Received 19 January 2023, accepted 12 February 2023, date of publication 22 February 2023, date of current version 1 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3247631

APPLIED RESEARCH

Reservoir Computing Model for Human Hand Locomotion Signal Classification

THONGKING WITCHUDA[1], ARDI WIRANATA[2], SHINGO MAEDA[3,4], (Member, IEEE), AND CHINTHAKA PREMACHANDRA[5], (Senior Member, IEEE)

[1]Department of Engineering and Science, Shibaura Institute of Technology, Tokyo 135-8548, Japan
[2]Department of Mechanical and Industrial Engineering, University of Gadjah Mada, Yogyakarta 55281, Indonesia
[3]Department of Mechanical Engineering, Tokyo Institute of Technology, Meguro, Tokyo 152-8550, Japan
[4]Living Systems Materialogy (LiSM) Research Group, International Research Frontiers Initiative (IRFI), Tokyo Institute of Technology, Midori, Yokohama 226-8501, Japan
[5]Department of Electronic Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan

Corresponding author: Chinthaka Premachandra (chintaka@sic.shibaura-it.ac.jp)

This work was supported by the Japan Society for the Promotion of Science under Grants-in-Aid for Scientific Research on Innovative Areas under Project 18H05473.

ABSTRACT Human-movement recognition is a novel challenge in soft robotics. In recent years, there have been several attempts to develop soft wearable devices for supporting human-robot interfaces. Many algorithms and programming languages are available to integrate a wearable device with a soft robot. One such promising algorithm is reservoir computing (RC), which includes of a group of recurrently and randomly connected nodes. The RC model can easily process multidimensional signal data and can handle nonlineardata and has been extensively used in robotic control. It has been reported that the RC algorithm can speed up network training and solve complex data sets. However, the main existing limitations in hand-locomotion classification are the considerable run-time and the delayed response. In this study, we figure out the best machine learning algorithms to handle three-dimensional hand-gesture data. We employ a two-part strategy: a loopback filter is included in the preprocessing of the initial dataset to support the 3-dimensional (3D) signs of each hand posture; subsequently, the training dataset is applied to the machine learning algorithm which includes an artificial neural network (ANN), convolutional neural network (CNN), long short-term memory(LSTM), and reservoir computing(RC). Each training network is optimized with various hyperparameters. Furthermore, we compare the performance of several machine-learning algorithms in classifying the three-dimensional hand-signal posture data. The results show that the classification of nonlinear hand-locomotion signals by RC requires a comparatively shorter training duration (12 minutes for training times), and that optimal accuracy 94.17, precision 94.10, and recall 93.99 is realized for time series data.

INDEX TERMS Human–machine interface, human hand-locomotion signal, reservoir computing, time series, multi dimension, and nonlinear data.

I. INTRODUCTION

The human-machine interface (HMI) is a novel breakthrough technology for improving and supporting human life and can be applied for device control to assist severely injured patients and for remote object manipulation [1], [2], [3], [4]. HMI units enable interaction between human gestures and a machine tracker via a user interface [5], [6]. The most expressive human gesture is hand locomotion [7], [8]. Initially, the classification of hand gestures in sign language from two-dimensional images was accomplished using a personal computer (PC) [7]. However, this classification process could not classify complex features nor reduce noise and blur motion. There were several attempts to develop software including artificial intelligence (AI) for solving this challenge and recognizing complex human signals [9], [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Guangcun Shan.

Using AI software, it is possible to assess multidimensional and complex locomotion signals and classify the features of human movements [11], [12], [13]. Thus, people can interact with media or robots through a HMI application. Human-gesture sign includes electromyography (EMG) [14], [15], [16], electroencephalography (EEG), radar [17], and strain signals [18], [19]. We focus on real-time and multidimensional data, and volunteer comfort. Radar is the best option for acquiring such signals. Due to hand locomotion, these signals have several patterns. AI algorithms such as image processing [20], [21], computer vision [22], principal component analysis (PCA) [8], support vector machine (SVM) [23], artificial neural networks (ANNs) [24], and convolutional neural network (CNNs) [25] can be used for the appropriate analysis and classification of these signals.

The main limitations in hand-locomotion classification are the considerable run-time and the delayed response. General machine learning (ML) algorithms do not support certain characteristic hand-locomotion data such as nonlinear and multidimensional time series data. Reservoir computing (RC), which is a type of recurrent neural network (RNN), is an excellent algorithm for learning a nonlinear dataset. A traditional RNN approached the input data and the output weighting that was the training vector, readjusting all input, learning perceptron, and the weighting output [26]. In contrast, in RC architecture, the input and perceptron weights are trained, and the training process trains the output weights. This algorithm has a simple training process and includes a series of connected nonlinear data nodes in a feedforward network and feedback connections. Moreover, RC has effective hyperparameters for handling complex signal data [26], [27], [28], [29]. Thus, some studies used RC algorithms to analyze bio-signal data for epileptic seizure detection and artificial finger control [30]. In weather forecasting, RC was used for predicting the chaotic nature of the weather during a period [31].

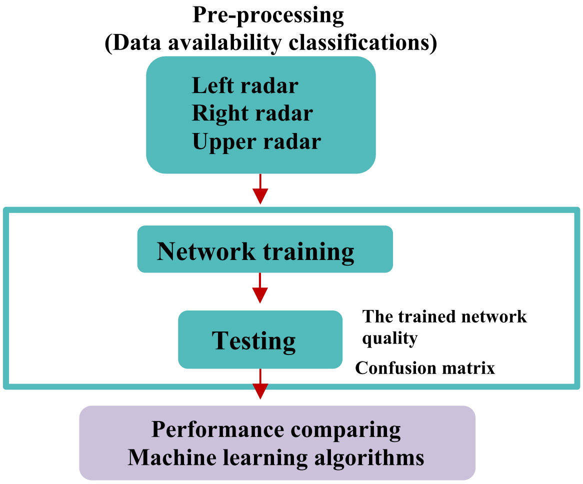


FIGURE 1. Conceptual diagram of machine learning implementation.

