

Algorithm performance comparison for earthquake signal recognition on smartphone's accelerometer

Hapsoro Agung Nugroho¹, Haryas Subyantara Wicaksana², Hariyanto³, Rista H. Virgianto⁴

^{1,2,3}Department of Instrumentation for MKG, School of Meteorology, Climatology, and Geophysics, Indonesia

⁴Department of Climatology, School of Meteorology, Climatology, and Geophysics, Indonesia

Article Info

Article history:

Received Nov 29, 2019

Revised Jan 13, 2020

Accepted Apr 10, 2020

Keywords:

Accelerometer

Earthquakes

Human activity

K-NN algorithm

Signal recognition

ABSTRACT

Micro-electro-mechanical-system accelerometer is able to detect acceleration signal caused by earthquake. Such type of accelerometer is also used by smartphones. There are few algorithms that can be used to recognize the type of acceleration signal from smartphone. This study aims to find signal recognition algorithm in order to consider the most proper algorithm for earthquake signal detection. The initial stage of designing the recognizer is data collection for each type of signal classification. The next step is to apply a highpass filter to separate the signals collected from the gravitational acceleration signal. The signal is divided into several segments. The system will extract features of each signal segment in the time and frequency domain. Each signal segment is then classified according to the type of signal using the classifier through a series of training data processes. The classifier which has the highest accuracy value is exported into the new input signal modeling. As the result, fine K-NN algorithm has the highest level of accuracy in the classification. The fine K-NN algorithm has an accuracy rate of 99.75% in the classification of human activity signals and earthquake signals with a memory capacity of 6,044 kilobytes and processing time of 15.93 seconds. This algorithm has the best classifier criteria compared to decision tree, support vector machine and linear discriminant analysis algorithms.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Hapsoro Agung Nugroho,

Department of Instrumentation for Meteorology Climatology and Geophysics (MKG),

School of Meteorology Climatology and Geophysics,

I/5 Perhubungan St., Pondok Betung, Pondok Aren, 15221, Banten, Indonesia.

Email: hapsoro.agung@bmgk.go.id

1. INTRODUCTION

Indonesia is situated in the cross zone of three tectonic plates, namely Eurasia, Indo-Australia and Pacific. Based on this condition, Indonesia becomes a dangerous area of earthquake zones [1, 2]. Thousands of strong motion earthquakes with more than 6.0 moment magnitude impacted Indonesia in 1976-2006 [3, 4]. Law number 31 Year 2009 of Indonesian Republic states that the Meteorology, Climatology and Geophysics Agency (BMKG) is an agency authorized to publish information on earthquake events, both pre-earthquake and post-earthquake cases [5]. Confirmation of the incident at the earthquake site is needed to support the BMKG performance. This certainly involves the active role of the community.

Crowdsourcing is now a prominent terminology to collect, process, and distribute information to society. It possibly becomes a robust idea to overcome the small number of seismic monitoring networks in Indonesia. One of the media for crowdsourcing is smartphone. Nowadays, smartphones are embedded

by digital accelerometer. This component has potential to detect ground strong motion parameters. The Quake-Catcher Network (QNC) and Community Seismic Network (CSN) utilize Micro electro mechanical system (MEMS) accelerometer to detect building vibration caused by earthquake [6]. Such type of accelerometer is also used by smartphones. However, the noise of smartphone's accelerometer faces big obstacles, mainly due to human activities [7].

Anguita et al. from the University of Genova produced a set of accelerometer data to identify six types of human movement. These movements include standing, walking, lying down, sitting, going up and down the stairs [8]. Bayat, Pomplun and Tran from the University of Massachusetts examined the use of a single triaxial accelerometer sensor on smartphones based multilayer perceptron with an accuracy of 90% [9]. Chen et al. succeeded in designing a HAR based on long-short term memory (LSTM) with an accuracy of 92.1% [10]. However, it has not been improved for seismology. This study aims to compare signal recognition algorithm in order to determine the most proper algorithm for earthquake signal detection.

2. RESEARCH METHOD

Smartphone accelerometer sensor type is LSM6DSL. It is a triaxial accelerometer with 16-bit resolution and 50 Hz sampling frequency [11]. Several types of MEMS-type accelerometer signal inputs include gravity acceleration signals, acceleration due to human body movements, offsets and noise [12, 13]. The initial stage of designing the recognizer is to collect data for each type of signal classification. The next step is to apply a highpass filter to separate the signals collected from the gravitational acceleration signal. The signal is divided into several segments. The system will extract features of each signal segment in the time and frequency domain. Each signal segment is then classified according to the type of signal using the classifier through a series of training data processes. The classifier which has the highest accuracy value is exported into the new input signal modeling. The classification test is done by using Python 3 language program in Linux Ubuntu terminal.

Data are collected by recording smartphone accelerometer signals on activities carried out by 10 subjects. Each subject was instructed to sit, stand, lie down, walk and run. The activities were carried out at varying speed and gestures according to the subject's habits. The activities were carried out when the smartphone is placed in shirt pocket and trouser pocket with a variety of types according to the subject's clothing during the study. Samples of earthquake signals were taken from BMKG accelerometer signals that record earthquake events in Lombok and Palu. The amount of earthquake signal raw data is 214 data, while the subject activity raw data are 2545 data in the shirt pocket and 2430 data in the trouser pocket.

Highpass filters are designed to separate the linear acceleration signal of the subject's movement from the gravitational acceleration signal. This filter is a Butterworth type 3 highpass filter with a cutoff frequency of 0.1 Hz. Order 3 is considered to be quite effective to reduce gravitational acceleration signals with dominant frequency ranging from 0.1-0.5 Hz [8]. Data extraction was undertaken by collecting all sample data into a data set. Data windowing uses non-overlapping techniques with frame duration of one second, or 50 raw data in one signal type. The signal was extracted to obtain the signal features in the time domain and frequency domain [14]. Figure 1 shows a flowchart of the process used in this study.

