



Classification of human activity recognition utilizing smartphone data of CNN-LSTM

Widya Rizka Ulul Fadilah¹, Wahyu Andhyka Kusuma², Agus Eko Minarno³, Yuda Munarko^{*4}

Universitas Muhammadiyah Malang, Indonesia^{1,2,3}

Article Info

Keywords:

Classification, Human Activity Recognition, CNN-LSTM, Hypermeter, Divide, Conquer

Article history:

Received: April 06, 2021

Accepted: May 12, 2021

Published: May 31, 2021

Cite:

Fadilah, W. R. U., Kusuma, W. A., & Minarno, A. E., Munarko, Y. (2021). Classification of Human Activity Recognition Utilizing Smartphone Data of CNN-LSTM. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 6(2).
<https://doi.org/10.22219/kinetik.v6i2.1319>

*Corresponding author.

Yuda Munarko

E-mail address:

yuda@umm.ac.id

Abstract

Human activity recognition has been applied in various areas of life by utilizing the gyroscope and accelerometer sensors embedded in smartphones. One of the functions of recognizing human activities is by understanding the pattern of human activity, thereby minimizing the possibility of unexpected incidents. This study classified of human activity recognition through CNN-LSTM on the UCI HAR dataset by applying the divide and conquer algorithm. This study additionally employs tuning hyperparameter to obtain the best accuracy value from the parameters and the proposed architecture. From the test results with the CNN-LSTM method, the accuracy rate for dynamic activity is 99.35%, for static activity is 96.08%, and the combination of the two models is 97.62%.

1. Introduction

The current development of technology has been rapidly elevating, requiring the application of technology in a day-to-day life basis. Technology has been evidently capable of assisting humans in completing most of work, preventing the excessive consumption of energy [1]. The development of smartphones as a smart device with high-level of capabilities, is in accordance with the attached sophisticated components, one of which is the sensor [2] detecting the surrounding conditions, measuring and producing an output from the detected results [3]. Sensors that are generally embedded in smartphones include accelerometer and gyroscope sensors. The accelerometer sensor refers to a responsive sensor which detects the orientation of the device to measure the speed value of the device movement in three directions of the x, y, and z axes [4][5]. Meanwhile, the gyroscope sensor refers to a responsive sensor which measures the rotation of a device based on motion [4][5]. The different position was tasted such as arm, waist, head, shoulder, pocket [6][7].

Utilization of smartphones having accelerometer and gyroscope sensors has been frequently used in Human Activity Recognition, detecting specific recognition of human activity patterns such as standing, walking, jumping, and others [8]. One of the implementations is in monitoring patients suffering from mental illness such as bipolar disorder [9]. Another implementation includes in monitoring a person with disabilities who require 24-hour surveillance [10]. Hence, this study aims at navigating the pattern of a person's activities, as an effort to minimize the possibility of unexpected incidents such as injury, danger, or accident.

Several studies have been devoted on the classification of Human Activity Recognition utilizing several algorithm models that support this research. A study conducted by entitled "Classification of Static and Dynamic Activity in Human Activity Recognition Datasets Using Convolutional Neural Networks", employing the divide and conquer algorithm, further dividing the dataset into two subs of dynamic and static, modeled by using CNN presenting an accuracy of 97% [11].

Another relevant study conducted by Mohib Ullah, Habib Ullah, Sultan Daud Khan, and Faouzi Alaya Cheikh (2019) entitled "Stacked LSTM Network for Human Activity Recognition Using Smartphone Data", applying a single neural network to process data that has been preprocessed, progressed through a stacked LSTM consisting of five LSTM cells and the output layer with softmax activation where the dataset includes the HAR dataset from UCI and the results of this study indicate an accuracy of 93% [12].

Another study conducted by Ronal Mutegeki and Dong Seog Han (2020) entitled "A CNN-LSTM Approach to Human Activity Recognition" employs the CNN layer followed by the LSTM layers, applying the two different HAR datasets sourced from UCI with six classes and iSPL with three classes, involving the three architectural scenarios. The

first architectural scenario is CNN-LSTM with a convolutional layer of 64 filters and a kernel size of three, followed by a max pooling layer and a flatten layer, then an LSTM layer with a ReLU activation function. The second architectural scenario removes the convolutional layer and adds the LSTM layer. The third architectural scenario substitutes the first scenario, which adds a fully connected layer with 50 hidden neurons and ReLU activation [13].