This study indicates how to handle the nonlinear dataset or dynamic data in an optimal way by loopback filter and presents the optimal training process by using RC algorithms in the shortest training times and provides the highest performance when compared with other algorithms. Then, we suggest an appropriate ML algorithm for handling nonlinear and time-series hand-locomotion signal datasets. In section II, we arrange the raw radar signal data, obtained using three-radar sensor equipment. In section III, we handle the negative values of the hand-locomotion signal data with the loopback filter equation. We represent the axis arrangement of each radar sensor. Further, we use ML algorithms for learning nonlinear and high-dimensional data. The implementation is presented in section IV. In section V, we compare the performance of several ML algorithms in classifying the three-dimensional hand-signal posture data and indicate the ML algorithm with the best performance. We discuss the findings in section VI and conclude the paper in section VII.

II. DATASET

Hand-locomotion signs are expressive data for human-machine interaction. General classification algorithms (e.g., ANN, CNN) involve several processes and the time required for training the machine learning model is considerable. Moreover, as previously mentioned, general ML algorithms do not support certain characteristic data such as nonlinear and multidimensional time series [17]. Therefore, we introduce a high-performance ML algorithm for the classification of hand-locomotion signals.

In this study, we adopt the hand-locomotion data set collected by Ahmed et al. [17]. This data set includes hand movement signals tracked by three radar sensors. The radars were positioned at the top, right, and left, as shown in Fig.1 and the data were collected from eight volunteers. Each hand gesture was recorded for 4.5 s (90 slow time frames) and the radar period was 1.2 m (189 fast time frames). The data comprise twelve different hand movements. The hand gesture includes left-right swipe, right-left swipe, top-down swipe, down-top swipe, oblique left-right to top-down swipe, oblique left-right to down-top swipe, oblique right-left to top-down swipe, oblique right-left to down-top swipe, clockwise rotation, opposite clockwise rotation, push button, and free movement as shown in Fig 2.

III. SENSORY DATASET

We extracted the data from the hand movements based on normalization in the comma-separated values (CSV) File. We used Python language and the Scikit library to extract the features of the radar dataset. This process is the first step for generating the ML model. Further, we preprocessed the data, generated the ML model, and evaluated the model as shown in Fig.1.

IV. PROPOSED DATA PREPROCESSING APPROACH

We first describe the preprocessing step for extracting the features of the hand-gesture signals. This data provides

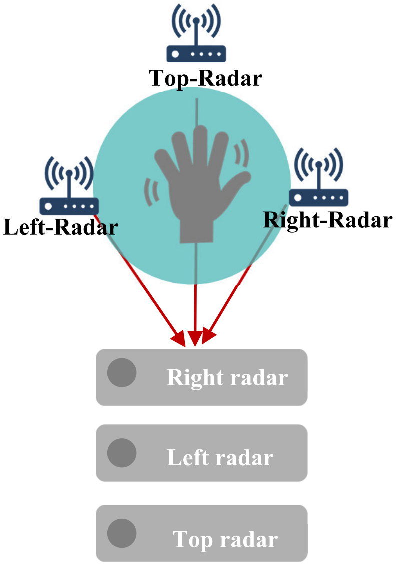


FIGURE 2. Schematic of the overall implementation of radar sensors for collecting hand-gesture signals.

12 features of the hand movements obtained using radar sensors at the top, right, and left positions. We examined the change in the negative data values over time and the implementation of the key policy interventions. We then used a filter for manipulating the negative values of the hand-gesture signal data. The loopback filter at a learning rate k is calculated as follows:

$$\boldsymbol{C}_n[k] = \alpha \boldsymbol{C}_n[k-1] + (1-\alpha)\boldsymbol{X}_n[k] \tag{1}$$

where $\boldsymbol{C}_n[k]$ denotes the extract clutter of the radar hand- movement signal at each hand posture state ($n = 1, 2, 3, \ldots, 12$). k denotes the learning rate in state which rate is 0.1 -1.0. Alpha (α) is the weighting factor. $\boldsymbol{X}_n$ is the hand posture type denoted by the radar hand-gesture signs.

Further, we arranged the hand-locomotion signal as a time series ($S_n r_n$). The preparation of ML is the first step for managing the dataset. The hand-gesture signals were manipulatedusing the NumPy library in Python. The input data was managed from a single axis to three dimensions (x, y, z) . The signal matrix represents the conversion dimensions. Each axis of the hand posture was multiplied as shown in equation 2:

$$\begin{bmatrix} L_1r_1 & L_1r_2 & \cdots & L_1r_{\mathrm{n}} \\ L_2r_1 & L_2r_2 & \cdots & L_2r_n \\ \vdots & \vdots & \ddots & \vdots \\ L_{\mathrm{n}}r_1 & L_{\mathrm{n}}r_2 & \cdots & L_{\mathrm{n}}r_n \end{bmatrix} \begin{bmatrix} R_1r_1 & R_1r_2 & \cdots & R_1r_n \\ R_2r_1 & R_2r_2 & \cdots & R_2r_n \\ \vdots & \vdots & \ddots & \vdots \\ R_nr_1 & R_nr_2 & \cdots & R_nr_n \end{bmatrix}$$
$$\times \begin{bmatrix} U_1r_1 & U_1r_2 & \cdots & U_1r_n \\ U_2r_1 & U_2r_2 & \cdots & U_2r_n \\ \vdots & \vdots & \ddots & \vdots \\ U_nr_1 & U_nr_2 & \cdots & U_nr_n \end{bmatrix}$$

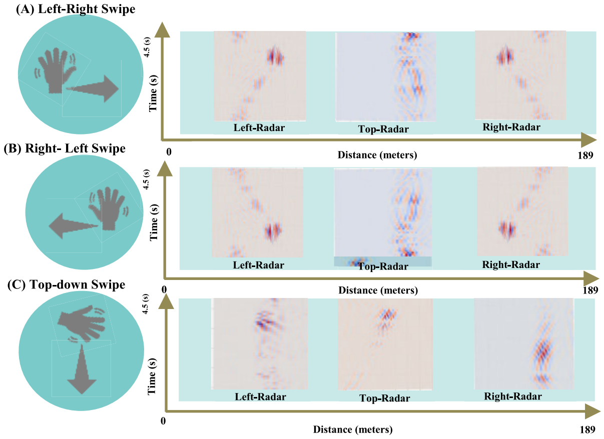


FIGURE 3. Features of the hand-movement signals from the top, right, and left radars: (A) Left-right swipe, (B) Right-left swipe, and (C) Top-down swipe.

$$= \begin{bmatrix} S_1r_1 & S_1r_2 & \cdots & S_1r_n \\ S_2r_1 & S_2r_2 & \cdots & S_2r_n \\ \vdots & \vdots & \ddots & \vdots \\ S_nr_1 & S_nr_2 & \cdots & S_nr_n \end{bmatrix}, \tag{2}$$

where $S_n r_n$ is the radar hand-movement signal arranged using equation 2. $L_n r_n$ represents the left data of the signal collected by the left radar. $R_n r_n$ represents the right data of the signal collected by the radar on the right. $U_n r_n$ represents the sign of the top position collected by the top radar. Each sign is 90 × 189 (fast time × slow time).

