Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 11, No. 3, September 2023, pp. 685~699
ISSN: 2089-3272, DOI: 10.52549/ijeei.v11i3.4397

Implementation of Supervised Machine Learning on Embedded Raspberry Pi System to Recognize Hand Motion as Preliminary Study for Smart Prosthetic Hand

Triwiyanto[1], Sari Luthfiyah[1], Wahyu Caesarendra[2], Abdussalam Ali Ahmed[3]
[1] Department of Medical Electronics Technology, Poltekkes Kemenkes Surabaya, Indonesia
[2] Faculty of Integrated Technologies, Universiti Brunei Darussalam, Brunei Darussalam
[3] Department of Mechanical and Industrial Engineering, Bani Waleed University, Libya

Article Info

Article history:
Received Dec 30, 2022
Revised Jun 11, 2023
Accepted Aug 6, 2023

Keywords:
Surface EMG,
Pattern recognition,
Hand motion,
Prosthetic hand,
Embedded system,
Machine learning

ABSTRACT

EMG signals have random, non-linear, and non-stationary characteristics that require the selection of the suitable feature extraction and classifier for application to prosthetic hands based on EMG pattern recognition. This research aims to implement EMG pattern recognition on an embedded Raspberry Pi system to recognize hand motion as a preliminary study for a smart prosthetic hand. The contribution of this research is that the time domain feature extraction model and classifier machine can be implemented into the Raspberry Pi embedded system. In addition, the machine learning training and evaluation process is carried out online on the Raspberry Pi system. The online training process is carried out by integrating EMG data acquisition hardware devices, time domain features, classifiers, and motor control on embedded machine learning using Python programming. This study involved ten respondents in good health. EMG signals are collected at two lead flexor carpi radialis and extensor digitorum muscles. EMG signals are extracted using time domain features (TDF) mean absolute value (MAV), root mean square (RMS), variance (VAR) using a window length of 100 ms. Supervised machine learning decision tree (DT), support vector machine (SVM), and k-nearest neighbor (KNN) are chosen because they have a simple algorithm structure and less computation. Finally, the TDF and classifier are embedded in the Raspberry Pi 3 Model B+ microcomputer. Experimental results show that the highest accuracy is obtained in the open class, 97.03%. Furthermore, the additional datasets show a significant difference in accuracy (p-value <0.05). Based on the evaluation results obtained, the embedded system can be implemented for prosthetic hands based on EMG pattern recognition.

Corresponding Author:
Triwiyanto,
Department of Medical Electronics Technology,
Poltekkes Kemenkes Surabaya, Indonesia,
Jl. Pucang Jajar Timur No. 10, Surabaya, Indonesia.
Email: triwiyanto@ieee.org

1. INTRODUCTION

Surface electromyography signal (EMG) is a bioelectric signal generated by muscles during contraction [1][2]. EMG signals are widely used to develop rehabilitation equipment, including prosthetic hands [3][4]. Recently, the development of prosthetic hand has progressed rapidly in terms of controller technology [5], sensor system [6], power management [7], and 3D printing design [8][9]. An EMG-based smart prosthetic hand is expected to recognize EMG signal patterns for different movements and adapt to new users. However, EMG signals have a random and stochastic nature [10] that tends to change every time, even with the same contraction. Furthermore, EMG signals are non-linear and non-stationary [11]–[14] with respect to the dynamics of hand movements, which requires in-depth investigation into feature extraction and machine learning. A computer machine capable of EMG signal data acquisition, time domain feature extraction, pattern recognition implementation, and real-time motor control is still a challenge for researchers to develop prosthetic hands based on EMG pattern recognition [15]–[18]. An embedded system based on a microcontroller or microcomputer will be the answer to developing a portable smart prosthetic hand. However, prosthetic hands based on EMG pattern recognition mostly used another system to train and evaluate the model [19][20][7]. Therefore, this research focuses on implementing EMG data recording and EMG feature extraction, performing supervised machine learning, and controlling servo motors in real-time or online training on a Raspberry Pi microcomputer.

A prosthetic hand is generally classified into prehension and anthropomorphic hand models [21]–[24]. The prehension hand prosthetic model can only perform simple hand close and open movements with two fingers. In comparison, the anthropomorphic hand prosthetic model has a movement function close to humans with five fingers. Prosthetic hand anthropomorphic design is more widely used because it is cosmetically close to the human hand [25]. Simple prosthetic hand movement control can be done using an open and close switch. However, in development, this will depend on the normal part of the hand to press the open and close buttons to hold or release the object. Another prosthetic hand development is to use shoulder control. However, this will interfere with real-time operation as it requires coordination between the shoulder muscles, biceps, flexor carpi radialis longus and extensor muscles. On the other hand, prosthetic hand development using EMG signal control is widely applied because EMG signals have a faster response than mechanical sensors [26]. However, a prosthetic hand with an EMG signal based on threshold control will result in a model that cannot adapt to the new user [27]. Therefore, this research develops a smart prosthetic hand based on EMG pattern recognition that can recognize new users.

In the EMG pattern recognition process, the EMG signal is cut into several windows with a certain window length which is then carried out in the feature extraction process. Feature extraction of EMG signals can be done using time domain [28][29], frequency domain [30]–[32] and time-frequency (wavelet) [33]domain. However, feature extraction based on the time domain is more often used because it has a faster computation time than others [31]. Time domain features are generally grouped into those based on energy, wave complexity, and frequency change [34][35]. Furthermore, energy-based time domain features such as root mean square (RMS), mean absolute value (MAV), integrated EMG (iEMG), variance (VAR), average amplitude change (AAC), and Difference Absolute Standard Deviation Value (DASDV) are more widely used for feature extraction process. Therefore, this research will utilize energy-based feature extraction.

The classifier machine is the main part of the smart prosthetic hand. The classifier will recognize hand movement patterns based on the EMG signal feature available at the previous stage. Several researchers have applied various supervised machine learning for EMG pattern recognition, including support vector machine (SVM) [36], decision tree (DT) [37][38], Bayesian (BY) [39], random forest (RF)[40], k-nearest neighbor (k-NN) [41][42], and artificial neural network (ANN) [43]. Machine classifiers can solve the problem of non-linearity and randomness of EMG signals. Some modern machine learning is also applied to developing prosthetic hands based on EMG pattern recognition, including deep learning classifiers [44][24]. Gabriel et. al used a support vector machine to classify rest, flexion, extension, and grasp movements with an accuracy of 84.93% for real-time mode [45]. In addition, EMG pattern recognition was performed using two lead electrodes, the flexor and extensor muscles.

However, in that study, the classifier model was implemented on a computer system and not yet implemented on a prosthetic hand device. Other researchers explored machine learning to find out better accuracy, as done by Geethanjali et al. [37]. In the study, the author investigated seven types of classifiers including decision tree (DT), feature ensemble (FE), linear discriminant analysis (LDA), logistic model tree (LMT), neural network (NN), simple logistic regression (SLR), and support vector machine (SVM). The investigation results show that each classifier has different mean error values ranging from 8 to 30% where the lowest mean error is obtained in the linear discriminant analysis (LDA) classifier. In this study, the implementation of embedded machine learning is carried out on the TMS320F28335 microcontroller board but the training process is carried out outside the system using the MATLAB application. An embedded machine learning that can perform training and testing processes on microcontroller or minicomputers online will be an added value because all processes are carried out in one system. Fajardo developed a prosthetic hand with artificial neural network (ANN) implementation to ARM Cortex-M4 microcontroller to recognize five hand gestures [46]. The results of EMG signal pattern recognition based on five gestures obtained an accuracy value of 86%. However, in this study, the ANN training process is carried out using the MATLAB application. Furthermore, the machine learning model that has been trained and tested with MATLAB is then converted to C language format to be implemented into a microcontroller system. Triwiyanto et al. have developed a prosthetic hand to recognize four gesture patterns of open, close, wrist supination, and wrist pronation using embedded raspberry Pi [47]. However, the training process was performed in the computer system with the same environment programming language (Python). Cabegin et al. successfully implemented machine learning to the Raspberry Pi microcomputer system. The research implemented principal component analysis (PCA) and support vector machine (SVM) to classify two hand close and hand open gestures with an accuracy of 99.7%. However, in the implementation of prosthetic hands, some gestures that are common in everyday life are needed, such as curve and pinch gestures.

