTw Awinda
6. "Stop" button to stop all running processes, including webcam and classification processes.
7. Dropdown menu to select the intended sensors (based on the sensor type of MTw / MVN / etc.). This research used MTw.
8. The file name input place for the sensor data recording
9. The "Record" button to activate the data recording
10. Widget to display the webcam capture
11. A place for displaying the roll and pitch angles of each sensor
12. A place for classification result

3. Results

3.1. Result of ANN Model Development

The ANN architecture can be seen in Figure 5. Modifications on the model classifier were created using a fully-connected layer. The dropout value was 0.15 by using *the activation function* of *ReLU* with *softmax activation value* at the final layer and *Adam optimizer*. *The loss function* was *categorical cross-entropy*. The matrix used matrix accuracy. The model training process was performed with a total of 400 epochs and 95 batch size. The model evaluation employed 10-Fold Cross Validation. Figure 7 and 8 show the graph model of accuracy of 86% and graph model of loss of 27%. In other words, as the epochs run, the accuracy got better and the loss would decrease. This means there is a learning process by the model to recognize and learn sit-stand movement from the dataset.

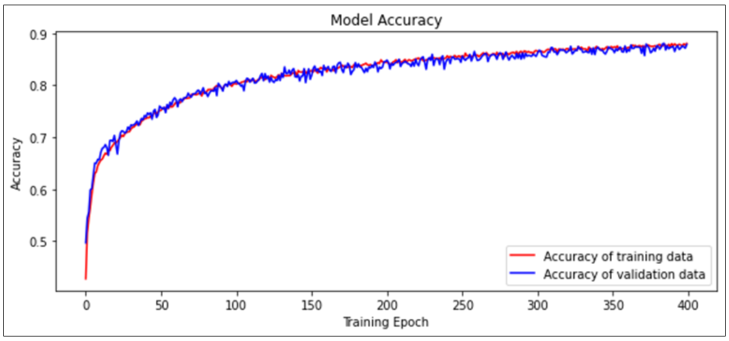


Figure 7 Graph Model of Accuracy

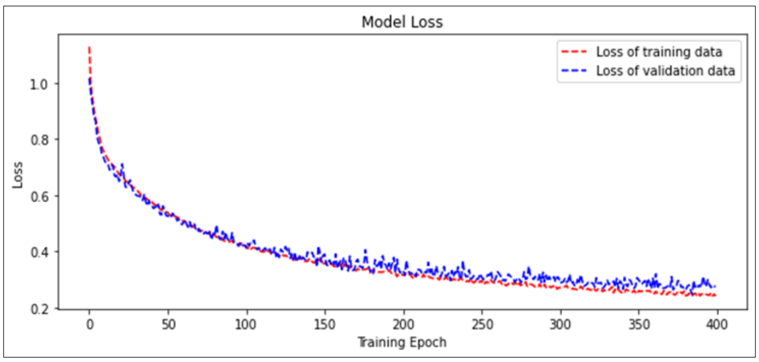


Figure 8 Graph Model of Loss

Figure 9 illustrates the confusion matrix from the visualization of the model's performance.

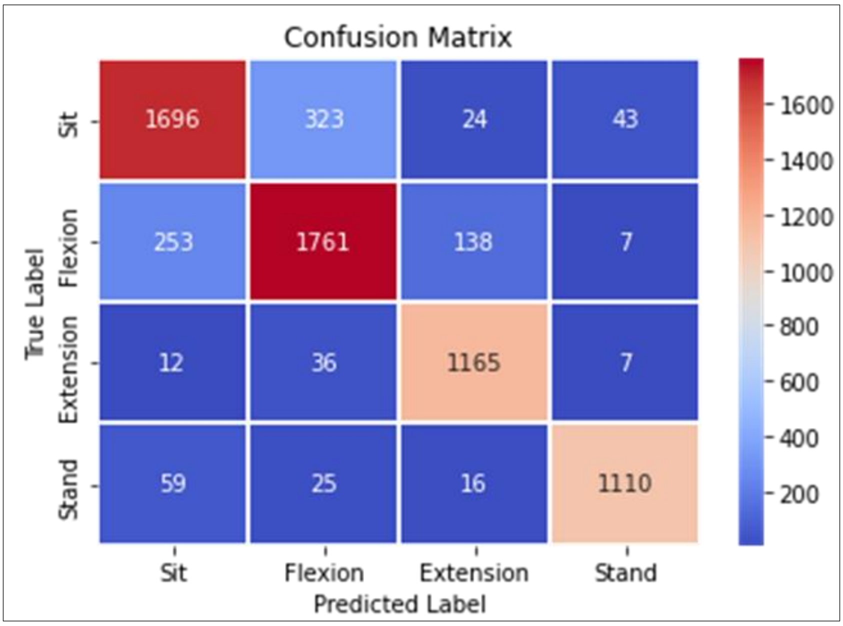


Figure 9 Confusion matrix of model classification results

Table 2 shows the result of confusion matrix from Figure 9. Precisions of each class were 87%, 82%, 84%, and 95%. Recalls of each class were 95%, 82%, 81%, and 92%. F1-Scores of each class were 91%, 82%, 83%, and 93%.