2.1. Decision tree (DT) algorithm

Decision tree is a classifier that works by arranging decision trees based on predictors or features that exist to determine the class of objectives. The parameters required are entropy and information gain of each signal feature. Information gain is a measure of the effectiveness of a feature in classifying data. The information gain equation is formulated as follows [15, 16]:

$$IG(S, f) = Entropy(S) - \sum_{v \in value(f)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$IG(S, f)$ is the information gain value for a particular feature or predictor. $Entropy(S)$ is the value of overall data entropy. V is a possible value for predictor f , while $value(f)$ is a set of possible values for predictor f . $|S_v|$ is the number of samples for values V and $|S|$ is the sum of all data samples. $Entropy(S_v)$ is the value of entropy for samples that have a value of V . The predictor that has the highest IG will be the root of the other predictors in the decision tree. The remaining IG predictor value is recalculated to become the second root, and so on. Root continues branching down, therefore it will appear a destination class at the top of the decision tree [10].

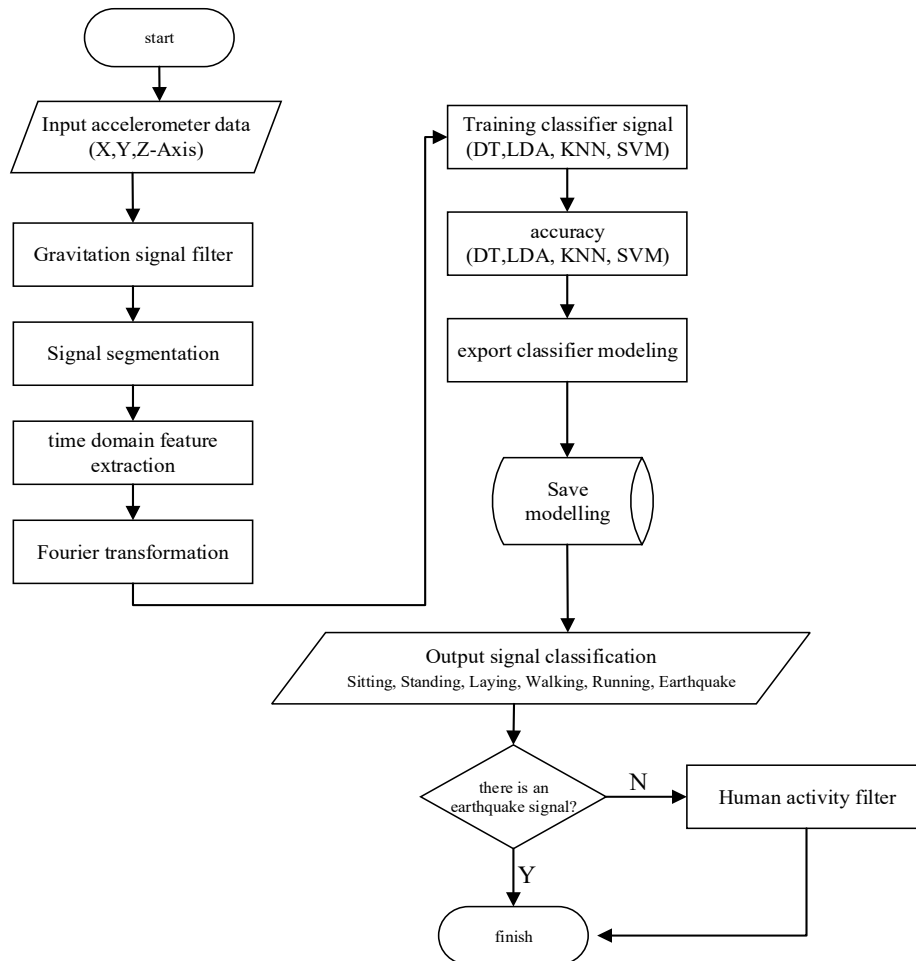


Figure 1. Flowchart activity signal and earthquake signal

2.2. Linear discriminant analysis (LDA) algorithm

Linear discriminant analysis is to reduce the dimensionality of input data predictors. LDA aims to separate the two destination classes by dividing the decision regions [17]. LDA consists of three steps of work [18]. The first step is to calculate separability between classes using the value of the interclass variance. The second step is to calculate the distance between the class average against each sample in the class itself or intraclass variance. The third step is to reduce the dimensional space between classes hence the value of interclass variance is increased while the value of intraclass variance is decreased [12]. The LDA algorithm is coherently described as follows [19]:

- Determine a data matrix with row numbers M and column numbers N
- Calculate the average of each class (1 x M)
- Calculate the average of all classes (1 x M)
- Calculate interclass variance:

$$S_B = \sum_{i=1}^c n_i (\mu_i - \mu) (\mu_i - \mu)^T$$

- Calculate intraclass variance:

$$S_W = \sum_{j=1}^c \sum_{i=1}^{n_j} (x_{ij} - \mu_j) (x_{ij} - \mu_j)^T$$

- Calculate variance matrix:

$$W = S_W^{-1} S_B$$

- Calculate Eigen value and eigen vector from matrix W
- Arrange Eigen vector based on Eigen value. First Eigen vector is used as lower dimensional space (V_k)
- Project all samples to V_k
- Calculate the distance of new sample to data average value which are projected by V_k for each class. The smallest distance value is included into certain class.