Based on the narrated background, the authors propose the CNN-LSTM method to perform HAR classification on the UCI HAR dataset due to the ability of to study complex activities. In addition, LSTM is deemed effective in capturing temporal information from time series data as it can learn and remember long sequential input data to support multiple parallel sequences of data input.

2. Research Method

2.1 Dataset

Particular studies on this topic, the data are homogeneous such as age similarity in a certain range, the same range of education [14]. The utilized dataset was accessed publicly from Human Activity Recognition applying Smartphones of the Machine Learning UCI Repository [15] which is a renewal of the previous dataset [16], generated from an experiment with 30 volunteers ranging from 19 to 48 years, in which each volunteer will perform six different activities such as walking, ascending upstairs, descending downstairs, sitting, standing, laying through a Samsung Galaxy SII, attached at the waist with an accelerometer sensor and an embedded gyroscope.

2.2 Preprocessing

The applied dataset is inserted into an array, then transposed to change the output dimensions, combining the signals with timesteps to obtain sample, timesteps, feature. The next process is conducted by label encoding and splitting the dataset into 30% testing and 70% training. Hence, the total number of training data classes is 7352 and testing data is 2947. The division for dynamic activities involves 3285 training data and 1387 testing data, while for static activities involves 4067 training data and 1560 testing data as depicted in Table 1.

Table 1. Dataset Division

Dataset	Training	Testing	Total
Dynamic	3.285	1.387	4.672
Static	4.067	1,560	5.627
Combined	7.352	2,947	10,299

2.3 CNN-LSTM

CNN consists of neurons, in which each neuron has a weight and bias, along with several layers including input layer, output layer and several hidden layers; the hidden layer consists of convolutional layer, pooling layer, fully connected layer and various normalization layers [17].

1) Convolutional layers

Convolutional layers becomes the first layer consisting of several filters for feature extraction from input data by applying convolution operations to combine information sets [17][18].

2) Pool layer

Pool layer serves to reduce the input spatially (reducing the number of parameters) from the convolution feature so as to reduce the required computational resources to process the data and to accelerate the computing process [17][18].

3) Dropout

Dropout refers to a process of preventing overfitting and accelerating the learning process.

4) Batch Normalization

Batch Normalization normalizes and reconstructs the input data for each training sample set to ensure the stability of the output from the previous layer, thereby accelerating the training speed and accuracy.

5) Fully-connected layer

Fully connected layer contains a layer to perform the transformations on the data dimensions for linear classification [18], connecting each neuron in one layer to other neurons in another layer in order to classify inputs into classes, based on the trained dataset [17].

Upon the completion process through the CNN layer, further step is continued through the LSTM layer, capable of handling complex serial information with long dependencies, because it uses a gate Scheme for data representation [19]. At last, upon the completion process in both the CNN and LSTM layers, dataset is classified into several classes. This study applies CNN-LSTM for the three dataset testing schemes, comprising dynamic and static datasets.

Table 2. Dynamic Scheme of Best Parameters

Parameter	Mark
Conv1D filters	64
kernel size	5
activation	relu
kernel_initializer	he_uniform
input_shape	128.9
MaxPooling1D	3
Dropout	0.27340242009298493
BatchNormalization()	
Conv1D filters	32
kernel size	5
activation	relu
kernel_initializer	he_uniform
MaxPooling1D	3
Dropout	0.3237482480238347
BatchNormalization()	
flatten	
LSTM	64, return_sequence=True
Dense	3
Optimizer	activation=softmax RMSprop(lr= 0.0001)
Batch_size	10
Epochs	50

Table 3. Static Scheme of Best Parameters

Parameter	Mark
Conv1D filters	64
kernel size	5
activation	relu
kernel_initializer	he_uniform
input_shape	128.9
MaxPooling1D	3
Dropout	0.27340242009298493
BatchNormalization()	
Conv1D filters	32
kernel size	5
activation	relu
kernel_initializer	he_uniform
MaxPooling1D	3
Dropout	0.3237482480238347
BatchNormalization()	
flatten	
LSTM	64, return_sequence=True
Dense	3
Optimizer	activation=softmax RMSprop(lr= 0.0001)
Batch_size	10
Epochs	50