To the best of our knowledge, researchers related to implementing machine learning in microcontroller or microcomputer systems are still rare. Prosthetic hand research based on EMG pattern recognition that applies EMG data acquisition, time domain feature extraction, classifier, and motor control embedded in an embedded system is still an interesting research topic. Furthermore, the training and testing process on the embedded system will be an added value to the developed system so that it does not require another system. Therefore, this research aims to implement EMG pattern recognition on an embedded Raspberry Pi system to recognize hand motion as a preliminary study for a smart prosthetic hand. The system we develop is expected to perform training and evaluation time domain feature extraction and machine learning processes directly on the Raspberry Pi system by implementing Python programming. This study will train the prosthetic hand to recognize four gestures: open, close, pinch, and curve. These four main gestures are often used in everyday life.

2. MATERIALS AND METHOD

2.1. Subjects.

This research is a preliminary study on the development of a smart prosthetic hand. Therefore, the subjects in this study are healthy humans with the characteristics of 20.3±2.6 years old and a body weight 60.4±5.5 kg, totaling 20 people. The subjects involved had no recorded physical injuries to the hand or arm, and the subjects had no history of other serious illnesses. After reading the informed consent form provided, subjects agreed to be involved in the data collection process. This study has undergone an Ethical Clearance examination from Health Polytechnic Surabaya Ethics Committee.

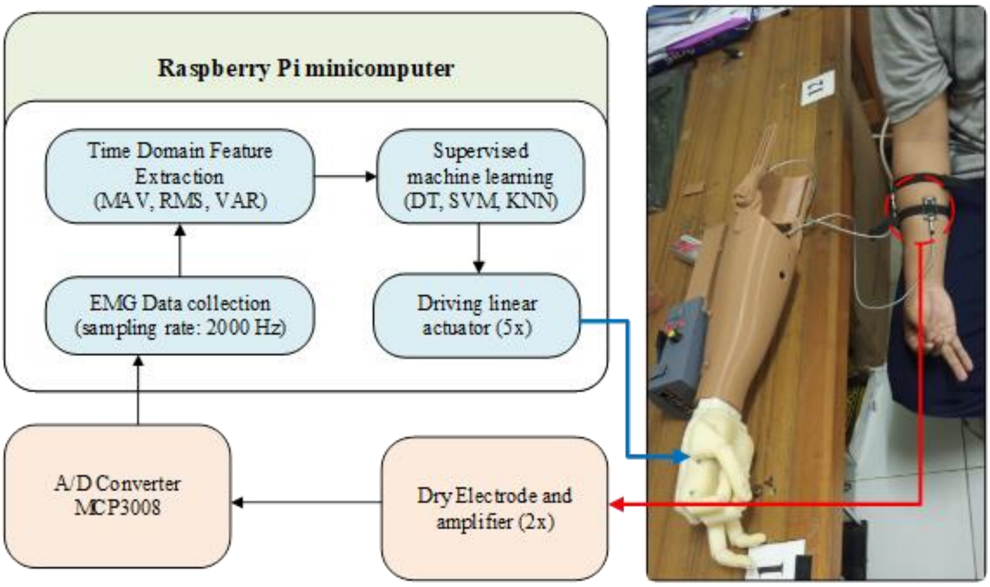

Figure 1. Block diagram of the machine learning embedded on Raspberry Pi to control the upper limb exoskeleton built using 3D printing technology.

2.2. System Operational.

This research uses Raspberry Pi (3B+, Quad-core A53 (ARMV8), US) as the main part of the system. The software for the pre-evaluation process of the system is using Python programming version 3 using Anaconda Navigator and Spyder IDE (version 1.10.0, 2016, Anaconda, Inc) running on a computer machine (Windows, core I5, SDRAM 8 MB). Software for the implementation of machine learning, feature extraction and prosthetic hand control system using Python programming (Thony IDE, Python 3.7). EMG signals are tapped at the flexor carpi radialis (CH1), and extensor digitorum (CH2) points using a dry electrode [48][49][50] (Dfrobot, Oymotion, China) with sensor dimensions of 22mm x 35 mm and weight of 36 grams. The two muscles produce considerable contractions when respondents perform hand open, hand close, pinch and curve movements. The EMG signal is then converted to digital form using an A/D converter MCP3008 (Microchips, USA). MCP3008 is an A/D with high performance, low power consumption, 10 bit resolution with 200k samples/second, and 8 input channels. Two EMG channels are recorded by the Raspberry Pi device using a sampling frequency of 2000 Hz. The amount of sampling frequency has complied with the Nyquist rule [51]. Time domain feature extraction (TDFE) is applied to the EMG signal to obtain patterns of distinguishing features for the four movements to be trained (open, close, pinch, and curve). The TDFE used in this implementation are mean absolute value (MAV), root mean square (RMS), and variance (VAR). These features have been widely applied in the development of exoskeleton and prosthetic hand devices that require the system to run in real-time. After going through the TDFE process, the EMG features will be used as input for supervised machine learning (DT, SVM, and KNN). During the training process, machine learning will be introduced with four prosthetic hand movement patterns (open, close, pinch, and curve). In the implementation step, the machine learning will recognize the EMG movement patterns and give movement commands to the prosthetic hand. In more detail, the prosthetic hand is moved using a mini linear actuator (Actuonix, PQ12-R, 63:1, Canada).

2.3. Data Collection.

The EMG recording data collection process is carried out by installing a dry electrode (Dfrobot, Oymotoin, China) on the flexor carpi radialis (CH1) and extensor digitorum (CH2) muscles [1]. The skin surface at the two tapping points is cleaned first before the dry electrode is installed to remove oil and dust on the skin surface [52]. An elastic strap is attached to the electrode so that the electrode does not shift during the EMG signal recording process. In recording EMG signals, subjects are in a relaxed sitting position with their hands on their thighs. The researcher instructed the subject to perform four sequential movements: hand open, hand close, hand open, hand pinch, hand open, and hand curve. Each movement is performed within four seconds. Between the hand close, pinch, and open movements, the open hand movement is inserted for 5 seconds to provide rest time to the subject while avoiding muscle fatigue. The position of the four movements (hand close, open, curve, and pinch) are shown in Figure 2.

For the purpose of the machine learning training process, ten subjects are recorded in turn. One cycle of hand movements is open-close-open-curve-open-pinch-open. The process of the movement sequence is carried out for ten cycles. After ten cycles have been completed, the subject is instructed to rest for 5 minutes. After that, the process was repeated five times. To keep the rhythm of the four hand movements consistent and fixed, the subject must follow the guidance through a metronome application with a period setting of 5 seconds. The EMG recording data is grouped based on the subjects' names, with the fields being time, channel 1, and channel 2, which are saved to a file in CSV format. The continuous EMG recording data is labeled according to the movement performed (open, close, pinch, and curve).