Table 2 Accuracy matrix

	Precision	**Recall**	**F1-Score**
Sit	0.87	0.95	0.91
Flexion	0.82	0.82	0.82
Extension	0.84	0.81	0.83
Stand	0.95	0.92	0.93
Accuracy	0.86		

As soon as the confusion matrix and accuracy matrix in Table 2 was generated, the model's ability and performance to do generalization were tested using 10-Fold Cross Validation. The model testing results using 10-Fold Cross Validation are presented in Table 3.

Table 3 shows that the average accuracy was 86%. Symbol (+/-) indicates that the model accuracy was in the range of 86% ± 5%, which was 81% to 91%. After being tested, the model generated the classification results using argmax and inverse functions on the encoder label. The display example is shown in Figure 10.

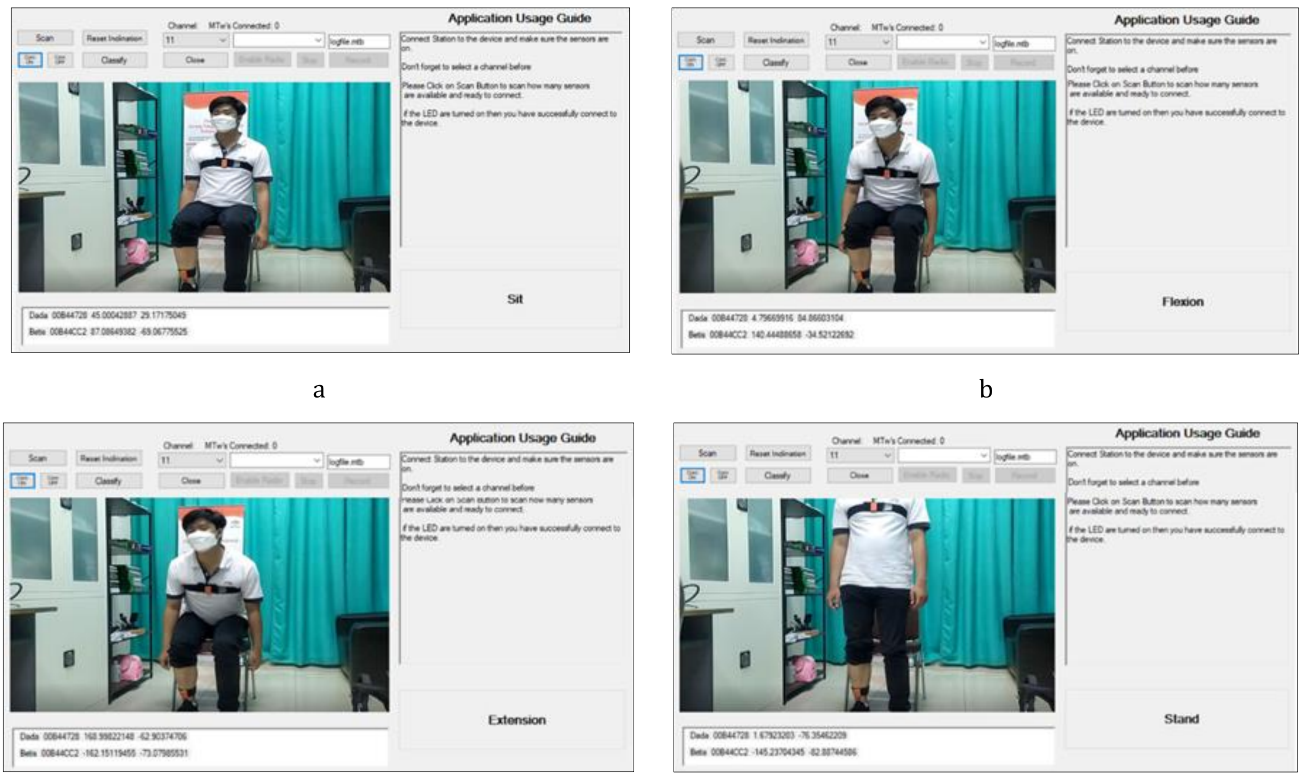
Figure 10 Example of model classification result

Table 3 Model testing results using 10-fold cross validation

Nth fold.	Accuracy	Loss
1	0.83	0.34
2	0.85	0.30
3	0.90	0.24
4	0.88	0.26
5	0.88	0.29
6	0.84	0.36
7	0.91	0.25
8	0.86	0.25
9	0.83	0.31
10	0.86	0.27
Average	0.86± 0.05	0.29± 0.05

After successfully training and providing results, the next stage was connecting the ANN model to the Keras.NET application.