Table 4. Dataset Scheme of Best Parameters in Six Classes

Parameter	Mark
Conv1D filters	64
kernel size	5
activation	relu
kernel_initializer	he_uniform

input_shape	128.9
MaxPooling1D	3
Dropout	0.27340242009298493
BatchNormalization()	
Conv1D filters	32
kernel size	5
activation	relu
kernel_initializer	he_uniform
MaxPooling1D	3
Dropout	0.3237482480238347
BatchNormalization()	
flatten	
LSTM	64, return_sequence=True
Dense	3 activation=softmax
Optimizer	RMSprop(lr= 0.0001)
Batch_size	10
Epochs	50

2.4 Evaluation

Evaluation refers to the stage where the results of model testing will be measured and assessed. In this study, a confusion matrix is utilized by a classification technique to measure the performance. Meanwhile, the results of the confusion matrix are to determine accuracy, precision, recall, and f1-score. Confusion matrix is illustrated in Table 5.

Table 5. Confusion Matrix

		actual	
		Positive	negative
Predicted	Positive	True Negative (TN)	False Positive (FP)
	negative	False Negative (FN)	True Positive (TP)

1) Accuracy

Accuracy represents the number indicating whether the model presents the expected accuracy between the proximity of the actual data and the predicted data [20], defined as Equation 1 [12].

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

2) Precision

Precision denotes a value indicating whether the model presents the level of accuracy between the requested and answered information [21], defined as Equation 2 [12].

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3) Recall

Recall indicates a value indicating whether the model is capable of redefining information [21], defined as Equation 3 [12].

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4) F1-Score

F1-Score is regarded as one of the evaluation calculations that combines the recall value and the precision value [21], defined as Equation 4 [21].

$$F1 - Score = 2 \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{4}$$

3. Results and Discussion

Table 2 presents a dynamic dataset Scheme parameter indicating the best accuracy results from the tuning hyperparameter of 99.35% for testing, and 100% for training with an epochs value of 50, with a training time of 239.13 seconds. As visualized in the following Figure 1 and Figure 2, the plotting loss and accuracy is presented in Table 2.

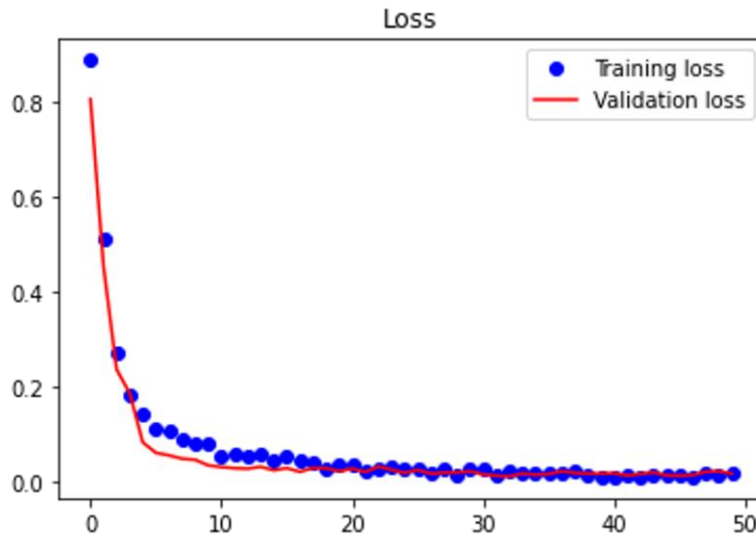


Figure 1. Plotting Loss of Dynamic Scheme

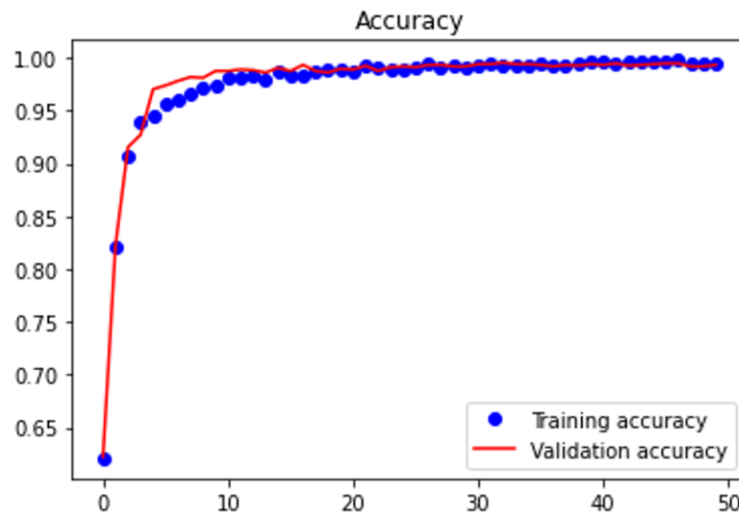


Figure 2. Plotting Accuracy of Dynamic Scheme

The evaluation results of CNN modeling on a dynamic dataset Scheme employing a confusion matrix are illustrated in Table 6.