2.3. Support vector machine (SVM) algorithm

Support vector machine is to determine the best hyperplane as a separator between destination classes [20]. Hyperplane is a delimiter that divides a vector space into two parts, therefore two different classes can be separated. The best characteristic of hyperplane is the maximum margin value. Margin is the distance between the hyperplane and the nearest predictor vector of the two classes. This predictor vector is called support vector. Hyperplane can be used to separate linear and nonlinear data [21]. The predictor is denoted as x_i and the class is denoted as y_i which contains of two classes that assumed to be -1 and 1, while the w vector is a support vector. The hyperplane equation for linear data is formulated as follows:

$$\vec{y}_i = \vec{w} \cdot \vec{x}_i + b \text{ Hyperplane equation}$$

$$\vec{w} \cdot \vec{x}_i + b \leq -1 \text{ for class -1}$$

$$\vec{w} \cdot \vec{x}_i + b \geq 1 \text{ for class 1}$$

The largest margin can be determined by maximizing the value of the hyperplane's distance to the vector at its closest point. If the margin value has greater value, then it has better classification. Hyperplane for nonlinear data is designed using the Kernel function, which transforms a 2-D vector field into a 3-D vector. Nonlinear predictors are easier to separate in 3-D vector space. The commonly used kernel functions are the polynomial, gaussian and sigmoid kernels [22].

- Polynomial kernel transformation equation:

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^p, p \text{ is the highest number of exponent}$$

- Gaussian kernel transformation equation:

$$K(\vec{x}_i, \vec{x}_j) = \exp(-\sigma \|\vec{x}_i - \vec{x}_j\|^2), \sigma \text{ is variance of vector } x$$

- Sigmoid kernel transformation equation:

$$K(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \vec{x}_i \cdot \vec{x}_j + \beta), \alpha \text{ and } \beta \text{ are sigmoid constant}$$

The hyperplane equation applied to nonlinear data by utilizing the Kernel function is calculated as follows:

$$\vec{y}_i = \vec{w} \cdot K(\vec{x}_i, \vec{x}_j) + b$$

2.4. K-nearest neighbor (K-NN) algorithm

K-Nearest Neighbor algorithm is to calculate the distance of a new data input against the K-data learning model [23]. This algorithm is also useful to search for the nearest neighbor from a new data input. The proximity of new input data to the model data is generally calculated using the Euclidean Distance [24] as follows:

$$E(A, B) = \sum_{i=1}^N \sqrt{(A_i - B_i)^2}$$

A is the new input data, while B is the learning model data. The new input data is tested against each learning data point, then against the neighbor sequence with the smallest distance value according to the K number. K-NN has several types of processing algorithms [25], namely:

- Fine K-NN, using the closest neighbor (K=1).
- Medium K-NN, using ten closest neighbors (K=10).
- Coarse K-NN, using one hundred closest neighbors (K=100).
- Cosine K-NN, using distance calculation of the closest neighbor based on cosine distance matrix:

$$d = 1 - \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|}$$

- Cubic K-NN, using distance calculation of the closest neighbor based on cubic distance matrix:

$$d = \sqrt[3]{\sum_{i=1}^n |u_i - v_i|^3}$$

In general, the K-NN algorithm steps can be described as follows:

- a. Determine K value or number of the closest neighbor
- b. Calculate new input distance to all learning models.
- c. Arrange the new input distance from the closest to far distance
- d. Detemine closest neighbor category based on K value.
- e. Use majority category class of closest neighbor to predict the result of new input data.

3. RESULTS AND ANALYSIS

Table 1 shows the results of the smartphone accelerator signal extraction for various activities in the time domain. Table 1 proves that significant difference occurs in characteristics between human activity signals and earthquake signals in the time domain. Linear acceleration signals due to human activity reported greater value for all features compared to earthquake signals. The signal distribution of acceleration of human activity tends to be leptokurtic, while earthquake signals tend to be platykurtic when viewed from the value of signal kurtosis. The skewness value of each signal type does not show a significant difference.

Table 1. Signal's feature extraction result in time domain

Feature (m/s ²)	Sitting	Standing	Laying	Walking	Running	Earthquake
meanX	1.65	1.41	1.88	0.80	0.97	0.04
meanY	2.77	1.77	2.31	2.98	3.39	0.10
meanZ	1.88	1.97	2.58	0.82	0.98	0.01
median	1.68	1.41	1.87	0.83	1.17	0.04
median	2.83	1.79	2.39	2.97	4.49	0.10
medianZ	1.95	1.95	2.61	0.93	1.25	0.01
madX	0.93	0.78	1.20	1.52	3.74	0.02
madY	1.26	0.90	1.33	2.70	7.61	0.04
madZ	1.16	1.02	1.32	1.67	3.60	0.02
rmsX	2.17	1.84	2.58	2.26	5.13	0.05
rmsY	3.41	2.25	3.03	4.97	10.48	0.12
rmsZ	2.57	2.49	3.28	2.51	5.29	0.02
stdevX	1.15	0.97	1.54	2.03	4.98	0.02
stdevY	1.53	1.11	1.69	3.52	9.43	0.04
stdevZ	1.42	1.26	1.68	2.26	5.11	0.02
maxX	2.77	2.17	3.80	5.19	13.04	0.05
maxY	4.00	2.55	4.00	8.23	19.55	0.13
maxZ	3.34	3.10	4.30	6.22	16.14	0.02
minX	-2.42	-2.19	-3.24	-5.20	-13.07	-0.02
minY	-2.94	-2.38	-3.32	-7.93	-19.03	-0.08
minZ	-2.81	-2.40	-3.56	-6.12	-15.40	-0.02
SMA	728.42	582.28	779.85	776.74	1640.87	18.87
skewnessX	-0.06	0.01	0.06	0.01	-0.01	-0.06
skewnessY	0.11	0.05	0.03	0.02	0.01	-0.11
skewnessZ	0.06	-0.06	0.04	-0.04	0.10	-0.02
kurtosisX	3.37	3.15	3.71	4.19	4.20	1.19
kurtosisY	2.94	3.33	3.46	3.98	3.37	1.27
kurtosisZ	3.41	3.40	3.62	4.88	6.12	1.10