In our classification model, we emphasize the extent of association between the characteristic of the hand-gesture signal and the significant parameter of RC. Through preprocessing, the input data in the radar signal space and the time sequence is obtained as shown in Fig. 3.

We used the hand-locomotion signal dataset for training and evaluating the ML algorithms. The learning data comprised 2,400 samples per hand locomotion. Each hand posture was acquired using a three-radar equipment. Three dimensions were introduced, as shown in Fig. 3. The graphs indicate the features of the hand-movement signals. Further, we converted the dependent dimension per locomotion into three-dimensional data as shown in Fig. 4.

The hand-movement signal dataset included 9,600 data. Each hand posture comprised 800 data. We separated this dataset into training (7,680), testing (960), and validation datasets (960) as shown in Fig. 5.

V. MODEL IMPLEMENTATION

A machine learning (ML) model for the classification of human-hand locomotion signals was implemented. We applied ML to classify the hand-gesture radar signals as depicted in equation 2. We converted the radar signals into three dimensions, and input the converted data to a node of the ML architecture [10]. We then classified the hand-posture signals with three dimensions (top, right, and left radar signals). Python (with TensorFlow, Keras, NumPy,

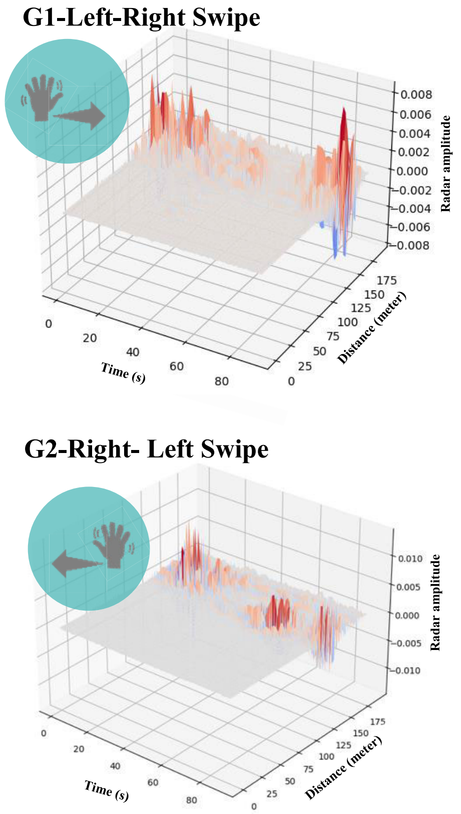


FIGURE 4. Hand-movement signal features (12) in 3-dimension. The 12 features comprise (G1) Left-right swipe and (G2) Right-left swip.

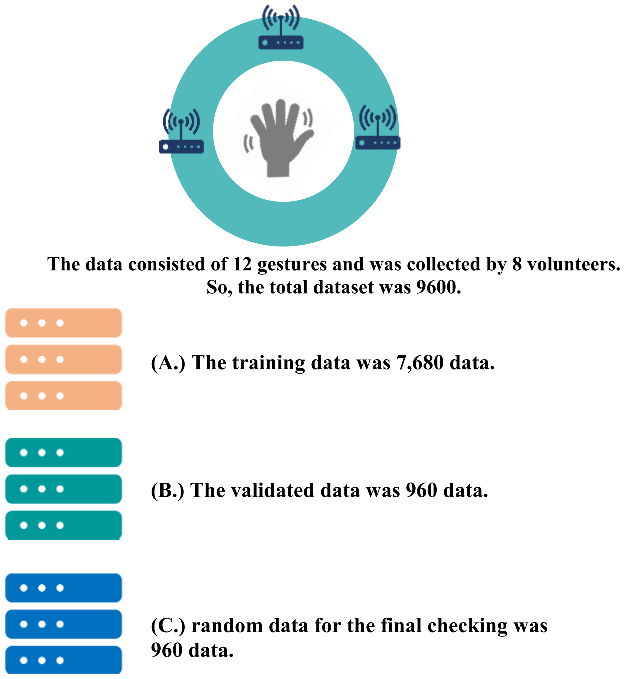

The data consisted of 12 gestures and was collected by 8 volunteers.
So, the total dataset was 9600.
(A.) The training data was 7,680 data.
(B.) The validated data was 960 data.
(C.) random data for the final checking was 960 data.

FIGURE 5. Splitting of the dataset for machine learning implementation: (A.) training, (B.) validation, and (C.) testing data.

Matplotlib, and Sklearn libraries) was used for developing the ML algorithm to classify the hand-locomotion signals.

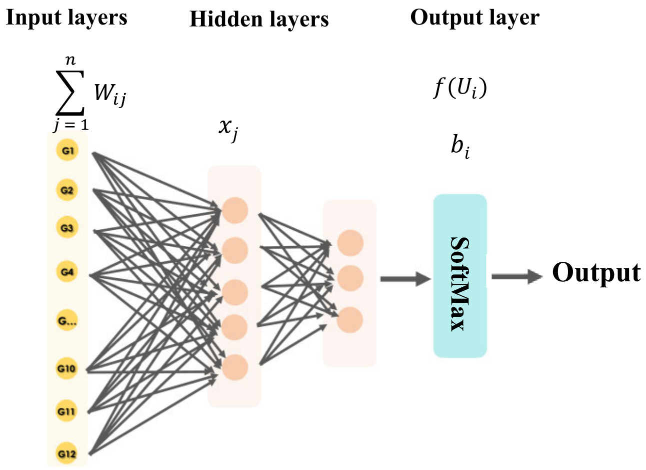


FIGURE 6. Artificial neural network architecture comprises an input layer, hidden layer, and output layer.

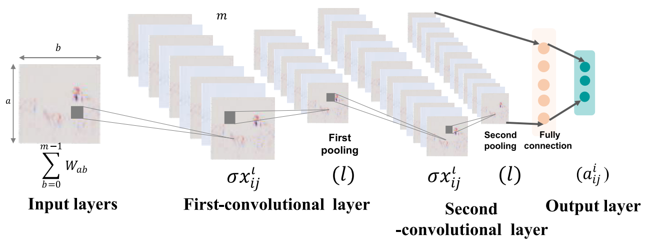


FIGURE 7. Convolutional neural network architecture consists of the input layer, number of convolutional layers, number of pooling layers, connection layer and output layer.