Table 6. Confusion Matrix on Dynamic Dataset Scheme

<i>Walking</i>	493	1	2
<i>Ascending Upstairs</i>	0	465	6
<i>Descending Downstairs</i>	0	0	420
	<i>Walking</i>	<i>Ascending Upstairs</i>	<i>Descending Downstairs</i>

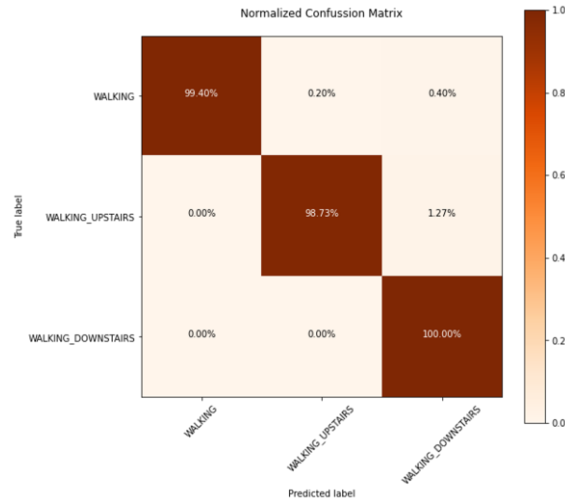


Figure 3. Normalized Confusion Matrix of Dynamic Scheme

Figure 3 indicates a normalized confusion matrix that displays the accuracy of each dynamic class where “Walking” is 99.40%, Ascending Upstairs is 98.73%, and Descending Downstairs is 100%. From the three accuracy values, the highest value is obtained by Descending Downstairs.

The results of implementing CNN-LSTM on a static dataset scheme based on static tables indicate that the best accuracy results from tuning hyperparameter are 96.08% for testing and 98.94% for training with an epochs value of 50, thereby obtaining a training time of 597.55 seconds. Figure 4 and Figure 5 visualize the plotting loss and accuracy of static dataset Scheme modeling.

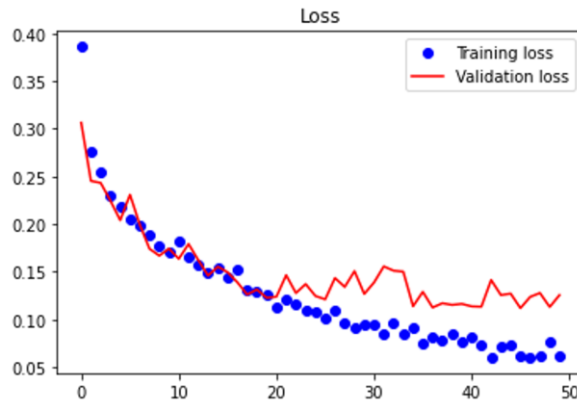


Figure 4. Plotting Loss of Static Scheme

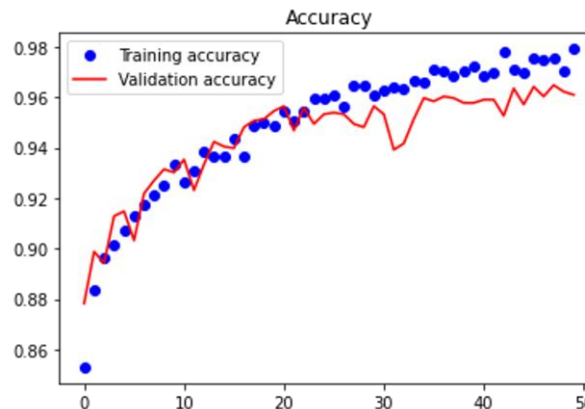


Figure 5. Plotting Accuracy of Static Scheme

The evaluation results of CNN modeling on a static dataset Scheme using a confusion matrix are presented in Table 7.