Table 2 shows that the activity signal energy is much greater compared to the earthquake signal energy, while the dominant frequency of earthquake signal tends to be greater than the dominant frequency of non-locomotor activity signal. Running and running activity signals hold more unique information content than other activity signals due to their high entropy values, therefore it is more easily identified. Earthquake signals also demonstrate quite different feature values compared to human activity signals. Prominent features to classify human activity signals and earthquake signals include maximum amplitude, high school values, dominant frequency, spectral centroids and signal energy.

Table 2. Signal's feature extraction result in frequency domain

Frequency Feature	Sitting	Standing	Laying	Walking	Running	Earthquake
fdomX (Hz)	1.11	1.15	1.28	2.97	4.17	2.36
fdomY (Hz)	1.23	1.49	1.39	2.34	3.02	2.30
fdomZ (Hz)	1.00	1.01	1.21	5.33	12.24	2.70
energyX (kJ)	78.98	56.90	93.55	64.57	343.12	0.48
energyY (kJ)	153.25	72.85	120.60	333.08	1240.18	3.86
energyZ (kJ)	85.44	83.50	140.17	79.12	345.94	0.02
centroidX (Hz)	5.63	5.55	6.86	9.36	11.56	41.69
centroidY (Hz)	4.04	4.46	6.01	7.16	8.75	39.98
centroidZ (Hz)	5.07	4.86	6.56	10.18	14.26	44.73
entropyX (bits)	1.67	1.89	1.51	2.21	1.66	1.21
entropyY (bits)	1.04	1.56	1.17	1.37	1.16	1.26
entropyZ (bits)	1.41	1.36	0.98	2.12	1.68	1.17
meanfreqX (Hz)	1.70	1.68	2.19	4.33	5.93	45.12
meanfreqY (Hz)	1.18	1.48	1.89	2.86	3.81	43.75
meanfreqZ (Hz)	1.61	1.42	1.93	5.01	8.94	47.09

3.1. Signal classification using decision tree (DT) algorithm

The decision tree classification is built based on a decision tree scheme. The types of DT algorithms used include simple DT, medium DT and complex DT. The difference in the three types of DT is based on the number of branches/decision tree nodes. The number of DT medium nodes is greater than that of simple DT, while the number of DT complex nodes is greater than the number of DT medium nodes. The Simple DT decision tree has 7 nodes. The signal feature with the highest information gain (IG) is the mean absolute deviation on the z axis. This feature is used by the DT algorithm to classify earthquake signals and human activity signals. Other features used are spectral power and mean absolute deviation on the y axis. The simple DT algorithm is unable to classify signals of sitting, standing and lying activity due to the limited number of nodes.

The Medium DT decision tree has 34 nodes. Features with high IG values used by the medium DT include mean absolute deviation, spectral power, spectral centroids, root mean square, mean frequency, standard deviation and dominant frequency. DT medium algorithm is able to classify almost all HAR data, except signal from sitting activity. Sitting activity signal cannot be classified properly because its features have a small IG value, therefore it is classified as other activity signals. The complex DT decision tree has 193 nodes. All features are used in the complex DT algorithm. These features can classify signals of sitting, standing, lying, walking, running and earthquake signals. All signal features, both time and frequency domain features, are still used in the complex DT, hence this algorithm is more complex compared to other DT algorithms and this algorithm is able to map nodes with a small IG.

Table 3 shows the results of testing the simple DT, medium DT and complex DT. The complex DT algorithm has higher level of classification accuracy compared to the medium DT, both when the smartphone is in shirt pocket and trouser pocket, with accuracy values of 84.36% and 83.49%, respectively. This proves that the number of decision tree nodes is directly proportional to the accuracy of the DT algorithm. An increase in the number of decision tree nodes can increase the classification ability of the DT algorithm.

Table 3. The performance of DT algorithm

Smartphone Position	Type of DT	True (TP+TF)	False (FP+FN)	Accuracy (%)
Trouser Pocket	Simple DT	1753	792	68.88
	Medium DT	1877	668	73.75
	Complex DT	2147	398	84.36
Shirt Pocket	Simple DT	1534	896	63.13
	Medium DT	1696	734	69.79
	Complex DT	2029	401	83.49

3.2. Signal classification using linear discriminant analysis (LDA) algorithm

LDA algorithm classifies signals using vectors in the form of linear lines. Table 4 shows the accuracy of the LDA algorithm. The accuracy of this algorithm is lesser than the accuracy of the decision tree algorithm. Walking, running and earthquake signals have low predictive error rates in the LDA algorithm. Figure 2 shows the cluster of signal data based on features with the highest IG (information gain) value. These features include mean absolute deviation on the z axis and y axis and spectral power on the y axis. Figure 2 demonstrates that the three types of signals can be separated using linear lines vector to be classified properly. Sitting, standing and lying activity signals have overlapping clusters. Therefore, if these activities are plotted based on features with the highest IG, it will be difficult to separate using linear line vectors. This causes the signal of sitting, standing and lying activity cannot be classified properly by the LDA algorithm. Figure 3 shows the proof of such statement.