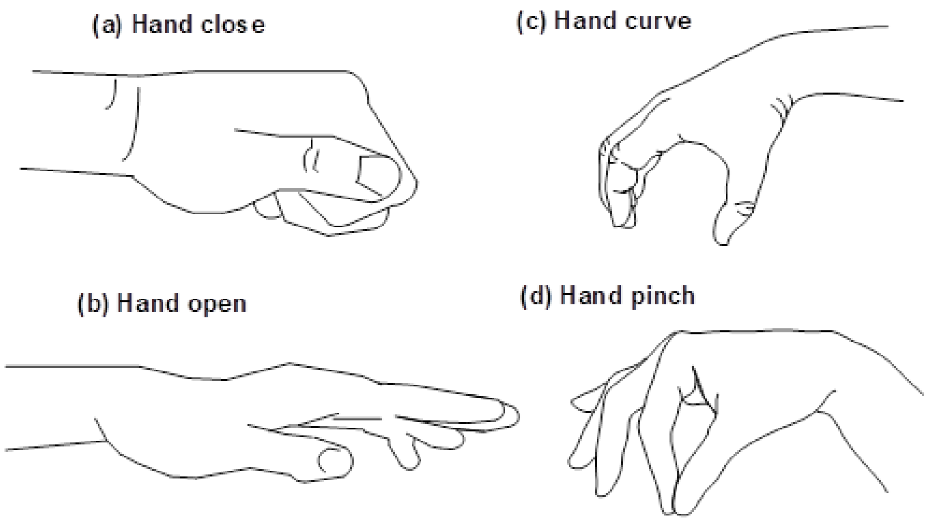

Figure 2. Hand motion (a) close, (b) open, (c) curve, and (d) pinch to train the prosthetic hand.

.

2.4. Data Processing.

The results of EMG signal recordings in CSV files from 10 subjects are processed using time domain feature extraction. In the time domain feature extraction (TDFE) evaluation stage, the calculation process is carried out offline using a computer device. The three EMG features evaluated at this stage are VAR, RMS, and MAV.

2.4.1. Time Domain Features Extraction.

Researchers still apply this feature widely, especially in developing real-time systems. Time domain features have advantages in terms of simple mathematical equations so that the computational time required to extract EMG signals is shorter. At this stage, the three EMG features evaluated are variance (VAR), root mean square (RMS), and mean absolute value (MAV), as shown in equations (1), (2), and (3). The variance of EMG (VAR) is the average power value of the EMG signal. VAR is formulated as follows [53]

$$VAR = \frac{1}{N-1}\sum_{i=1}^{N} x_i^2 \tag{1}$$

Mean Absolute Value (MAV) is an average of absolute EMG signal for *N* window length. The MAV is formulated as [53]:

$$MAV = \frac{1}{N}\sum_{i=1}^{N} | x_i | \tag{2}$$

Root Mean Square (RMS) represents the mean power of signal over a window length of EMG samples. The mathematical equation to describe this feature is written as follows [53]

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{3}$$

where x_i indicates the *i*-th EMG signal and *N* indicates the length of the EMG signal.

2.5. Machine Learning.

Conventional machine learning is still widely used in some simple and real-time applications; this is due to the simplicity of the model used and ease of implementation on various platforms, including embedded microcontroller systems or Raspberry Pi mini computers. One of the advantages of using conventional machine learning is the fast processing and training time. There is several supervised machine learning which will be evaluated in this study, including decision tree (DT), K-nearest neighbor (KNN), and support vector machine (SVM).

2.5.1. Decision Tree.

A decision tree is the simplest machine learning for classification and regression purposes. The classification step is done by splitting the data according to the branches of the decision tree. The classification process is carried out by calculating the Entropy(s) value of the probability of the results of the decision tree that has been carried out using equation (4) [54].

.

$$Entropy(s) = -\sum_{c \in C} p(c) Log_2 p(c) \tag{4}$$

where S represents the data set that entropy is calculated, c represents the classes in set S, and p(c) represents the proportion of data points that belong to class c to the number of total data points in set S. Attribute selection is done by looking at the highest information gain value, the calculation of information gain (S,a) is done according to equation (5) [54] .

$$InformationGain(S,a) = Entropy(S) - \sum_{v \in value(a)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{5}$$

Where *a* represents a specific attribute or class label, *Entropy(S)* is the entropy of dataset, S; |Sv|/ |S| represents the proportion of the values in S_v to the number of values in dataset, S; *Entropy*(S_v) is the entropy of dataset, S_v. Furthermore, a *Gini* parameter is used to determine how well a decision tree classifier classifies the data. *Gini* is measured using equation (6) [54].

$$Gini = 1 - \sum_{v \in value(a)} (p_i)^2 \tag{6}$$

Gini impurity is the likelihood that a random data point in the dataset would be classified wrongly if the dataset's class distribution determines its label. In this study, the max_depth was adjusted equal to ten in the decision tree algorithm design.

2.5.2. K-Nearest Neighbor.

K-nearest neighbor (KNN) is a classifier that applies classification based on the closeness of data in a particular group. The similarity of data in a group is calculated using Euclidean (7), Manhattan (8), and Minkowski [55].

$$d(x,y)=\sqrt{\sum_{i=1}^{p}(x_i-y_i)^2} \tag{7}$$

$$d(x,y)=\sum_{i=1}^{k}|x_i-y_i| \tag{8}$$

$$d(x,y)=(\sum_{i=1}^{k}(|x_i-y_i|)^q)^{\frac{1}{q}} \tag{9}$$

where, x and y are the data being measured while the d is the distance. After calculating the distance using one of the measurement methods according to equations (7), (8), and (9), then mark as a certain class based on the highest probability [55].

$$\mathrm{P}_r(Y=j\mid X=x_0)=\tfrac{1}{k}\sum_{i\in N_0} I(y_i=j) \tag{10}$$

where, Pr indicates the probability of the distance, and I indicate the label. In this study, the neighbor (k) was selected as five in the k-NN algorithm design.

2.5.3. Support Vector Machine.

Support Vector Machine (SVM) is a classifier that applies a hyperplane in a high-dimensional space used to separate a group of data. Furthermore, SVM can be used for classification and regression purposes. In general, a hyperplane can be written as a mathematical equation (1) or (2). In general, the boundary of a hyperplane for classification purposes can be described through hyperplane equations (13), (14) and (15) [56].

$$h(x)=w^T x+b \tag{11}$$

$$h(x)=w^T x_1+w^T x_2+...+w^T x_p+b \tag{12}$$

$$w^T x+b=0 \tag{13}$$

$$w^T x+b=+1 \tag{14}$$

$$w^T x+b=-1 \tag{15}$$

where w is hyperplane weight, and b is intercept of the hyperplane. The optimization function is used to determine local minima where the normalization function is shown in equation (16). The final optimization function is shown in equation (17) [56].

$$(x_1-x_2)=\frac{2}{\|w^T\|} \tag{16}$$

$$f(x)=\frac{\|w^T\|}{2}+C_i\sum_{i=1}^{n}d_1 \tag{17}$$

where, h(x)=0 then x lies on the hyperplane; when h(x) <0 or h(x)>0 then x fall to one side of the hyperplane. Furthermore, an illustration depicting the classification process of two groups using SVM classifier is shown in equation (3).

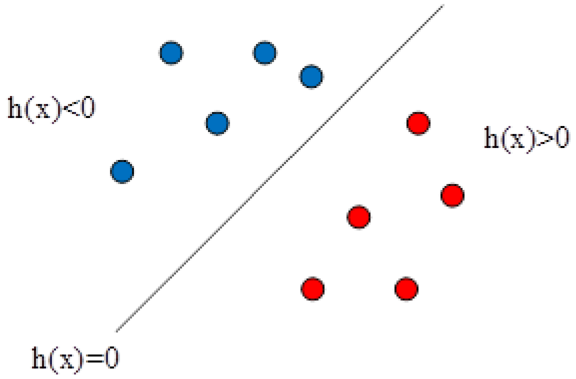

Figure 3. Support vector machine with hyperplane-based boundary method

In this research, the training and testing process is carried out online and offline using either the system on Raspberry Pi or a microcomputer. The training process that we applied is 80% training and 20% testing. Furthermore, we selected a linear kernel function in the SVM algorithm. After the initialization process, the program will call several threads, including data acquisition, feature extraction, machine learning and driving linear actuator. In this research, the thread programming model is applied. It aims to run the program in parallel or concurrently. Therefore, all variables a thread uses must be declared globally so all threads can use them together. The thread programmed will run continuously as long as there is no reset on the Raspberry Pi system. Overall, the embodied machine learning and time domain feature extraction (TDFE) system is shown in Figure 4.