Table 7. Confusion Matrix on Dynamic Dataset Scheme

Sitting	448	41	2
Standing	17	514	1
Laying	0	0	537
	Sitting	Standing	Laying

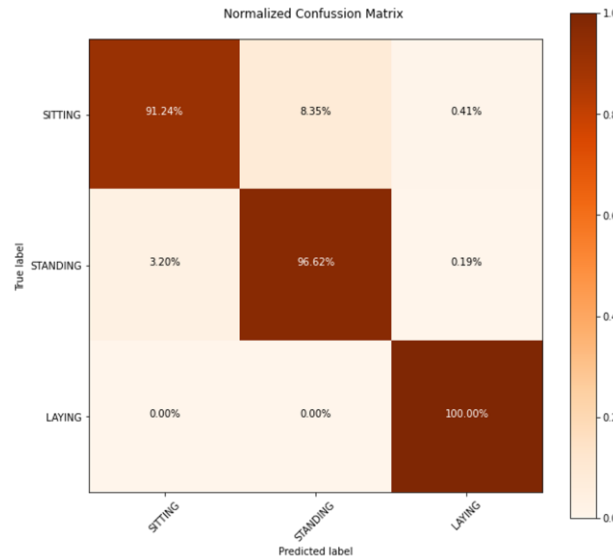


Figure 6. Normalized Confusion Matrix of Static Scheme

Figure 6 indicates a normalized confusion matrix that displays the accuracy of each static class where Sitting has an accuracy of 91.24%, Standing is 96.62%, and Laying is 100%. From the three accuracy values, the highest value is obtained by Laying.

The combined results of the CNN-LSTM model from dynamic and static dataset Scheme modeling obtain an accuracy of 97.64% for testing and 99.41% for training. The Normalized Confusion Matrix in Figure 7 below shows the accuracy values for each class.

From the results of combining these models, the highest accuracy value for descending downstairs and laying activities is 100%, while the lowest accuracy value is from sitting of 91.24%.

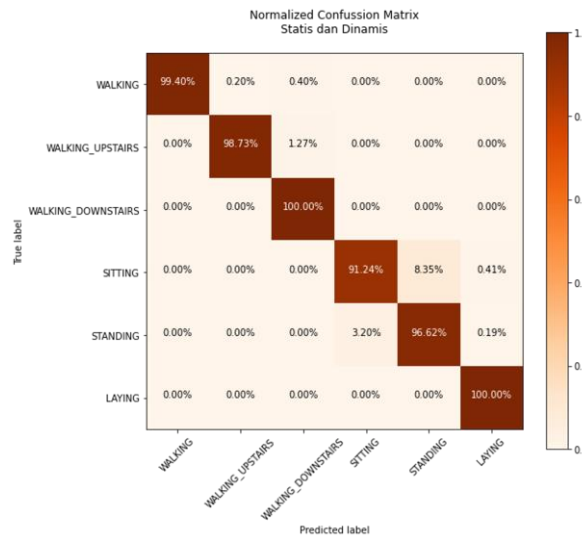


Figure 7. Normalized Confusion Matrix Combination of Dynamic and Static

	precision	recall	f1-score	support
WALKING	0.99	1.00	1.00	493
WALKING_UPSTAIRS	0.99	1.00	0.99	466
WALKING_DOWNSTAIRS	1.00	0.98	0.99	428
SITTING	0.91	0.96	0.94	465
STANDING	0.97	0.93	0.95	555
LAYING	1.00	0.99	1.00	540
accuracy			0.98	2947
macro avg	0.98	0.98	0.98	2947
weighted avg	0.98	0.98	0.98	2947

Figure 8. Precision, Recall, F1-score of Model Mix

Based on the results of the normalized of confusion matrix to the value of precision, recall, and f1-score in Sitting and Standing activities obtain an accuracy value that reaches below 95% of the recall value, for Sitting obtaining 91% and the recall value in Standing obtaining 93%. The total accuracy value for the combined model is 97.62% rounded up to 98%, as illustrated in Figure 8.

The next model applies the CNN-LSTM method which does not apply the divide and conquer algorithm, utilizing tuning hyperparameter to obtain the best parameters and architecture. Table 4 indicates the architecture and parameters with the best accuracy.

Based on the parameter table, the accuracy obtained is 93.17% for testing and 96.74% for training with an epochs value of 30 obtaining a training time of 87.18 seconds. As illustrated in Figure 9 and Figure 10 below, the plotting loss and accuracy in CNN-LSTM modeling is presented without the divide and conquer algorithm.