Table 4. The performance of LDA algorithm

Smartphone Position	True (TP+TF)	False (FP+FN)	Accuracy (%)
Trouser Pocket	1900	645	74.67
Shirt Pocket	1745	685	71.81

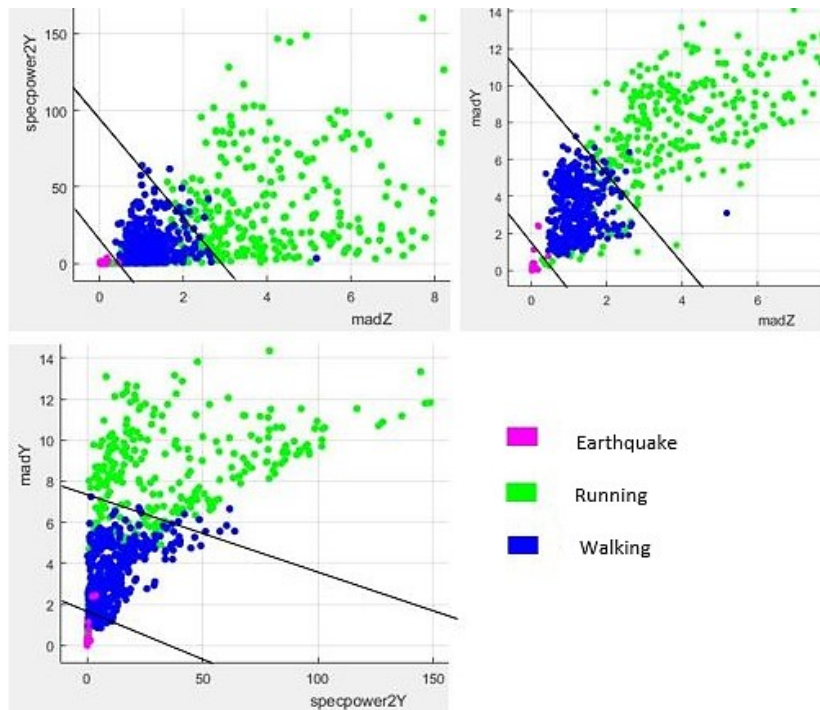


Figure 2. Cluster data of locomotor activity signal and earthquake signal

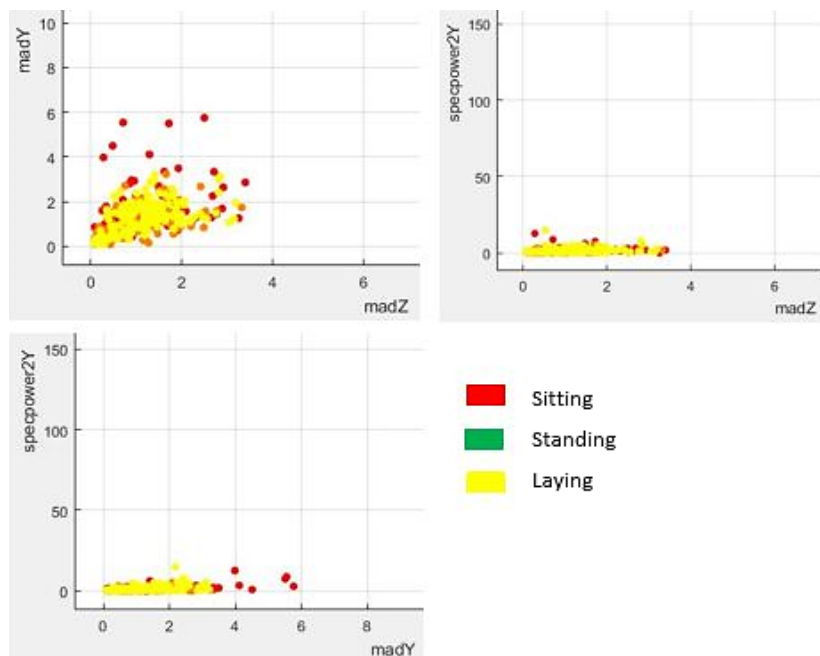


Figure 3. Cluster data of nonlocomotor activity signal

3.3. Signal classification using support vector machine (SVM) algorithm

SVM algorithm uses hyperplane to classify signal classes. Types of SVM algorithms tested include linear SVM, quadratic SVM, cubic SVM, fine SVM, medium SVM and coarse SVM. These types are obtained based on

different types of hyperplane. Quadratic SVM and cubic SVM use polynomial kernels. This kernel projects hyperplane fields with nonlinear functions in three-dimensional space. Quadratic SVM uses polynomial power of 2, while cubic SVM has polynomial power of 3. The classification results of SVM algorithm with polynomial kernel report good accuracy, which is higher than 80%. Polynomial kernels are designed for nonlinear data and made hyperplane to be flexible in separating between classes of activity signal data, therefore the classification accuracy is relatively high.

Fine SVM, medium SVM and coarse SVM use gaussian kernels with tiered kernel scale values. Kernel scale type fine SVM is 2.9; Medium SVM type is 9.1; and Coarse SVM type is 36. Kernel scale is a multiplier factor in the Gaussian kernel. An increase in the value of the kernel scale generally causes a decrease in the accuracy of the Gaussian type SVM algorithm as shown in Table 5, both for the location in the trouser pocket and shirt pocket. The increase in kernel scale decreases the hyperplane margins, therefore the classification between classes of activity data becomes more ambiguous. The confusion in classification results in suboptimal accuracy.

Linear SVM does not utilize the kernel because it considers the activity signal data as linearly correlated data. SVM Linear algorithm is formed in the form of linear vector in two-dimensional space. Activity signal data is nonlinear data, therefore the results of linear SVM classification show lower accuracy compared to other types of SVM algorithm, except Coarse SVM. Figure 4 shows a cluster of activity signal data in the time domain feature. Figure 5 shows a cluster of activity signal data in the frequency domain feature. Both figures prove that activity signal data is difficult to classify using linear functions, because they overlap in the same plot area.