2.6. Online Training.

At the evaluation stage, the time domain feature and machine learning investigation were carried out on the computer system; this is to get the best type of features and classifier. Furthermore, after getting the right features and classifiers, then the feature and machine learning are implemented directly into the Raspberry Pi system using Python-based programming through the Thony application. Through the Raspberry Pi device, the system performs the process of data acquisition, feature extraction, classification and moving the prosthetic hand directly. The training and testing process is carried out online on a Raspberry Pi device.

2.7. Statistical Analysis.

Several variables are statistically tested to see whether they had a significant effect on accuracy or not. The T-test statistic is applied to see if there is a significant difference in accuracy when number of the dataset is different. In this statistical test, we used alpha=0.05.

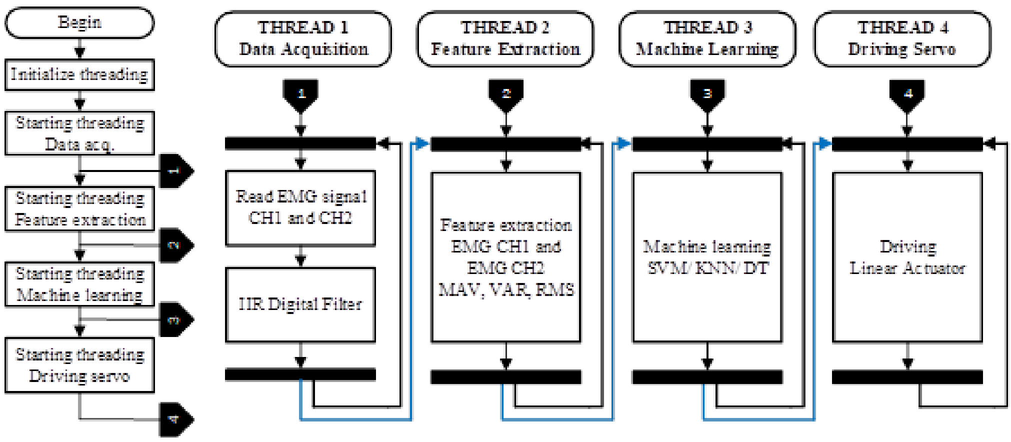

Figure 4. Flowchart controls the upper limb exoskeleton, divided into four parts: data acquisition, feature extraction, machine learning and driving the servo motor.

Machine learning decision tree (DT), k-nearest neighbor (KNN), and random forest (RF), along with time domain feature extraction, are successfully embedded on a Raspberry Pi machine to classify movement patterns. The highest accuracy results are obtained in the KNN classifier, with an accuracy of 94.06%. Other details related to the form of raw EMG signal, features and confusion matrices will be explained in more detail in the following sub-sections.

3. RESULT

3.1. Raw EMG Signal.

EMG signals are tapped at the flexor carpi radialis (CH1) and extensor digitorum (CH2) points. These two muscles are responsible for performing flexion and extension movements. EMG signals have a random shape with different signal patterns for each different form of movement. Figure 3 shows the EMG signal pattern during close, open, pinch, open, curve, and open movements. EMG signals generated by CH1 and CH2 have the same signal pattern when the hand performs close-open-pinch movements, but when the hand performs curve movements, the CH1 EMG signal produces a large enough amplitude (20 mV) compared to the CH2 EMG signal (1mV).

3.2. Time Domain EMG Features.

The output of EMG features produces amplitudes that vary depending on the type of time domain feature used. In this study, the VAR feature has the highest amplitude compared to the other two features (MAV and RMS). The VAR feature is shown with a red line plot color, while the MAV and RMS features are shown with blue and black line plot colors, respectively. Although each EMG feature shows different amplitudes for each movement pattern, the pattern of each feature shows similarities, as shown in Figure 6. Furthermore, Figure 7 shows the calculation results of time domain feature extraction for EMG signals tapped on CH2. The feature patterns show that there are similarities in hand close, open and pinch movements for both CH1 and CH2. However, when the hand performs curve movements, it appears that the EMG feature pattern shows different activity.

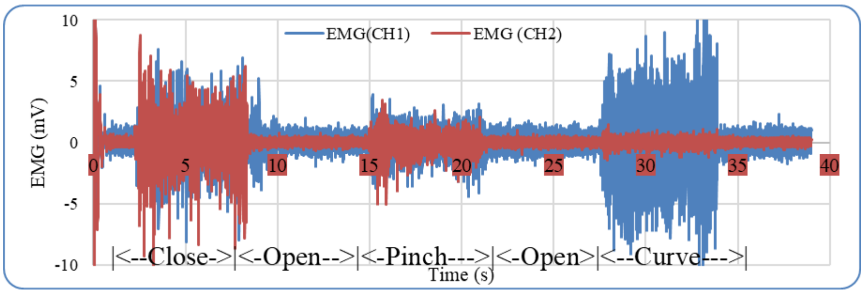

Figure 5. Raw EMG signals from the flexor carpi radialis (CH1) and extensor digitorum (CH2) points when the respondent performed close-open-pinch-curve motion.

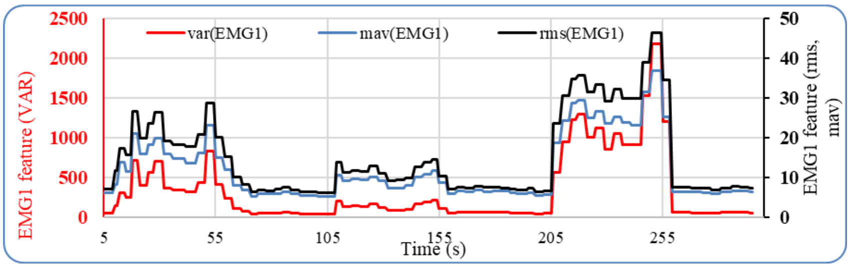

Figure 6. EMG (var, mav, and rms) feature outputs from the flexor carpi radialis (CH1).

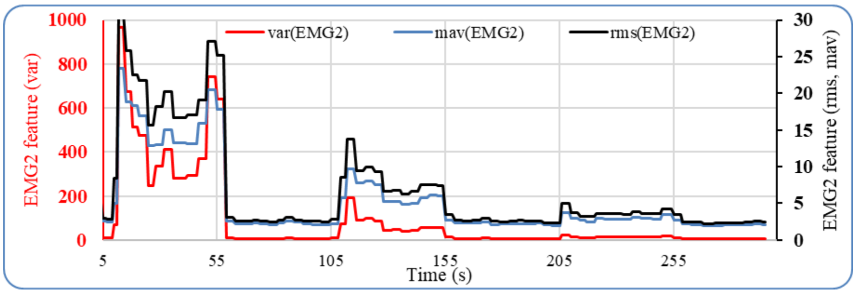

Figure 7. EMG (var, mav, and rms) feature outputs from the flexor carpi radialis (CH2).

3.3. Scatter Plot.

Scatter plot is used to see if a feature is able to classify and group data according to its class or attribute (Figure 8). This study compares features between channels (CH1 and CH2) to see the separation between labels or classes (open, close, curve, and pinch). The Euclidean calculation results for class hand close, open, pinch and curve are 3.70±3.14, 4.26±0.18, 3.20±1.60, and 23.43±4.58, respectively.