In this study, we apply and review different ML algorithms.

1)The ANN is a popular algorithm for classification. An ANN with three parts (input node, hidden layer node, and output node) is shown in Fig.6. The input node represents the signal node variables. x_i represents the 12 human gestures (i = 1, 2, 3, . . . , 12). In the input layer, W_{ij} is the weight of the hidden 12 human gestures and the neuron proceeding (j). The weight sum (W_i) of the neuron is summed as follows:

$$W_i = \sum\nolimits_{j=1}^{n} W_{ij}x_j. \tag{3}$$

The bias equation (b_i) generalizes the error of the model as follows:

$$U_i = W_i + b_i \tag{4}$$

The summation of the ANN algorithm (y_i), referred to as the activation function $f(U_i)$, is depicted in equation 5.

$$y_i = f\ (U_i) \tag{5}$$

The ANN network is studied in relation to the performance as shown in Fig.6 [32].

2)The CNN is a powerful neural network for ML. The learning ability of the CNN is due to its multiple feature extraction stages including object detection, video processing, natural language processing, speech recognition, image classification, and signal classification [25]. The CNN comprises an input layer, convolutional layers, a connection layer, and an output layer. When data is input to the input layer, it is worked upon by the feed-forward neural network. The convolutional layers learn and manipulate the input data with the dot product of the convolution kernel as shown in Fig.7. Suppose that the convolutional layer is preceded by an $N \times N$ square neuron layer and an $m \times m$ filter is used for manipulating the signal input. To compute the pre-nonlinearity input to a unit a^{i}_{ij} in the learning layer (W_{ab}), the contributions of the previous layers are summed up as shown in equation 6. Theneuron node represents the corresponding weight-sharing in the neural network: a to b.

$$a^{i}_{ij} = \sum_{a=0}^{m-1}\sum_{b=0}^{m-1} W_{ab} b^{l-1}_{(i+a)(j+b)} \tag{6}$$

Then, the convolutional layer applies its nonlinearity. The activation function is denoted by σx^{i}_{ij}, as follows:

$$y^{i}_{ij} = \sigma x^{i}_{ij} \tag{7}$$

The hyperparameters of the convolutional layer such as the kernel size, input shape time, max pooling, and its activation function are adjusted. The training process is followed by other processing, for e.g., in the pooling layers and connection layers as shown in Fig 7.

3) Long short-term memory (LSTM) networks involve interaction between the memory and gates. The gates are a combination of input, forget, and output gates. These gates facilitate multiplicative interaction. The architecture learns and predicts the sequence of the gate cycle. When input data at a sequential time step (X_t) flow into the LSTM algorithm, the data are learned through a mechanism such as a forget gate and input gate.The LSTM cell has three inputs (X_t, h_{t-1}, and C_{t-1}) and two outputs (h_t and C_t). C_{t-1} represents the cell state or memory and X_t indicate the current data point or input data. h_{t-1} is the hidden state of the previous time stamp and h_t is the hidden state of the current time stamp.

The first step is the keep process that decides whether the information from the previous timestamp should be kept or forgotten. The forget gates include the sigmoid activation (σ), which calculates the cell state. The equation for the forget gate is

$$f_t = \sigma\left(x_t U_f + (h_{t-1})W_f\right) \tag{8}$$

The input gate quantifies the importance of the new information carried by the input. The input gate equation is

$$i_t = \left(x_t U_i + (h_{t-1})W_i\right) \tag{9}$$

The equation of the output gate (C_t) is similar to those of the two previous gates:

$$C_t = \left(x_t U_o + (h_{t-1})W_o\right) \tag{10}$$

The last LSTM state was calculated the current hidden state. Here, C_t and *tanh* of the updated cell state are used to

$$h_t = (C_t x_i \tanh) \tag{11}$$

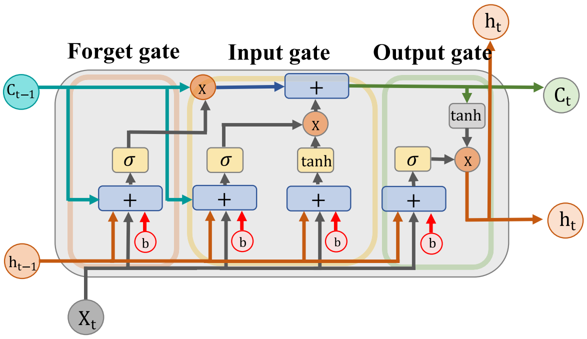


FIGURE 8. Long short-term memory architecture comprising an input gate, forget gate, and output gate.

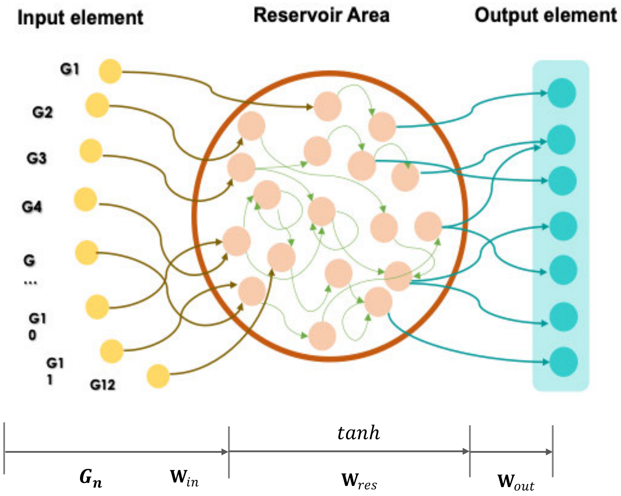


FIGURE 9. Reservoir computing architecture comprising an input element, reservoir area, and output element.

create C_t of the new candidate values as shown below:

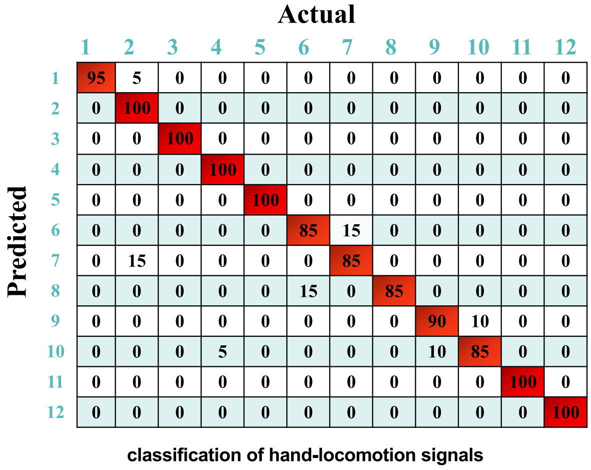
The results of the two layers undergo point-wise multiplication to produce the output (h_t) of the cell as shown in Fig. 8 [33].