Table 5. The performance of SVM algorithm

Smartphone Position	Type of SVM	True (TP+TF)	False (FP+FN)	Accuracy (%)
Trouser Pocket	Linear	1887	658	74.14
	Quadratic	2435	110	95.67
	Cubic	2233	312	87.74
	Fine	2367	181	92.89
	Medium	2084	496	80.77
Shirt Pocket	Coarse	1785	759	70.16
	Linear	1727	703	71.07
	Quadratic	2168	262	89.22
	Cubic	2380	50	97.94
	Fine	2232	198	91.85
	Medium	1878	552	77.28
	Coarse	1588	842	65.35

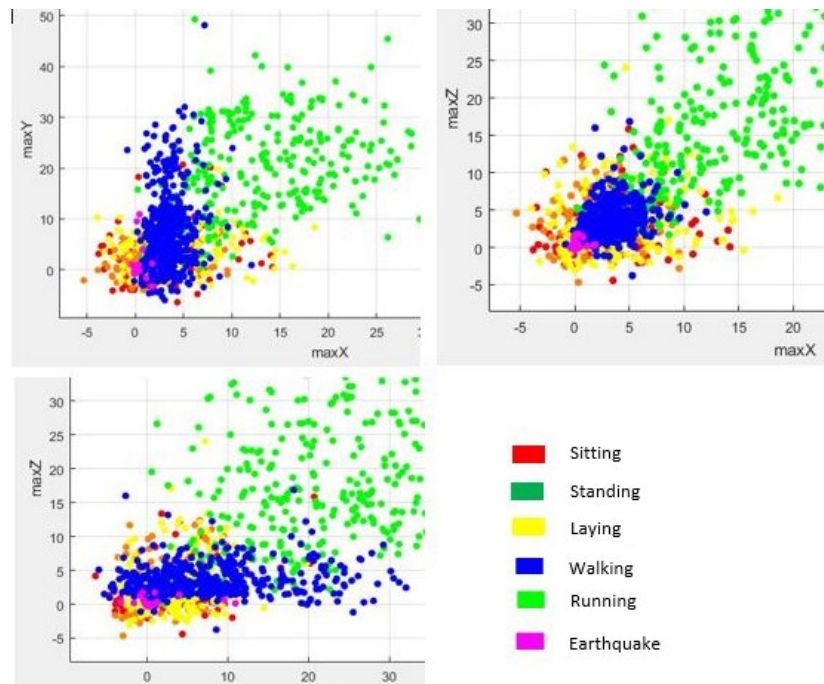


Figure 4. Cluster data of activity acceleration signal in time domain

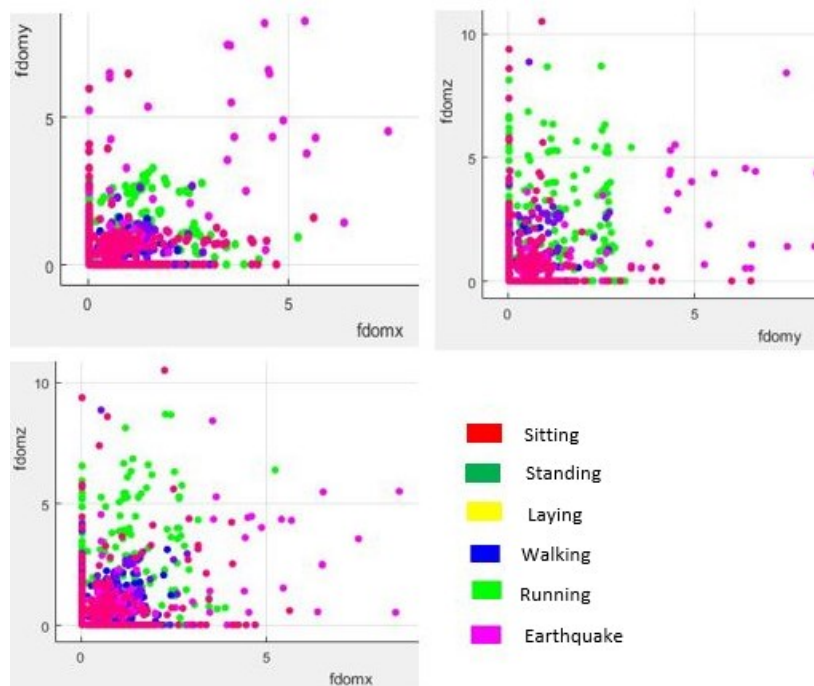


Figure 5. Cluster data of activity acceleration signal in frequency domain

3.4. Signal classification using K-nearest neighbor (K-NN) algorithm

Types of K-NN algorithms tested include fine K-NN, Medium K-NN, Coarse K-NN, Cosine K-NN and Cubic K-NN. The fine K-NN, Medium K-NN and Coarse K-NN algorithms are distinguished based on the number of nearest neighbors, while Cosine K-NN and Cubic K-NN are distinguished according to the provisions of the calculation of the distance to the nearest neighbor. Table 6 shows the results of testing the signal data classification using the K-NN algorithm. Fine K-NN only considers 1 nearest neighbor to classify data. Medium K-NN considers the 10 closest neighbors, while Coarse K-NN considers 100 closest neighbors in a data cluster. Table 6 demonstrates that the increase of the number of nearest neighbors as a consideration of the K-NN algorithm actually causes a decrease in the level of classification accuracy. The number of nearest neighbors as predictors actually adds to the level of algorithm ambiguity in determining the signal input class. Nonlinear HAR data has a cluster area with overlapping data classes, therefore it is heterogeneous. The criteria to increase the number of nearest neighbors expands the coverage of the cluster area, causing data classes to become more heterogeneous, and resulting in confusion over the results of classification predictions. The Cosine K-NN algorithm uses cosine metric distance to determine the distance to the nearest neighbor, while Cubic K-NN employs cubic metric distance. The accuracy of these two distance rules are not lower than the Euclidean distance used by fine K-NN and Medium K-NN to classify signals. Cosine K-NN and Cubic K-NN accuracy levels are lower than fine K-NN and Medium K-NN.