3.2. Application Development Results



a b

c d

Figure 11 (a) Sit Classification Results; (b) Flexion Classification Results; (c) Extension Classification Results; (d) Stand Classification Results

Figures 11a to 11d display the C#-based application that has been connected to the model by using *Keras.NET*. The application could perform movement classification correctly, with a delay time of 1-3 seconds. The next process was the testing of the application.

3.3. Application Testing Results

The results of the classification testing by using ANN in real-time were stated in the confusion matrix which is presented in Table 4 and the accuracy matrix in Table 5.

Table 4 Confusion matrix result of application testing

		True Class			
		Sit	**Flexion**	**Extension**	**Stand**
Predicted Class	Sit	133	3	6	5
	Flexion	3	140	38	5
	Extension	8	45	107	39
	Stand	6	7	44	135

Table 5 Accuracy matrix result of application testing

	Precision	**Recall**	**F1-Score**
Sit	0.90	0.89	0.90
Flexion	0.75	0.72	0.73
Extension	0.54	0.55	0.54
Stand	0.70	0.73	0.72
Accuracy	0.72		

4. Discussion

4.1. During the Training Process

Table 2 shows that the flexion phase had balanced *precision* and *recall* values and *F1-Score*. This model was indeed possible for multiclass classification cases. The two causes were: First, the flexion phase was classified by the model correctly and produced the same False Positive and False Negative values. The *F1-Score* would also be similar to the *precision* and *recall*, as the *F1-Score* was the harmonic average of the two. Second, the data contained an imbalance class [13] because each subject had their rhythm and tempo in performing as many sit-stand movements as they wanted during the time provided. It is presumed that the model can classify the flexion phase well, although the sit or the stand phases are easier to organize.

4.2. During the Real-time Application Testing Process

The Confusion Matrix and the accuracy metrics in Tables 4 and 5 show that the sit phase was the easiest to classify. This was signed by balanced and relatively high *precision* and *recall* values (0.9 and 0.89). Meanwhile, the following observation was the extension phase with relatively low *precision* and *recall* values, 0.54 and 0.55. The low precision value indicates that the model gave a considerable false positive value [14] in classifying extension phase. This means that the extension phase was not correctly classified. In this case, it can also be interpreted that the extension phase was quite difficult to classify with a precision level of 54%. The low *recall* value in the extension phase indicates a lot of missing data [15] or included as classified data that should not have been. For instance, the phase that was initially a flexion phase is considered the extension phase by the model.

Based on the results, if researchers intend to develop assistive tools to be used effectively, they can see the *precision* value resulted by the application. The average *precision* issued by the application was 72%. However, if the researchers want to apply artificial intelligence models to the tools created, they only need to look at the value of *recall*. Moreover, the researchers should pay more attention to using the *recall* because the low *recall* may cause the model to classify the sit phase into the extension or vice versa. This will undoubtedly harm the users if the model is applied to the tool. Such issue can be overcome by hyper tuning the parameters of artificial intelligence. From these results, the researcher can accelerate the process of making seating aids for stands by looking at the model's *precision* or *recall* values of a phase.

5. Conclusion

The developed application can classify the four phases of STS, namely Sit-Flexion-Extension-Stand using ANN. The application delay is approximately 3 seconds with the model’s training accuracy of 86%, and testing accuracy of 72%. The application has been tested using 10-Fold Cross Validation to determine its performance and accuracy. However, the model cannot be used as a center for artificial intelligence applied to assistive devices due to its low accuracy, precision, and recall. However, the application can still be employed to obtain information to be used as a basis for other research. This research provides information on which phase of the model seems difficult to classify—by looking at the value of precision or recall. Based on the testing result, it can be concluded that the extension phase is the most difficult to classify. This is due to the low precision, recall, and F1-score value during this phase (0.54; 0.55; and 0.54). A low recall value will risk the subject in terms of misclassification—the flexion phase is interpreted as the extension phase. Misclassification can cause the subject to be pushed forward. Finally, these findings will provide an important basis for further research in identifying the most difficult phase for the subject in performing STS activities, which shall be useful for the development of assistive technology in the future.