4)Reservoir computing (RC) is an architecture for computing the weight of a neural network. G^{n}_{i} represents the 12 human gestures ($n = 1, 2, 3, \ldots, 12$). The signal input and input weight co-efficient ($\mathbf{W}_{in}$) are trained into higher-dimensional computation spaces or reservoir areas. $\mathbf{W}_{res}$ are fixed to the initialized values. Tanh function is a nonlinear function. In the readout training mechanism, the reservoir training weight is $\mathbf{W}_{in}$, the final output is $\mathbf{W}_{out}$. The outputs $y(t)$ are obtained by equations 12 and 13, and Fig. 9 [26], [27].

$$\mathbf{G}(t) = \tanh(\mathbf{W}_{res}\mathbf{x}^{n}_{i}(t-1) + \mathbf{W}_{in}G^{n}_{i}(t)) \tag{12}$$

$$\mathbf{y}(t) = \mathbf{W}_{out}\mathbf{x}(t) \tag{13}$$

where $\mathbf{W}_{out}$ is the readout weight matrix that is connections from the neurons in the reservoir layer to those in the output layer, typically in closed form through direct methods such as pseudo-inversion or ridge-regression.



	1	2	3	4	5	6	7	8	9	10	11	12
1	95	5	0	0	0	0	0	0	0	0	0	0
2	0	100	0	0	0	0	0	0	0	0	0	0
3	0	0	100	0	0	0	0	0	0	0	0	0
4	0	0	0	100	0	0	0	0	0	0	0	0
5	0	0	0	0	100	0	0	0	0	0	0	0
6	0	0	0	0	0	85	15	0	0	0	0	0
7	0	15	0	0	0	0	85	0	0	0	0	0
8	0	0	0	0	0	15	0	85	0	0	0	0
9	0	0	0	0	0	0	0	0	90	10	0	0
10	0	0	0	5	0	0	0	0	10	85	0	0
11	0	0	0	0	0	0	0	0	0	0	100	0
12	0	0	0	0	0	0	0	0	0	0	0	100

FIGURE 10. Reservoir computing confusion matrix for the classification of hand-locomotion signals.

TABLE 1. Performance of the machine learning models in classifying human hand-locomotion signals.

NO.	ALGORITHM	ACCURACY (%)	PRECISION (%)	RECALL (%)	SD
1.	RC [27]	94.17	94.10	93.99	0.27
2.	LSTM [34]	81.67	81.45	81.67	0.29
3.	ANN [10,24]	8.75	13.33	9.12	1.28
4.	CNN [12,14]	17.08	16.70	9.93	1.58

TABLE 2. Run- time performance in classifying human hand-locomotion signals.

NO.	ALGORITHMS	RUN-TIME (min)
1.	RC [27]	12.38
2.	LSTM [34]	27.62
3.	ANN [10,24]	35.35
4.	CNN [12,14]	30.10

The computational performance of the ML model is a measure of the hand-locomotion signal classification. In the implementation, we set size $N = 3500$ nodes. The ML model was trained in three dimensions. We systematically selected the hyperparameters of the model, for e. g., the reservoir node, spectral radius, and leak rate.

In this study, we simplified the evaluation of the timing package to present the training time of the ML models. We use the sklearn package. This package is called "time". The management considers a matrix vector X, the estimated vector Y of the datasets. For each model of our studies, time will output both the estimated time and its confidence interval. In ML, the confusion matrix is used for calculating the performance of the algorithm as shown in Fig.10. Each row of the confusion matrix represents the actual class, whereas each column represents the predicted class. The confusion matrix renders it easy to visualize whether the system correctly classifies multiple classes. We utilized the calculation matrix to calculate the accuracy, precision, and recall of the classification of hand-locomotion signals. The accuracy and precision are used for measuring the observational error. Accuracy indicates how close or far-off are a given set of measurements from their true values, whereas precision indicates how close or dispersed the measurements are with each other. The recall or sensitivity mathematically represents the presence or absence of the accuracy condition. The number of positive predictions is (P) and the number of negative predictions is (N) .

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

$$Precision = \frac{TP}{TP + TN} \tag{15}$$

$$Recall\ or\ Sensitivity = \frac{TP}{TP + FN} \tag{16}$$

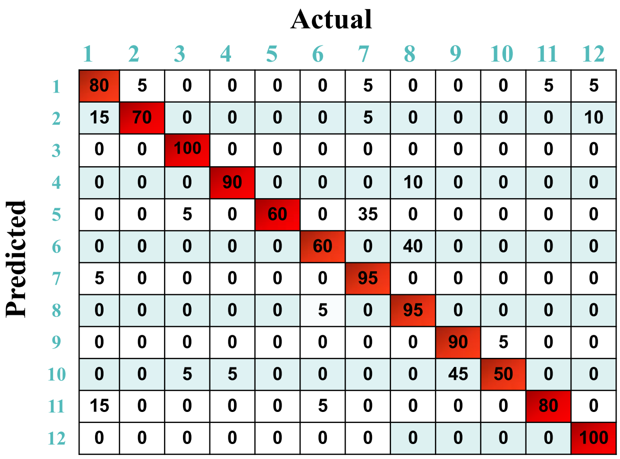
Here, *TP* is true positive, which indicates the presence condition of the model. *TN* is true negative, which indicates the absence condition of the model. *FP* is false positive, which indicates a wrong presence condition of the model. *FN* is false negative, which indicates a wrong absence condition.

VI. RESULTS

We proposed an approach involving the usage of ML algorithms for classifying the hand-movement signalsin a time series dataset and indicated the appropriate algorithms for evaluating the performance of the classification model. The classification algorithms included the ANN, CNN, LSTM, and RC. All the ML models were trained and tested on a PC with an Intel(R) Core (TM) i7-10700KF CPU @ 3.80 GHz and 16.0 GB RAM. The results of our optimized algorithms are presented below.