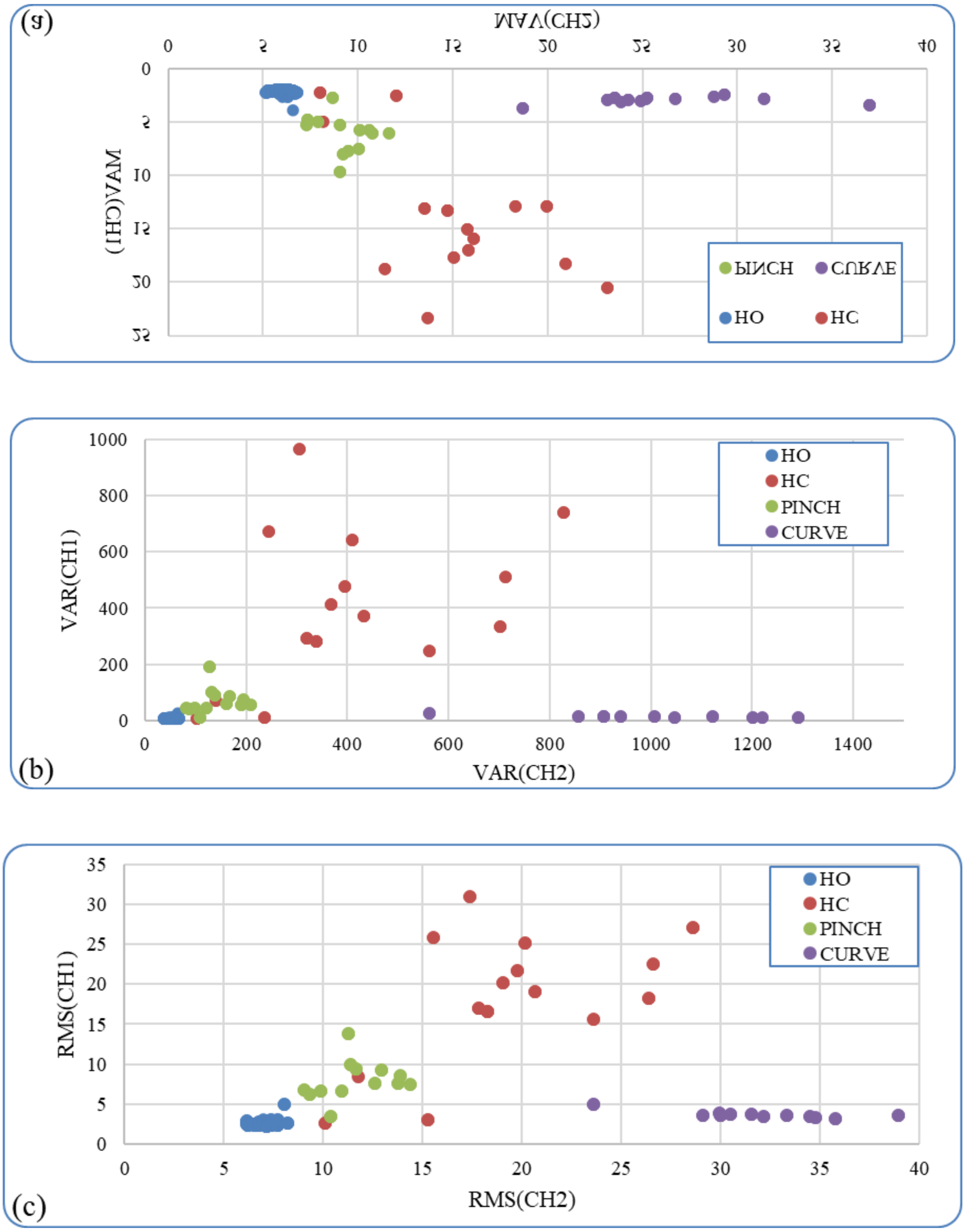

Figure 8. Scater plot between CH1 and CH2 for each feature, a) MAV, b) VAR, and c) RMS

3.4. Machine Learning Accuracy.

Machine learning produces different accuracy for each tested motion model (open, close, pinch, and curve). Figure 9 shows the confusion matrices for the decision tree (DT), support vector machine (SVM), and k-nearest neighbor (KNN) machine learning using the RMS feature.

TRUE	Open	Close	Pinch	Curve
Open	92.08	0.00	3.96	3.96
Close	1.18	87.06	0.00	11.76
Pinch	1.03	1.37	97.26	0.34
Curve	0.89	5.36	3.57	90.18

ACC. 100 0

(a) DT PREDICTED

TRUE	Open	Close	Pinch	Curve
Open	86.14	0.00	11.88	1.98
Close	1.18	89.41	0.00	9.41
Pinch	0.68	1.03	97.26	1.03
Curve	2.68	6.25	1.79	89.29

ACC. 100 0

(b) SVM PREDICTED

TRUE	Open	Close	Pinch	Curve
Open	97.03	0.00	1.98	0.99
Close	1.18	94.12	0.00	4.71
Pinch	1.71	1.71	96.58	0.00
Curve	2.68	8.93	3.57	84.82

ACC. 100 0

(c) KNN PREDICTED

Figure 9. Machine learning produces different accuracy for each tested motion model (open, close, pinch, and curve).

In the DT machine learning, the pinch movement has the highest accuracy compared to the others (97.26%), while the movement with the lowest accuracy is the close movement (87.06%). Some confusion matrices columns show small accuracy values ranging from 0.00 to 11.76%. This indicates machine learning errors in classification. For example, in Figure 9(a), the machine learning has misclassified what should be detected as close but is identified as a curve (11.76%). Figure 9(b) shows the lower accuracy generated by SVM machine learning. Furthermore, the accuracy generated by each motion varies. The highest accuracy is generated by the pinch motion (97.26%). The highest misclassification is generated by the open gesture, where the open gesture is detected as a pinch (11.88%). On the other hand, KNN machine learning produces higher accuracy than the others (Figure 9(c)). Three out of four gestures produce accuracy >90%. The open movement produces the highest accuracy (97.03%) compared to the other movements. In this machine learning, the curve movement is misclassified as a close movement (8.93%). Figure 9 shows the confusion matrices for the decision tree (DT), support vector machine (SVM), and k-nearest neighbor (KNN) machine learning using the RMS feature. In the DT machine learning, the pinch movement has the highest accuracy compared to the others (97.26%) while the movement that has the lowest accuracy is the close movement (87.06%). Some confusion matrices columns show small accuracy values ranging from 0.00 to 11.76%. These accuracy values indicate the machine learning errors in classification. For example, in Figure 9(a), the machine learning has misclassified what should be detected as close, but is identified as a curve (11.76%). Figure 9(b) shows the lower accuracy generated by SVM machine learning. Furthermore, the accuracy generated by each motion varies. The highest accuracy is generated by the pinch motion (97.26%). The open gesture generates the highest misclassification, where the open gesture is detected as a pinch (11.88%). On the other hand, KNN machine learning produces higher accuracy than the others (Figure 9(c)). Three out of four gestures produce accuracy >90%. The open movement produces the highest accuracy (97.03%) compared to the other movements. In this machine learning, the curve movement is misclassified as a close movement (8.93%). Overall, k-NN machine learning produces higher accuracy (94.06%) compared to the others, as shown in Figure 10. Furthermore,

Figure 10 shows the difference in machine learning accuracy for datasets derived from ten respondents and one respondent. It can be seen that the machine learning training using ten respondents produces higher accuracy than the machine learning training using one respondent. When DT machine learning is applied, it can be seen that ML accuracy using ten respondents produces better accuracy (93.56%). Similarly, when ML using SVM is applied, the accuracy for the ten respondents and one respondent groups are 92.71% and 88.7%, respectively.