Table 6. The performance of K-NN algorithm

Smartphone Position	Type of K-NN	True (TP+TF)	False (FP+FN)	Accuracy (%)
Trouser Pocket	2543	2	99.92	99,92
	1936	609	76.07	76,07
	1742	803	68.45	68,45
	1911	634	75.09	75,09
	1903	642	74.77	74,77
Shirt Pocket	2420	10	99.59	99,59
	1781	649	73.29	73,29
	1631	799	67.12	67,12
	1771	659	72.88	72,88
	1729	701	71.15	71,15

3.5. Algorithm performance comparison result

Each algorithm has a different level of classification accuracy. The algorithm selection criteria do not only include the level of accuracy, but also the speed of data processing and memory capacity. Table 7 shows

a comparison of all classifier algorithms based on these criteria. The accuracy of the algorithm is obtained from the average accuracy of the smartphone activity signal classification in trouser pocket and shirt pocket.

Table 7. The performance comparison of the algorithms

Algorithm	Accuracy (%)	Processing Time (seconds)	Memory Capacity (kilobytes)
Simple DT	66.01	5.17	12
Medium DT	71.77	5.27	22
Complex DT	83.93	6.91	54
LDA	73.10	4.82	387
Linear SVM	72.61	14.12	6,529
Quadratic SVM	92.44	21.97	21,213
Cubic SVM	92.84	25.57	16,564
Fine SVM	92.37	81.2	65,411
Medium SVM	79.03	32.05	29,087
Coarse SVM	67.76	33.63	29,641
Fine K-NN	99.75	15.93	6,044
Medium K-NN	74.70	13.36	6,044
Coarse K-NN	67.78	15.9	6,044
Cosine K-NN	73.98	13.24	6,044
Cubic K-NN	72.96	428.24	6,045

DT algorithm has fast processing time with the smallest memory capacity, but the level of accuracy is lower compared to the fine K-NN algorithm and some types of SVM algorithms. The processing time and memory capacity of the DT algorithm increases proportionally with the number of nodes and the level of complexity of the decision tree it builds. The LDA algorithm has the fastest processing time with a small memory capacity, but the level of accuracy is lower than the average accuracy of DT, SVM and K-NN algorithms. SVM algorithm has the largest memory capacity with a longer processing time compared to the DT algorithm, LDA and some K-NN. The design of hyperplane in the SVM algorithm produces greater number of variables compared to the design of the decision tree on DT, linear lines on the LDA and calculation of the nearest neighbor distance on K-NN. The number of variables affects the processing time and memory capacity of the algorithm. The K-NN algorithm has a processing time of 13 to 16 seconds, except cubic K-NN. The calculation of cubic matrix distance is more complex than Euclidean distance or cosine matrix distance, thus cubic-KNN requires long processing time. The average memory capacity of the K-NN algorithm is 6,044 kilobytes. This capacity is far greater compared to DT and LDA, but smaller than SVM. The fine-KNN algorithm has the highest accuracy value, medium processing time and medium memory capacity.

4. CONCLUSION

Fine K-NN algorithm has the highest level of accuracy in the classification of human activity signals with lower memory capacity and shorter processing time. The number of fine K-NN's closest neighbors is only one, therefore it does not cause ambiguity in the classification process. The fine K-NN algorithm has an accuracy rate of 99.75% in the classification of human activity signals and earthquake signals with a memory capacity of 6,044 KB and a processing time of 15.93 seconds. This algorithm has the best classifier criteria compared to DT, SVM and LDA algorithms.

REFERENCES

- [1] R. Harris, J. Major, "Waves of destruction in the East Indies: the Wichmann catalogue of earthquakes and tsunami in the Indonesian region from 1538 to 1877," *Geol Soc London*, vol. 441, no. 1, pp. 9-46, 2017
- [2] J. F. Di Leo, J. Wookey, J. O. S. Hammond, et al., "Mantle flow in regions of complex tectonics: Insights from Indonesia," *Geochemistry, Geophys Geosystems*, vol. 13, no. 12, 2012.
- [3] T. Lay, H. Kanamori, C. J. Ammon, et al., "Depth-varying rupture properties of subduction zone megathrust faults," *J Geophys Res Solid Earth*, vol. 117, no. B4, 2012.
- [4] T. M. Rasyif, S. Kato, "Development of accurate tsunami estimated times of arrival for tsunami-prone cities in Aceh, Indonesia," *Int J Disaster Risk Reduct.*, vol. 14, pp. 403-410, 2015.
- [5] BMKG, "Law 31 Year 2009 of Indonesian Republic," Jakarta: Sekretariat Negara, 2009.
- [6] M. D. Kohler, T. H. Heaton, M. H. Cheng, "The Community Seismic Network and Quake-Catcher Network: Enabling structural health monitoring through instrumentation by community participants," *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2013, International Society for Optics and Photonics*, pp. 86923X, 2013.