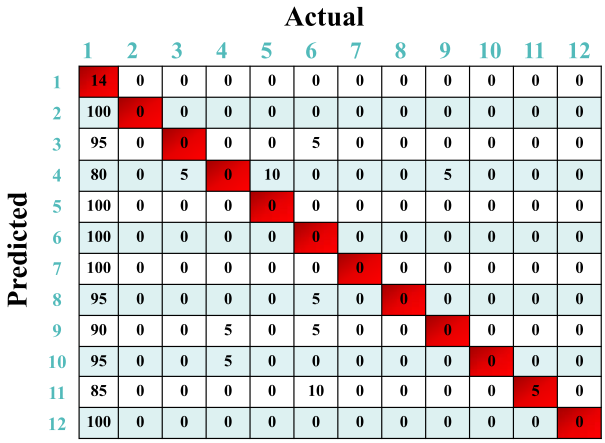
1) The performance of the product RC in hand- movement signal classification is shown in Fig.10. In this study, RC achieved the best performance in hand- locomotion signal classification. The optimizing hyperparameter was determined with the systematic parameters. The RC hyperparameters included the weight $(0.1 < \mathbf{W}_{in} < 1.00)$, spectral radius $(0.01 < \lambda < 0.95)$, and the number of neural nodes in a reservoir of 3500 nodes. The sequence of the feeding network was [90, 189]. In our experiments, optimal performance was realized with $\mathbf{W}_{in} = 0.55$, $\lambda = 0.55$, and tanh activation. The RC model was trained in 2000 epochs. When the sensitivity of neuron weight indicated that the neuron weight was less than 0.5, the accuracy of the performances was decreased. However, the accuracy decreased when the neuron weight was more than 0.5 as well [35]. The accuracy of RC in hand-locomotion classification was 94.17% and the error rate was 5.83%. The performances of the other models are also shown in tables 1 and 2.

2) The performance of the optimized product LSTM in hand- movement signal classification is shown in Fig.11. LSTM showed good performance in classifying hand-locomotion signals. The optimizing hyperparameter was


Actual
Predicted

	1	2	3	4	5	6	7	8	9	10	11	12
1	80	5	0	0	0	0	5	0	0	0	5	5
2	15	70	0	0	0	0	5	0	0	0	0	10
3	0	0	100	0	0	0	0	0	0	0	0	0
4	0	0	0	90	0	0	0	10	0	0	0	0
5	0	0	5	0	60	0	35	0	0	0	0	0
6	0	0	0	0	0	60	0	40	0	0	0	0
7	5	0	0	0	0	0	95	0	0	0	0	0
8	0	0	0	0	0	5	0	95	0	0	0	0
9	0	0	0	0	0	0	0	0	90	5	0	0
10	0	0	5	5	0	0	0	0	45	50	0	0
11	15	0	0	0	0	5	0	0	0	0	80	0
12	0	0	0	0	0	0	0	0	0	0	0	100

FIGURE 11. Long short-term memory confusion matrix confusion matrix for the classification of hand-locomotion signals.

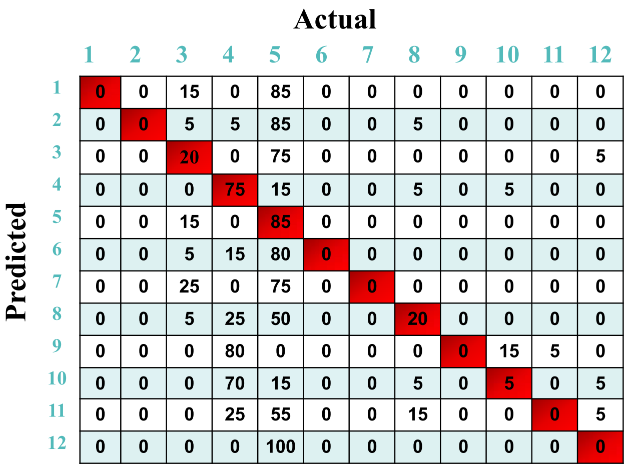

Actual
Predicted

	1	2	3	4	5	6	7	8	9	10	11	12
1	14	0	0	0	0	0	0	0	0	0	0	0
2	100	0	0	0	0	0	0	0	0	0	0	0
3	95	0	0	0	0	5	0	0	0	0	0	0
4	80	0	5	0	10	0	0	0	5	0	0	0
5	100	0	0	0	0	0	0	0	0	0	0	0
6	100	0	0	0	0	0	0	0	0	0	0	0
7	100	0	0	0	0	0	0	0	0	0	0	0
8	95	0	0	0	0	5	0	0	0	0	0	0
9	90	0	0	5	0	5	0	0	0	0	0	0
10	95	0	0	5	0	0	0	0	0	0	0	0
11	85	0	0	0	0	10	0	0	0	0	5	0
12	100	0	0	0	0	0	0	0	0	0	0	0

FIGURE 12. Artificial neural network confusion matrix for the classification of hand-locomotion signals.

determined using the systematic parameters. The LSTM hyperparameters included the sequential time-step, training weight of the forget gate ($0.01 < W_f < 1.00$), training weight of the input gate ($0.01 < W_o < 0.950$), and the number of neural nodes in reservoirs of 3500 nodes. In our experiments, the optimal performance was realized with $W_f = 0.5$, $W_o = 0.5$, and sigmoid activation. The classification accuracy was 81.67%, precision was 81.45%, and recall was 81.67%. The LSTM results show an increasing accuracy similar to that of RC. However, the algorithm does not exhibit high-performance under the same condition.

3) The product ANN shows capacity performance. The input coefficient is fixed at 12 hand locomotion signals. In Fig. 12, the X-axis shows the actual data for training the model, whereas the Y-axis shows the prediction data for the testing model. In general, the performance of the ANN in classifying hand-locomotion signals is the worst in this study. The accuracy, precision, and recall represented by equations (14), (15), and (16) are reduced because of the nonlinearity of the hand-locomotion signals in the dataset. ANN algorithms are similar with respect to the effect of the weight of the hidden 12 human gestures (W_{ij}) on the signal memory capacity. To understand the difference between the hyperparameters of the ANN algorithm and the nonlinear data, W_{ij} and the number of neural network nodes must be examined. We calculated the empirical total performance values of the ANN. The hyperparameters included the weight of the hidden 12 human gestures (W_{ij}) ($0.01 < (W_{ij}) < 1.00$) and the number of neural nodes in the ANN with 3500 nodes. The sequence of the feeding network is [90, 189]. In our experiments, optimal performance was generated by W_{ij} and the neural nodes. The ANN model was trained in 2000 epochs. The sensitivity indicated that it is not related to the training environment. The behavior of the ANN is analogous to the general quality of memory in neural network operation. This algorithm is unsuitable for a nonlinear dataset. The ANN classification accuracy was 8.75%, precision was 13.33%, and recall was 9.12%.