Compliance with ethical standards

Acknowledgments

This work was supported in part by the research funding from the Directorate General of Higher Education, Research, and Technology of the Ministry of Education, Culture, Research, and Technology Indonesia in collaboration with fund management agency LPDP in 2021 Scientific Research Program. This work can be completed due to the collaboration of the Human-Machine Interaction (HMI) Laboratory of Ma Chung University with all parties. Sincere gratitude is extended to the researchers of Human-Activity Recognition in the HMI laboratory.

Disclosure of conflict of interest

We have no conflicts of interest.

Statement of informed consent

Informed consent was obtained from all individual participants included in the study.

References

[1] R. Soangra and T. E. Lockhart, “A comparative study for performance evaluation of sit-to-stand task with body worn sensor and existing laboratory methods,” 49th Annu. Rocky Mt. Bioeng. Symp. 49th Int. ISA Biomed. Sci. Instrum. Symp., pp. 137–144, 2012.

[2] F. Jin, T. Nagasaki, and C. Wada, “An estimation of knee and ankle joint angles during extension phase of standing up motion performed using an inertial sensor,” J. Phys. Ther. Sci., vol. 29, no. 7, pp. 1171–1175, 2017, doi: 10.1589/jpts.29.1171.

[3] C. Wada and S. Ino, “Study of the relationship between sit-to-stand activity and seat orientation,” J. Adv. Comput. Intell. Intell. Informatics, vol. 21, no. 2, pp. 337–341, 2017, doi: 10.20965/jaciii.2017.p0337.

[4] J. Qi, Y. Yang, X. Peng, L. Newcombe, A. Simpson, and P. Yang, “Experimental Analysis of Artificial Neural Networks Performance for Physical Activity Recognition Using Belt and Wristband Devices,” Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS, pp. 2492–2495, 2019, doi: 10.1109/EMBC.2019.8856617.

[5] E. Valarezo, P. Rivera, J. M. Park, G. Gi, T. Y. Kim, and T. Kim, “Human Activity Recognition Using a Single Wrist IMU Sensor via Deep Learning Convolutional and Recurrent Neural Nets,” J. ICT, Des. Eng. Technol. Sci., vol. 1, no. 1, pp. 1–5, 2017, doi: 10.33150/jitdets-1.1.1.

[6] M. Çalişan and M. F. Talu, “Comparison of Methods for Determining Activity from Physical Movements,” J. Polytech., vol. 24, no. 1, pp. 17–23, 2021, doi: 10.2339/politeknik.632070.

[7] V. X. Rahn, L. Zhou, E. Klieme, and B. Arnrich, "Optimal Sensor Placement for Human Activity Recognition with a Minimal Smartphone–IMU Setup," in Proceedings of the 10th International Conference on Sensor Networks (SENSORNETS 2021), 2021, no. Sensornets, pp. 37–48. doi: 10.5220/0010269100370048.

[8] H. Kale, P. Mandke, H. Mahajan, and V. Deshpande, "Human Posture Recognition using Artificial Neural Networks," in IEEE 8th International Advance Computing Conference (IACC), 2018. Accessed: Oct. 19, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/8692143

[9] Xsens, MTw Awinda User Manual, no. May. Xsens Technologies B.V., 2018.

[10] Xsens, MT Manager User Manual, no. February. Xsens Technologies B.V., 2020. [Online]. Available: www.xsens.com

[11] V. Mansur, S. Reddy, and R. Sujatha, "Deploying complementary filter to avert gimbal lock in drones using quaternion angles," in 2020 IEEE International Conference on Computing, Power and Communication Technologies, GUCON 2020, 2020, pp. 751–756. doi: 10.1109/GUCON48875.2020.9231126.

[12] P. D. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Ed. Boston: Artech House, 2013.

[13] H. Ali, M. N. M. Salleh, R. Saedudin, K. Hussain, and M. F. Mushtaq, "Imbalance class problems in data mining: A review," Indones. J. Electr. Eng. Comput. Sci., vol. 14, no. 3, pp. 1552–1563, 2019, doi: 10.11591/ijeecs.v14.i3.pp1552-1563.

[14] O. Theobald, Machine Learning with Python: A Practical Beginner Guide, First. Scatterplot Press, 2019.

[15] A. R. Rae, M. E. Savery, J. G. Mork, and D. Demner-Fushman, "A High Recall Classifier for Selecting Articles for MEDLINE Indexing," AMIA ... Annu. Symp. proceedings. AMIA Symp., vol. 2019, pp. 727–734, 2019.