- [7] J. Han, E. Owusu, L. T. Nguyen, A. Perrig, J. Zhang, "Accomplice: Location inference using accelerometers on smartphones," *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, pp. 1-9, 2012.
- [8] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," *ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 437-442, 2013.
- [9] Bayat, Pomplun and Tran, "A Study on Human Activity Recognition Using Accelerometer Data from Smartphones," *The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014)*, pp. 450-457, 2014.
- [10] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, "LSTM Networks for Mobile Human Activity Recognition," *International Conference on Artificial Intelligence: Technologies and Applications (ICAITA 2016)*, pp. 50-53, 2016.
- [11] G. M. Guzman-Acevedo, G. E. Vazquez-Becerra, J. R. Millan-Almaraz, H. E. Rodriguez-Lozoya, A. Reyes-Salazar, J. R. Gaxiola-Camacho, et al., "GPS, Accelerometer, and Smartphone Fused Smart Sensor for SHM on Real-Scale Bridges," *Advances in Civil Engineering*, vol. 2019, pp. 1-15, 2019.
- [12] A. Khan, M. Siddiqi, S. W. Lee, "Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones," *Sensors*, vol. 13, no. 10, pp. 13099-13122, 2013.
- [13] M. Arif, M. Bilal, A. Kattan, S. I. Ahamed, "Better physical activity classification using smartphone acceleration sensor," *J. Med. Syst.*, vol. 38, no. 9, pp. 95, 2014;
- [14] J. Suto, S. Oniga, P. P. Sitar, "Feature Analysis to Human Activity Recognition Raw data pre-processing and classification Feature extraction," *Int. J. Comput. Commun. Control*, vol. 12, no. February, pp. 116-130, 2017.
- [15] Y. Zhang, S. Lu, X. Zhou, M. Yang, L. Wu, B. Liu, et al., "Comparison of machine learning methods for stationary wavelet entropy-based multiple sclerosis detection: decision tree, k-nearest neighbors, and support vector machine," *Simulation*, vol. 92, no. 9, pp. 861-871, 2016.
- [16] M. Amarnath, V. Sugumaran, H. Kumar, "Exploiting sound signals for fault diagnosis of bearings using decision tree," *Measurement*, vol. 46, no. 3, pp. 1250-1256, 2013;
- [17] R. Nathan, O. Spiegel, S. Fortmann-Roe, R. Harel, M. Wikelski, W. M. Getz, "Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures," *J. Exp. Biol.*, vol. 215, no. 6, pp. 986-996, 2012.
- [18] U. Raghavendra, P. K. Mahesh, A. Gudigar, "A novel face recognition method using pca, lda and support vector machine," *International Conference on Computer Science and Information Technology*, pp. 241-249, 2012.
- [19] A. Tharwat, T. Gaber, A. Ibrahim, A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Commun.*, vol. 30, no. 2, pp. 169-190, 2017.
- [20] A. A. Matkan, M. Hajeb, S. Sadeghian, "Road extraction from lidar data using support vector machine classification," *Photogramm Eng Remote Sens.*, vol. 80, no. 5, pp. 409-422, 2014;
- [21] H. Bhavsar, M. H. Panchal, "A review on support vector machine for data classification," *Int J Adv Res Comput Eng Technol.*, vol. 1, no. 10, 2012.
- [22] X. Ma, Y. Zhang, Y. Wang, "Performance evaluation of kernel functions based on grid search for support vector regression," *2015 IEEE 7th international conference on cybernetics and intelligent systems (CIS) and IEEE conference on robotics, automation and mechatronics*, pp. 283-288, 2015.
- [23] B. Yu, X. Song, F. Guan, Z. Yang, B. Yao, "K-Nearest neighbor model for multiple-time-step prediction of short-term traffic condition," *J Transp Eng.*, vol. 142, no. 6, pp. 4016018, 2016.
- [24] L. H. Lee, C. H. Wan, R. Rajkumar, D. Isa, "An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization," *Appl Intell.*, vol. 37, no. 1, pp. 80-99, 2012.
- [25] J. Vitola, F. Pozo, D. Tibaduiza, M. Anaya, "A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications," *Sensors*, vol. 17, no. 2, p. 417, 2017.

BIOGRAPHIES OF AUTHORS



Hapsoro Agung Nugroho is an Assistant Professor at the Department of Instrumentation for Meteorology, Climatology and Geophysics (MKG), School of Meteorology Climatology and Geophysics, Jakarta, Indonesia. He received Bachelor Degree in Electrical Engineering from Udayana University, Bali, Indonesia. He received Master Degree in Instrumentation and Control from Bandung Institute Technology, Bandung Indonesia, in 2012. His interest area are machine learning, intelligent instrument, and pattern recognition.



Haryas Subyantara Wicaksana, is a data analyst in Agency for Meteorology, Climatology and Geophysics, Jakarta, Indonesia. He received Bachelor Degree in Department of Instrumentation for Meteorology, Climatology and Geophysics (MKG), School of Meteorology Climatology and Geophysics, Jakarta, Indonesia, in 2019. His interest area are human activity recognition, intelligent instrument, and machine learning.



Hariyanto is an Assistant Professor at the Department of Instrumentation for Meteorology, Climatology and Geophysics (MKG), School of Meteorology Climatology and Geophysics, Jakarta, Indonesia. He received Bachelor Degree in Electrical Engineering from Institute of Technology Sepuluh November, Surabaya, Indonesia. He received Master Degree in Instrumentation and Control from Bandung Institute Technology, Bandung Indonesia, in 2014. His interest area are control system, meteorology instrument, and signal recognition.



Rista Hernadi Virgianto is an Assistant Professor at the Department of Climatology, School of Meteorology Climatology and Geophysics, Jakarta, Indonesia. He received Bachelor Degree in Climatology from School of Meteorology Climatology and Geophysics, Jakarta, Indonesia. He received Master Degree in Applied Meteorology and Climatology from University of Birmingham, England, in 2016. His interest area are data science, air quality instrument, and deep learning.