Actual
Predicted

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	15	0	85	0	0	0	0	0	0	0
2	0	0	5	5	85	0	0	5	0	0	0	0
3	0	0	20	0	75	0	0	0	0	0	0	5
4	0	0	0	75	15	0	0	5	0	5	0	0
5	0	0	15	0	85	0	0	0	0	0	0	0
6	0	0	5	15	80	0	0	0	0	0	0	0
7	0	0	25	0	75	0	0	0	0	0	0	0
8	0	0	5	25	50	0	0	20	0	0	0	0
9	0	0	0	80	0	0	0	0	0	15	5	0
10	0	0	0	70	15	0	0	5	0	5	0	5
11	0	0	0	25	55	0	0	15	0	0	0	5
12	0	0	0	0	100	0	0	0	0	0	0	0

FIGURE 13. Convolutional neural network confusion matrix for the classification of hand-locomotion signals.

4) The performance of the product CNN is shown in Fig. 13. The classification performance of the CNN was unsatisfactory. In our experiments, the pre-nonlinearity input (x^i_{ij}) was computed in the learning layer which W_{ab} occurs (the node represents the corresponding weight-sharing in the neural network) for the tasks in equations (6) and (7). The hyperparameters are the hidden weight (W_{ab}) ($0.01 < (W_{ab}) < 1.00$) and the number of neural nodes in a CNN of 3500 nodes. In the CNN experiments, optimal performance was realized with $W_{ab} = 0.5$ and 3500 neural nodes. The CNN model was trained in 2000 epochs. The sensitivity indicated that is not related to the training environment. The algorithm is unsuitable in the nonlinear regime. The CNN classification accuracy was 17.08%, precision was 16.70%, and recall was 9.93%.

Performance comparison between the ML algorithms for hand-movement classification indicated that the RC algorithm provided optimal performance as shown in tables 1 and 2. Sensitivity analysis of the classification models established that all the algorithms for hand-signal classification showed task-dependent sensitivity in different hyperparameter ranges. We used the optimal values of the hyperparameters to test and compare the classification performance. The correct values of accuracy, precision, and recall were determined through the experiments. Equations 14, 15, and 16 were used to calculate the accuracy, precision, and recall of the different algorithms, respectively.

VII. DISCUSSION

We studied ML architecture for understanding and classifying the relationship between the characteristics of the hand-locomotion signals and the machine leaning algorithms. Based on the findings, ML algorithms with the highest potential for analyzing, classifying, and predicting hand-gesture signals were suggested. Twelve hand-gesture signals were included for hand-locomotion signal classification. Four ML algorithms learned these hand gesture signs. The RC algorithm achieved the most optimal performance and the training process with time-series data was fast as well. The RC algorithm showed that hyperparameters such as the leak rate, spectral radius range, and number of neural nodes were important. A training weight ($\mathbf{W}_{in}$) of 0.5, spectral radius (λ) of 0.55, and tanh activation were assigned to the algorithm (1.00 being the highest weight and 1.00 being highest the spectral radius (λ)). We evaluated the performance of the algorithm by calculating the accuracy, precision, and recall (sensitivity) for each output and summing them. The hand-locomotion signal reflected the freedom behavior learning technique in each substrate. The 3500 RC neuron nodes showed a complete nonlinear capacity function for achieving higher capacity. When the sensitivity of neuron weight indicated that the neuron weight was less than 0.5, the accuracy, precision and recall of the performances were decreased. However, the performance decreased when the neuron weight was more than 0.5 as well [35]. The accuracy, precision, and recall of RC in hand-locomotion classification was 94.17, 94.10, and 93.99 percent. The error rate was 5.83%. The performances of the other models are also shown in tables 1 and 2. Furthermore, the training time was lesser compared to the other learning methods. As the RC algorithm consists of a bunch of recurrently connected units, the reservoir node is connected randomly. The RC algorithm transforms nonlinearly sequential inputs into a high-dimensional space. The features of the inputs can be efficiently read out by a simple learning algorithm [29], [30], [31]. With the other algorithms (ANN, CNN, and LSTM), the error increased with the training weight in the learning areas and standard deviation as shown in Fig 14 and Fig 15. The above-mentioned algorithms indicated instability in the network dynamics and the multi-dimension of the dataset [35]. Moreover, in the above-mentioned algorithms, the connecting node network is complicated, for e.g., the CNN algorithm includes a multi-convolutional layer and a polling step for learning. Furthermore, the CNN neuron nodes were connected through a sophisticated network [11], [12], [13]. The primary outcome of our study was the freedom learning areas capable of learning nonlinear data. The multiple weights of the neuron node parameter are critical success factors in ML models for the classification of hand-locomotion signals.

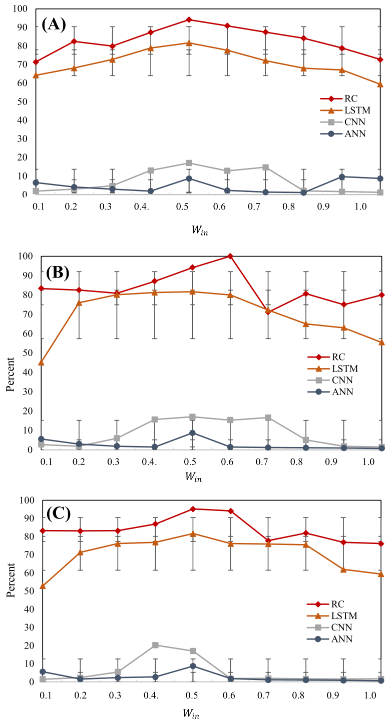


FIGURE 14. Human-locomotion classification performance: (A) Accuracy, (B) precision, and (C) recall.