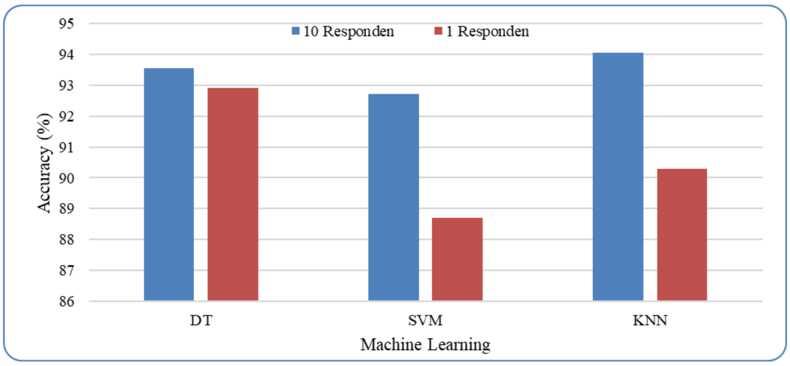

Figure 10. Various accuracies result from machine learning and different numbers of respondents.

Table 1. Statistical T test for accuracy with 10 respondents and 1 respondent

T TEST: Equal Variances			Alpha=0.05		
	std err	t-stat	p-value	t-crit	sig
One Tail	1.286	2.186	0.047	2.132	yes

This study compares the three-machine learning with different datasets (ten respondents and one respondent), namely, whether there is a significant difference in accuracy. Statistical T-test one tail with alpha=0.05 is set in this study to see if there is a significant difference. The T-test results show that the resulting p-value is 0.047, (p-value <0.05; significant = yes) as shown in Table 1.

4. DISCUSSION

This research can implement machine learning and feature extraction to the embedded system with good accuracy of 94.06% (KNN). The EMG signal measurement appears that the open-close-pinch-curve movement can be distinguished well using the flexor carpi radialis (CH1) and extensor digitorum (CH2) muscles. EMG signals originating from the extensor digitorum muscle (CH2) can distinguish open-close-pinch movements but cannot distinguish curve movements. On the other hand, EMG signals derived from the flexor carpi radialis muscle show significant signal activity when the hand performs curve gearing. So, a combination of the two muscles is needed to distinguish the four basic movements. EMG signals are very susceptible to 50 Hz mains frequency noise; therefore, all equipment supplied using voltage from the mains connected to the system must be disconnected during the recording process. Dry electrodes must be properly installed at the tapped point because a less tight installation can cause noise artifacts that affect when the subject moves the hand. Muscle fatigue also influences EMG signals as revealed by previous researchers. When muscles experience fatigue, the amplitude of the EMG signal will increase, and there is a shift in the median value of the EMG signal spectrum. Therefore, it is very important that during the data collection process, the subject is given a certain amount of rest time (at least 5 minutes) to prevent muscle fatigue. Determining the combination of EMG features as machine learning input is very important because this will determine the success of machine learning in classification. One way is by visualizing the class data using a scatter plot. The scatter plot shown in Figure 6 has different clustering variations depending on the combination of features used. Figure 6(c) proves that the RMS feature has the smallest average Euclidean value (3.20±1.60) compared to other features, indicating that this feature has the potential to be a good feature compared to other features. The pinch and open classes have a close distance between clusters, leading to misclassification in machine learning. Furthermore, the scatter plot of Figure 8(c) shows that some close hand features are mixed with hand pinch features. This may lead to misclassification between hand close and pinch movements. On the other hand, the MAV and RMS features show the intersection between three classes, namely hand open, close, and pinch. These slices can cause misclassification and lower the overall accuracy.

Confusion matrices are used to see each class's accuracy, as shown in Figure 9, where the vertical column shows TRUE, and the horizontal column shows PREDICTED. In Figure 9, the diagonal column shows the actual accuracy value which is marked with a darker color for accuracy close to 100%. Color degradation close to 0% is marked with a lighter color. For example, the open class can be classified with an accuracy of

97.03%, but the open class is misclassified as close, pinch, and curve with accuracies of 0.00, 1.98, and 0.99%, respectively. Ideally, a classifier should produce an accuracy value of 0% when tested with other classes. In this study, a considerable incidence of misclassification occurred in the curve class, where the curve class is predicted as the cure class with an accuracy value of 84.82%. However, machine learning experienced prediction errors where the class curve is detected as open, close, and pinch class with the accuracy of 2.68%, 8.93%, and 3.57%, respectively.

Research on prosthetic hands using EMG pattern recognition based on embedded machine learning has been carried out by several previous researchers with some similarities and differences. Fajardo developed a prosthetic hand using EMG pattern recognition to classify 5 different movements implemented on the ARM cortex M4 microcontroller with 86% accuracy [46]. To increase accuracy, the researcher uses multi-modal sensors, namely touch screen display and voice control, to control prosthetic hands with accuracy approaching 98%. On the other hand, Roy uses the Raspberry Pi with the Pi camera on the prosthetic hand to recognize five different grasp patterns using the Deep Neural Network (DNN) with an uncertainty of ±1cm [57]. However, this research requires a fairly heavy computation because image pattern recognition exists in the classifier.

This research is a preliminary study for developing a smart prosthetic hand by implementing machine learning to Raspberry Pi embedded system. In this initial stage, researchers evaluated SVM, KNN, and DT classifiers, simple supervised classifiers, and fast computation time. Feature extraction MAV, VAR, and RMS are applied for the EMG feature extraction process as classifier input. In this research, the process of collecting datasets and training on machine learning is still carried out offline using a personal computer, and then the training results are applied to the Raspberry Pi embedded system. Making a prosthetic hand must be packaged so that the electronic hardware system can be stored in a compartment inside the prosthetic hand. Making compact hardware is still a challenge for researchers, so later, portable and small hardware can be realized. The prosthetic hand in this research uses five linear actuators, each of which requires 210 mA power per actuator. Therefore, the operation of this prosthetic hand requires a large enough current consumption for five actuators, which is 1500 mA. A battery with a large enough capacity and good power management is needed in this research to keep the prosthetic hand operational for a long time.

The results of this research are expected to be used to help a trans-radial amputee who has a hand amputation caused by a certain disease, congenital birth, and work accident. The design of a prosthetic hand using 3D printing technology can produce a prosthetic hand that is cheap, lightweight and strong. Through the development of the smart prosthetic hand, it is expected to realize a prosthetic hand that is cheap, lightweight, and can function well as a replacement for the missing hand.

5. CONCLUSION

This research aims to implement supervised machine learning on the embedded raspberry pi system to recognize hand motion as a preliminary study for a smart prosthetic hand. The findings obtained in this research are that k-NN machine learning produces better accuracy (94.06%) compared to other classifiers (DT and SVM). Furthermore, based on the results of investigating the time domain features used, the RMS feature produces good accuracy compared to other features (MAV and VAR). Confusion matrices shows that when the classifier applies KNN and features RMS, the highest reading accuracy for the open class is 97.03%. Machine learning test results for different datasets show that there is a significant difference in accuracy (p-value <0.05), where the larger number of datasets results in better accuracy. Furthermore, some developments that can be done for future work include using other types of Raspberry Pi that are more portable so that a compact prosthetic hand can be developed. Exploration and investigation of machine learning and other types of feature extraction need to be done to get the best accurate results.

REFERENCE

[1] F. H. Martini, J. L. Nath, and E. F. Barholomew, *Fundamental of Anatomy and Physiology*, 9th ed. Boston: Pearson Education, 2012.

[2] E. A. Clancy, L. Liu, P. Liu, and D. V. Z. Moyer, "Identification of constant-posture EMG-torque relationship about the elbow using nonlinear dynamic models," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 1, pp. 205–212, 2012.

[3] H. Feys *et al.*, "An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation," *IEEE Trans. Haptics*, vol. 8, no. 2, pp. 924–929, 2018.