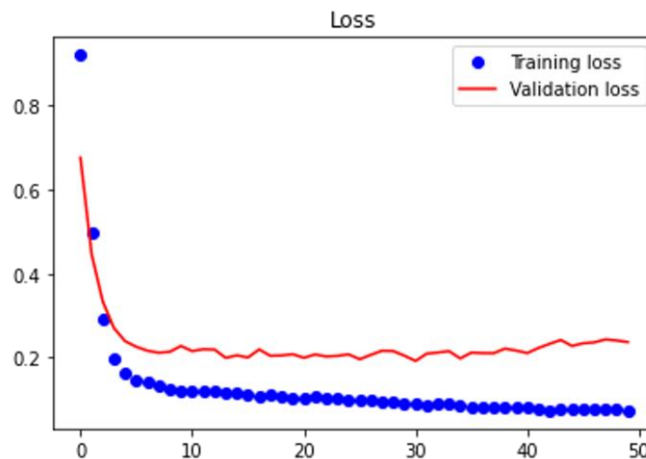


Figure 9. Plotting Loss of Six Class Dataset Scheme

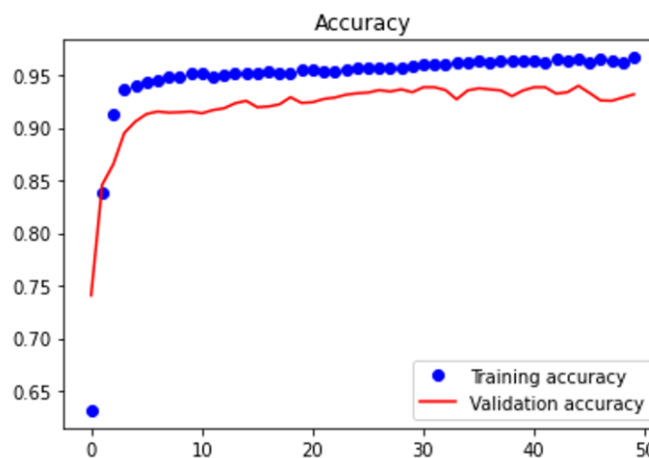


Figure 10. Plotting Accuracy of Six Class Dataset Scheme

The evaluation results of CNN modeling on a six-class dataset Scheme with a confusion matrix, illustrated Table 8.

Table 8. Confusion Matrix of Six-class Dataset Scheme

W	496	0	0	0	0	0
WU	4	455	12	0	0	0
WD	10	4	405	0	1	0
ST	0	1	1	437	47	5
SD	0	0	0	116	416	0
LY	0	0	0	0	0	537
	W	WU	WD	ST	SD	LY

From the results of the confusion matrix, the obtained accuracy value for each class is illustrated in Figure 11. From the results of the confusion matrix, the highest accuracy value is obtained for walking and laying activities, which is 100%, while the lowest accuracy value is obtained for standing activities, which is 78.20%.

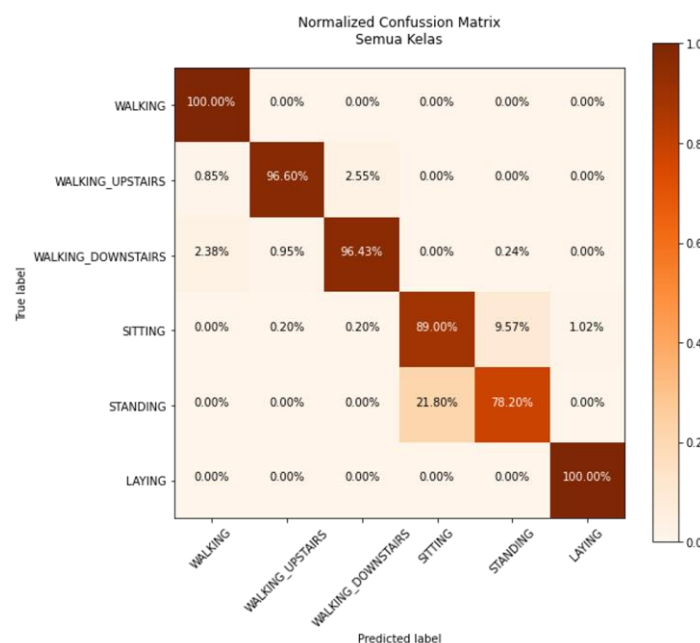


Figure 11. Normalized Confusion Matrix of Six Class Dataset Schemes

	precision	recall	f1-score	support
WALKING	1.00	0.97	0.99	510
WALKING_UPSTAIRS	0.97	0.99	0.98	460
WALKING_DOWNSTAIRS	0.96	0.97	0.97	418
SITTING	0.89	0.79	0.84	553
STANDING	0.78	0.90	0.84	464
LAYING	1.00	0.99	1.00	542
accuracy			0.93	2947
macro avg	0.93	0.93	0.93	2947
weighted avg	0.93	0.93	0.93	2947