[4] A. Pascual-Leone, "Robot-assisted Therapy in Stroke Rehabilitation.," *J. stroke*, vol. 15, no. 3, pp. 174–81, 2013.

[5] R. G. E. Clement, K. E. Bugler, and C. W. Oliver, "Bionic prosthetic hands: A review of present technology and future aspirations," *Surg.*, vol. 9, no. 6, pp. 336–340, Dec. 2011.

[6] T. Triwiyanto, T. Rahmawati, I. P. A. Pawana, and L. Lamidi, "State-of-the-Art Method in Prosthetic Hand Design : A Review," *J. Biomimetics, Biomater. Biomed. Eng.*, vol. 50, pp. 15–24, 2021.

[7] D. Brunelli, A. M. Tadesse, B. Vodermayer, M. Nowak, and C. Castellini, "Low-cost wearable multichannel surface

EMG acquisition for prosthetic hand control," *Proc. - 2015 6th IEEE Int. Work. Adv. Sensors Interfaces, IWASI 2015*, pp. 94–99, 2015.

[8] N. E. Krausz, R. A. L. Rorrer, and R. F. F. Weir, "Design and Fabrication of a Six Degree-of-Freedom Open Source Hand," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 5, pp. 562–572, 2016.

[9] T. Triwiyanto, S. Luthfiyah, B. Utomo, and T. Hamzah, "Implementation of Prosthetic Hand Design for Transradial Amputee by Using 3D Printing Technology to Enhance Life Quality," *Front. Community Serv. Empower.*, vol. 1, no. 1, 2022.

[10] S. Lee, S. Park, J. Kim, and I. Kim, "EMG pattern recognition based on evidence accumulation for prosthesis control," *Proc. 18th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 4, pp. 1481–1483, 1996.

[11] J. Hashemi, E. Morin, P. Mousavi, and K. Hashtrudi-Zaad, "Surface EMG force modeling with joint angle based calibration," *Journal of Electromyography and Kinesiology*, vol. 23, no. 2. pp. 416–424, 2013.

[12] Y. Chen, X. Zhao, and J. Han, "Hierarchical projection regression for online estimation of elbow joint angle using EMG signals," *Neural Computing and Applications*, vol. 23, no. 3–4. pp. 1129–1138, 2013.

[13] H. J. Yu, A. Y. Lee, and Y. Choi, "Human elbow joint angle estimation using electromyogram signal processing," *IET Signal Process.*, vol. 5, no. 8, p. 767, 2011.

[14] T. Triwiyanto, O. Wahyunggoro, H. A. Nugroho, and H. Herianto, "DWT Analysis of sEMG for Muscle Fatigue Assessment of Dynamic Motion Flexion-Extension of Elbow Joint," in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2016, pp. 1–6.

[15] M. Leon, J. M. Gutierrez, L. Leija, and R. Munoz, "EMG pattern recognition using Support Vector Machines classifier for myoelectric control purposes," in *2011 Pan American Health Care Exchanges*, 2011, pp. 175–178.

[16] A. M. Simon, K. Stern, and L. J. Hargrove, "A comparison of proportional control methods for pattern recognition control," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 3354–3357, 2011.

[17] A. A. Adewuyi, L. J. Hargrove, and T. A. Kuiken, "An Analysis of Intrinsic and Extrinsic Hand Muscle EMG for Improved Pattern Recognition Control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 4, pp. 485–494, 2016.

[18] G. Gini, M. Arvetti, I. Somlai, and M. Folgheraiter, "Acquisition and analysis of EMG signals to recognize multiple hand movements for prosthetic applications," *Appl. Bionics Biomech.*, vol. 9, no. 2, pp. 145–155, 2012.

[19] A. Beneteau, G. Di Caterina, L. Petropoulakis, and J. J. Soraghan, "Low-cost wireless surface EMG sensor using the MSP430 microcontroller," *EDERC 2014 - Proc. 6th Eur. Embed. Des. Educ. Res. Conf.*, pp. 264–268, 2014.

[20] P. Chrapka, "EMG Controlled Hand Prosthesis : EMG Classification System," 2010.

[21] T. Triwiyanto, I. P. A. Pawana, T. Hamzah, and S. Luthfiyah, "Low-Cost and Open-Source Anthropomorphic Prosthetics Hand Using Linear Actuators," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 2, pp. 281–288, 2020.

[22] T. Triwiyanto, T. Hamzah, S. Luthfiyah, I. P. A. Pawana, and B. Utomo, "A low cost and open-source anthropomorphic prosthetics hand for transradial amputee," in *International Conference on Science and Applied Science (ICSAS) 2019*, 2019, vol. 020086, no. December.

[23] T. Lenzi, J. Lipsey, and J. W. Sensinger, "The RIC Arm - A Small Anthropomorphic Transhumeral Prosthesis," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 6, pp. 2660–2671, 2016.

[24] L. T. Taverne, M. Cognolato, T. Butzer, R. Gassert, and O. Hilliges, "Video-based prediction of hand-grasp preshaping with application to prosthesis control," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2019-May, pp. 4975–4982, 2019.

[25] D. Djafri, "Prinsip Dan Metode Analisis Risiko Kesehatan Lingkungan," *J. Kesehat. Masy. Andalas*, vol. 8, no. 2, p. 100, 2014.

[26] K. O. Thielbar *et al.*, "Benefits of using a voice and EMG - driven actuated glove to support occupational therapy for stroke survivors," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 3, p. 1, 2016.

[27] C. Potluri *et al.*, "SEMG based real-time embedded force control strategy for a prosthetic hand prototype," *2011 IEEE International Conference on Electro/Information Technology, EIT 2011*. 2011.

[28] Triwiyanto, O. Wahyunggoro, H. A. Nugroho, and H. Herianto, "Quantitative Relationship Between Feature Extraction of sEMG and Upper Limb Elbow Joint Angle," in *2016 International Seminar on Application for Technology of Information and Communication*, 2016, pp. 44–50.

[29] a Matsuyama and M. Jonkman, "The application of wavelet and feature vectors to ECG signals.," *Australas. Phys. Eng. Sci. Med.*, vol. 29, no. 1, pp. 13–17, 2006.

[30] M. M. Janković and D. B. Popović, "An EMG system for studying motor control strategies and fatigue," *10th Symp. Neural Netw. Appl. Electr. Eng. NEUREL-2010 - Proc.*, pp. 15–18, 2010.

[31] A. Phinyomark, A. Nuidod, P. Phukpattaranont, and C. Limsakul, "Feature extraction and reduction of wavelet transform coefficients for EMG pattern classification," *Elektron. ir Elektrotechnika*, vol. 122, no. 6, pp. 27–32, 2012.

[32] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7420–7431, 2012.

[33] L. Smital, M. Vítek, J. Kozumplík, and I. Provazník, "Adaptive wavelet wiener filtering of ECG signals," *IEEE*

Trans. Biomed. Eng., vol. 60, no. 2, pp. 437–445, 2013.

[34] Triwiyanto, O. Wahyunggoro, H. A. Nugroho, and Herianto, “An investigation into time domain features of surface electromyography to estimate the elbow joint angle,” *Adv. Electr. Electron. Eng.*, vol. 15, no. 3, pp. 448–458, 2017.

[35] U. Côté-Allard *et al.*, “Unsupervised Domain Adversarial Self-Calibration for Electromyography-Based Gesture Recognition,” *IEEE Access*, vol. 8, pp. 177941–177955, 2020.

[36] D. Yang, J. Zhao, Y. Gu, L. Jiang, and H. Liu, “EMG pattern recognition and grasping force estimation: Improvement to the myocontrol of multi-DOF prosthetic hands,” *2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009*, pp. 516–521, 2009.