Figure 12. Precision, recall, f1-score of Six Class Dataset Scheme

Based on the results of the normalized confusion matrix to the values of precision, recall, and f1-score in the scheme for six classes without dividing the dataset, as depicted in Figure 12. Sitting and Standing activities obtain a less good accuracy value than other activities affecting the total accuracy.

Prior research scenario from Yoga Anggi Kurniawan [11] applied the divide and conquer algorithm utilizing the CNN method which applies the two convolutional layers followed by dropout and max pooling, further flatten and connected by layer, to obtain a combined model accuracy of 97%. Previous research scenario was presented by Mohib Ullah, Habib Ullah, Sultan Daud Khan, and Faouzi Alaya Cheikh [12] applying the LSTM method with five LSTM cells and the output layer contains softmax activation which obtains an accuracy of 93%. Furthermore, the third previous research referred by this study was Ronal Mutegeki and Dong Seog Han [13] pointing out that the CNN-LSTM method was conducted by applying TimeDistributed1D to wrap the convolutional layer and three kernel sizes, followed by the max pooling layer and flatten layer along with the LSTM layer with the ReLU activation function, obtaining an accuracy of 92%.

Based on the previous research, this research applies the CNN-LSTM method using the divide and conquer algorithm by separating the dataset into two sub-sections of dynamic and static; thus, the results of the solutions from the two datasets are combined to form a single entity with six activities. The test scenario in this research is conducted by applying the convolutional layer followed by max pooling, dropout and batch normalization, and progressed with flattening layer and LSTM layer. From the results of these scenarios, the combined accuracy of the models is 97.62%.

The results of the implementation, applying the divide and conquer algorithm could increase the total accuracy of all activities compared to the test scheme without dividing dynamic and static datasets. The obtained accuracy results will be compared with previous studies as illustrated in Table 9.

Table 9. Comparison with Previous Research

No	Method	Accuracy
1	CNN[8]	97%
2	LSTM[9]	93%
3	CNN-LSTM[10]	92%
4	CNN-LSTM proposed	97.62%

4. Conclusion

Based on the performed tests, it is concluded that the highest accuracy is obtained on CNN-LSTM, applying the divide and conquer algorithm with an accuracy value of 99.35% and static schemes of 96.08% and the combination of the two models is 97.62%. Accuracy results on CNN-LSTM which does not apply the divide and conquer algorithm is 93.17%, where the difference is relatively large, reaching 4.45%. In short, the application of the divide and conquer algorithm increases the total accuracy.

Acknowledgement

The authors of this study express their gratitude to the Department of Informatics, University of Muhammadiyah Malang for the completion of this research article. The authors further expect that this research could contribute to the Human Computer Interaction topics.