[37] P. Geethanjali and K. K. Ray, “A Low-Cost Real-Time Research Platform for EMG Pattern Recognition-Based Prosthetic Hand,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1948–1955, Aug. 2015.

[38] Q. She, Z. Luo, M. Meng, and P. Xu, “Multiple kernel learning SVM-based EMG pattern classification for lower limb control,” *11th Int. Conf. Control. Autom. Robot. Vision, ICARCV 2010*, no. December, pp. 2109–2113, 2010.

[39] K. H. Knuth, “Gesture-based control and EMG decomposition,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 36, no. 4, pp. 503–514, 2006.

[40] A. Dwivedi, Y. Kwon, A. J. McDaid, and M. Liarokapis, “A Learning Scheme for EMG Based Decoding of Dexterous, In-Hand Manipulation Motions,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 10, pp. 2205–2215, 2019.

[41] W. T. Shi, Z. J. Lyu, S. T. Tang, T. L. Chia, and C. Y. Yang, “A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study,” *Biocybern. Biomed. Eng.*, vol. 38, no. 1, pp. 126–135, 2018.

[42] Y. Paul, V. Goyal, and R. A. Jaswal, “Comparative analysis between SVM & KNN classifier for EMG signal classification on elementary time domain features,” *4th IEEE Int. Conf. Signal Process. Comput. Control. ISPCC 2017*, vol. 2017-Janua, pp. 169–175, 2017.

[43] S. Pati, D. Joshi, and A. Mishra, “Locomotion classification using EMG signal,” *2010 Int. Conf. Inf. Emerg. Technol.*, pp. 1–6, 2010.

[44] M. Atzori, M. Cognolato, and H. Müller, “Deep Learning with Convolutional Neural Networks Applied to Electromyography Data : A Resource for the Classification of Movements for Prosthetic Hands,” *Front. Neurorobot.*, vol. 10, no. September, pp. 1–10, 2016.

[45] G. Eisenberg, D., K. Fyvie, and G. Abdul-khaaliq, “Real-Time Segmentation and Feature Extraction of Electromyography : Towards Real-Time Segmentation and Feature Extraction of Electromyography : Towards Real-Time Segmentation and Feature of Electromyography : Towards Control Prosthetic Hand Control of a P,” in *IFAC-PapersOnLine*, 2017, vol. 50, no. 2, pp. 151–156.

[46] J. Fajardo, A. Lemus, and E. Rohmer, “Galileo bionic hand: sEMG activated approaches for a multifunction upper-limb prosthetic,” in *2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV)*, 2015, no. Concapan Xxxv, pp. 1–6.

[47] T. Triwiyanto, B. Utomo, D. Titisari, M. Ridha Mak’Ruf, and T. Rahmawati, “Investigation of the Number of Features and Muscles for an Effective Hand Motion Classifier Using Electromyography Signal,” in *Journal of Physics: Conference Series*, 2019, vol. 1373, no. 1.

[48] B. M. Idrees and O. Farooq, “Estimation of Muscle Fatigue Using Wavelet Decomposition,” in *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, 2015, pp. 267–271.

[49] S. N. Sidek and A. J. H. Mohideen, “Measurement system to study the relationship between forearm EMG signals and wrist position at varied hand grip force,” *2012 Int. Conf. Biomed. Eng. ICoBE 2012*, no. February, pp. 169–174, 2012.

[50] T. Triwiyanto, T. Rahmawati, I. P. A. Pawana, and E. H. Maulidia, “Investigation of Electrode Location to Improve the Accuracy of Wearable Hand Exoskeleton Trainer Based on Electromyography,” *J. Biomimetics, Biomater. Biomed. Eng.*, vol. 55, pp. 71–80, 2022.

[51] A. Phinyomark and E. Scheme, “A feature extraction issue for myoelectric control based on wearable EMG sensors,” *2018 IEEE Sensors Appl. Symp. SAS 2018 - Proc.*, vol. 2018-Janua, pp. 1–6, 2018.

[52] P. Konrad, *The ABC of EMG A Practical Introduction to Kinesiological Electromyography*, no. April. 2005.

[53] R. H. Chowdhury, M. B. I. Reaz, M. A. B. M. Ali, A. a a Bakar, K. Chellappan, and T. G. Chang, “Surface electromyography signal processing and classification techniques.,” *Sensors (Basel).*, vol. 13, no. 9, pp. 12431–12466, 2013.

[54] S. Ben-David and S. Shalev-Shwartz, *Understanding Machine Learning: From Theory to Algorithms*. 2014.

[55] W. Richert and L. P. Coelho, *Building Machine Learning Systems with Python*. .

[56] R. Berwick, “An Idiot’s Guide to Support vector machines (SVMs): A New Generation of Learning Algorithms Key Ideas,” *Village Idiot*, pp. 1–28, 2003.

[57] R. Roy and N. Kianoush, “A low-cost Raspberry PI-based vision system for upper-limb prosthetics,” in *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2020, pp. 20–23.

ACKNOWLEDGMENTS

This works is supported by Poltekkes Kemenkes Surabaya from PTUPT research fund

BIOGRAPHY OF AUTHORS

Dr. Triwiyanto received the B.S. degree in Physics from Airlangga University, Indonesia, M.S. degrees in Electronic Engineering from the Institut Teknologi Sepuluh Nopember Surabaya, Indonesia in 2004, and the Ph.D. degree in Electrical Engineering from Gadjah Mada University, Yogyakarta, Indonesia, in 2018. From 1998 to 2004, he was a Senior Lecturer with the Microcontrollers Laboratory. Since 2005, he has been an Assistant Professor with the Medical Electronics Technology Department, Health Polytechnic Ministry of Health Surabaya, Indonesia. In 2018, Triwiyanto received the best Doctoral Student award from Gadjah Mada University. Additionally, he is Editor-in-chief in several peer review journals, Chairman and Technical Programme Committee in several International Conferences. His current research interests include microcontroller, electronics, biomedical signal processing, machine learning, rehabilitation engineering, and surface electromyography (sEMG)-based physical human robot interactions.

Sari Luthfiyah received the master degree from Airlangga University, Indonesia. She works at Department of Medical Electronics Technology, Poltekkes Kemenkes Surabaya, Indonesia. She is an associate professor at Poltekkes Kemenkes Surabaya from 2014 to present. Additionally, she is member of editorial board in several international journal. Her current research interests include biomedical signal processing, machine learning, rehabilitation engineering, and surface electromyography (sEMG)-based physical human robot interactions

Wahyu Caesarendra, PhD. was born in Jakarta, Indonesia on January 31st, 1982. He received Bachelor of Engineering degree from Diponegoro University, Indonesia in 2005. He worked in automotive and electrical company prior to join Diponegoro University as a Lecturer in 2007. He received New University for Regional Innovation (NURI) and Brain Korea 21 (BK21) scholarships for Master study in 2008 and obtained his Master of Engineering (M.Eng) degree from Pukyong National University, South Korea in 2010. In 2011, Wahyu Caesarendra was awarded of University Postgraduate Award (UPA) and International Postgraduate Tuition Award (IPTA) from University of Wollongong. He received Doctor of Philosophy (Ph.D) degree from University of Wollongong in 2015. He published more than 30 research articles in Journals and Conference proceedings. One of them was being the top 25 most cited paper in Mechanical Systems and Signal Processing Journal. He has developed skills and experience in scholarly research, teaching and learning activities. He is currently Postdoctoral Fellow in Rolls-Royce@NTU Corp Lab, School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore.

Abdussalam Ali Ahmed, PhD was born in Libya. From 2007 to present, he works as Assistant Professor at Mechanical and Industrial Engineering Department, Bani Waleed University, Bani Waleed, Libya. Currently, he is member of Renewable Energy Society of India (RESI): India.