References

- [1] M. Danuri, "Perkembangan dan Transformasi Teknologi Digital," *infocam*, vol. XV, no. II, pp. 116–123, 2019.
- [2] GE Purna Sastriya, DC Khrisne, and Made Surdarma, "Aplikasi Asisten Untuk Lansia Dengan Memanfaatkan Smartphone Berbasis Android," *SINTECH (Science Inf. Technol. J.)*, vol. 2, no. 2, pp. 63–70, 2019. <https://doi.org/10.31598/sintechjournal.v2i2.315>
- [3] O. Ockikiryanto, "Rancang Bangun Tempat Tidur Pasien Otomatis Dengan Sensor Accelerometer Gyroscope Untuk Mengatur Keseimbangan Berbasis Mikrokontroler Arduino," *Cyclotron*, vol. 2, no. 2, 2019. <https://doi.org/10.30651/cl.v2i2.3256>
- [4] Haniah Mahmudah, Okkie Puspitorini, Nur Adi Siswandari, Ari Wijayanti, and Eliya Alfatekha, "Metode Naive Bayes Classifier – Smoothing pada Sensor Smartphone untuk Klasifikasi Aktivitas Pengendara," *J. Nas. Tech. Electrical and Technol. inf.*, vol. 9, no. 3, pp. 268–277, 2020. <https://doi.org/10.22146/v9i3.382>
- [5] O. C. Kurban and T. Yildirim, "Daily motion recognition system by a triaxial accelerometer usable in different positions," *IEEE Sens. J.*, vol. 19, no. 17, pp. 7543–7552, 2019.
- [6] M. Yang, H. Zheng, H. Wang, S. McClean, and D. Newell, "IGAIt: An interactive accelerometer based gait analysis system," *Comput. Methods Programs Biomed.*, vol. 108, no. 2, pp. 715–723, 2012.
- [7] W. Qi, H. Su, C. Yang, G. Ferrigno, E. De Momi, and A. Aliverti, "A Fast and Robust Deep Convolutional Neural Networks for Complex Human Activity Recognition Using Smartphone," *Sensors*, vol. 19, no. 17, p. 3731, Aug. 2019.
- [8] WS Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, "Human activity recognition using inertial sensors in a smartphone: An overview," *Sensors (Switzerland)*, vol. 19, no. 14, pp. 14–16, 2019. <https://doi.org/10.3390/s19143213>
- [9] N. Cruz Silva, J. Mendes-Moreira, and P. Menezes, "Features Selection for Human Activity Recognition with iPhone Inertial Sensors," *Adv. Arti. Intell. 16th Port. conf. Arti. Intell.*, no. September, pp. 560–570, 2013.
- [10] IA Bustoni, I. Hidayatulloh, AM Ningtyas, A. Purwaningsih, and SN Azhari, "Classification methods performance on human activity recognition," *J. Phys. conf. Ser.*, vol. 1456, no. 1, 2020. <https://doi.org/10.1088/1742-6596/1456/1/012027>
- [11] YA Kurniawan, "Klasifikasi Static Dan Dynamic Activity Pada Human Activity Recognition Dataset Menggunakan Convolutional Neural Network," Universitas Muhammadiyah Malang, 2020.
- [12] M. Ullah, H. Ullah, SD Khan, and FA Cheikh, "Stacked Lstm Network for Human Activity Recognition Using Smartphone Data," *Proc. - Euros. Work. vis. inf. Process. EUVIP*, vol. 2019–Octob, pp. 175–180, 2019. <https://doi.org/10.1109/EUVIP47703.2019.8946180>
- [13] R. Mutegeki and DS Han, "A CNN-LSTM Approach to Human Activity Recognition," *2020 Int. conf. Arti. Intell. inf. comm. ICAIIC 2020*, pp. 362–

- 366, 2020. <https://doi.org/10.1109/ICAIC48513.2020.9065078>
- [14] D. Yazdarsepas et al., "A Multi-featured Approach for Wearable Sensor-Based Human Activity Recognition," *Proc. - 2016 IEEE Int. Conf. Healthc. Informatics, ICHI 2016*, pp. 423–431, 2016.
- [15] J. L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-Aware Human Activity Recognition Using Smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.
- [16] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," *ESANN 2013 proceedings, 21st Eur. Symp. Artif. Neural Networks, Comput. Intell. Mach. Learn.*, no. April, pp. 437–442, 2013.
- [17] A. Dhillon and GK Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *prog. Arti. Intell.*, vol. 9, no. 2, pp. 85–112, 2020. <https://doi.org/10.1007/s13748-019-00203-0>
- [18] MR Alwanda, PRR Kurniawan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," vol. 1, no. 1, 2020. <https://doi.org/10.35957/algorithm.v1i1.434>
- [19] L. Alawneh, B. Mohsen, M. Al-Zinati, A. Shatnawi, and M. Al-Ayyoub, "A Comparison of Unidirectional and Bidirectional LSTM Networks for Human Activity Recognition," *2020 IEEE Int. conf. Pervasive Computing. comm. Work. PerCom Work. 2020*, 2020. <https://doi.org/10.1109/PerComWorkshops48775.2020.9156264>
- [20] A. Rahmawati, "Klasifikasi Tumor Otak Menggunakan Convolutional Neural Network," 2020.
- [21] A. Bimantara and TA Dina, "Klasifikasi Web Berbahaya Menggunakan Metode Logistic Regression," *Annu. res. Semin.*, vol. 4, no. 1, pp. 173–177, 2019.