Our findings advance the empirical and technical evidence from the current study in the classification of the hand-locomotion signals. We confirmed the result with standard deviations (SD). We represented that ANN and CNN was highly dispersion. The result indicated that the poor-quality mean results of the performance. The weight of neural networks provided the difference of performance results and negative feedback. However, RC and LSTM provided little dispersion and acceptable aggregation scores as shown in table 1. Therefore, it is the source of the negative feedback that arises. Thus, RC algorithms received the best feedback from the machine learning studies and almost no negative feedback. The adjustment of the neuron weights ($\boldsymbol{W}_{in}$) of proceeding has a major impact on the ML model and supports a nonlinear dataset [8], [17]. We determined the optimal values of the ML parameters to test and compare the performance in multistep classification. The normal neuron weight parameter was insufficient for training the dynamic and multidimensional dataset. However, the RC architecture has three training weights ($\boldsymbol{W}_{in}$, $\boldsymbol{W}_{out}$, $\boldsymbol{W}_{back}$) which facilitated learning and comprehension of the human-signal dataset behavior [35]. A reasonably good classification performance was achieved within a reasonable time. Thus, with multidimensional and nonlinear human-gesture signal data,

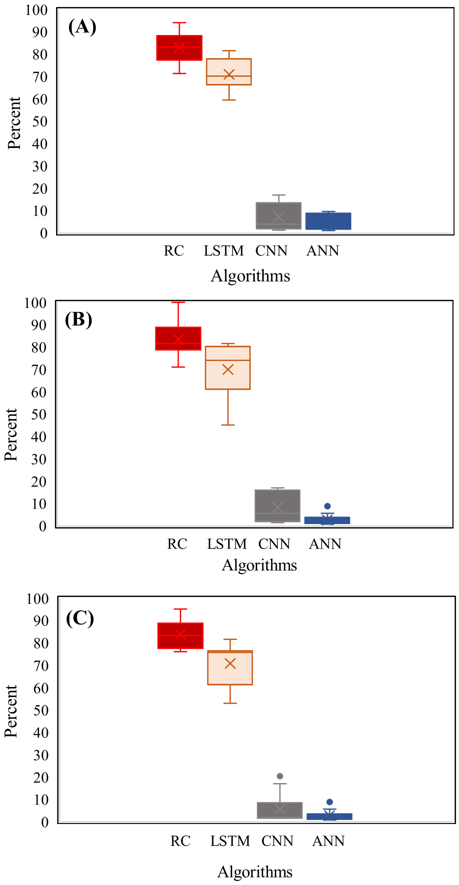


FIGURE 15. The boxplot of human-locomotion classification: (A) Accuracy, (B) precision, and (C) recall.

our study improves the machine learning performance, in the area of human- interaction.

Our models and research framework can be beneficial for policy making in ML algorithms and human-locomotion signals as follows. First, the ML models provide scenario classification to assess the nonlinear and multidimensional human-locomotion signal dataset. The input constraints in the models are the dimension of the data and raw data arrangement. As demonstrated in our study, the two dimensions of each radar sensor signal can be adjusted to generate three dimensions for learning in an actual scenario with human locomotion. In our study, we used the NumPy library for reshaping the data structure. The classification process reduced the negative value of the raw sensing data and provided appropriate pre-preparation for the ML model. Next, the number of neural nodes in the reservoir areas of the RC architecture could be adjusted. Our study indicated that the number of neural nodes was directly proportional to the number of inputs in the RC [26]. The other RC parameters were the number of epochs, leak rate, and spectral radius. The RC model has the least run time due to the simplicity of the interlayer architecture. Moreover, the dataset was not converted to another platform. RC architecture has the potential to learn from a numerical dataset. This study establishes that AI can be used in the HMI area and can access complex features.

However, our study has certain limitations that can be addressed through future research. The first is that the difficulty in identifying similar signal characteristics resulted in inaccuracies in classifying human-gesture signals. The ability of the RC network to classify human signals can be considered when designing controllers that can better conventional ML models. Furthermore, the current RC model can be improved by applying other hand-gesture signal datasets. The ML model should be trained with various resources (velocity and angle during gesture) for more detailed analysis.

VIII. CONCLUSION

We performed an empirical study of the relationship between the ML algorithm for classifying human hand-locomotion signals and the performance model. The employed dataset included nonlinear, multidimensional, and time series data. We showed that RC could directly classify nonlinear data in multidimension without transforming the dataset to another platform. Moreover, this algorithm required less time for classification compared to the other algorithms. Furthermore, the RC performed very well when paired with a dynamic data or nonlinear data, can be used in applications such as remote controlling machine. In the future, we aim to implement an RC algorithm for controlling electro strictive-type actuators such as soft robotic grippers [36] and linear-type dielectric elastomer actuators [37], and for recognizing the actuation pattern of dielectric elastomer actuators [38], [39], [40]. The findings of this study can contribute to the development of soft robotics, particularly, in human-robot interface and industrial applications.

SUPPLEMENTARY MATERIALS

The original contributions presented in the study are included in the Supplementary Materials; further inquiries can be directed to the corresponding author.

REFERENCES

[1] S. N. Young and J. M. Peschel, "Review of human–machine interfaces for small unmanned systems with robotic manipulators," *IEEE Trans. Human-Mach. Syst.*, vol. 50, no. 2, pp. 131–143, Apr. 2020.

[2] W. Dong, Y. Wang, Y. Zhou, Y. Bai, Z. Ju, J. Guo, G. Gu, K. Bai, G. Ouyang, S. Chen, Q. Zhang, and Y. Huang, "Soft human–machine interfaces: Design, sensing and stimulation," *Int. J. Intell. Robot. Appl.*, vol. 2, no. 3, pp. 313–338, Sep. 2018.

[3] C.-M. Chang, C.-S. Lin, W.-C. Chen, C.-T. Chen, and Y.-L. Hsu, "Development and application of a human–machine interface using head control and flexible numeric tables for the severely disabled," *Appl. Sci.*, vol. 10, no. 19, p. 7005, Oct. 2020.

[4] F. Stroppa, M. Luo, K. Yoshida, M. M. Coad, L. H. Blumenschein, and A. M. Okamura, "Human interface for teleoperated object manipulation with a soft growing robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, May 2020, pp. 726–732.

[5] M. Neghină, P. G. Larsen, and K. Pierce, *Multi-Paradigm Modelling Approaches for Cyber-Physical Systems*, V. Amaral, Ed. New York, NY, USA: Academic, 2021.

[6] Q. Ke, J. Liu, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Computer vision for human–machine interaction," in *Computer Vision for Assistive Healthcare: Computer Vision and Pattern Recognition*. New York, NY, USA: Academic, 2018, pp. 127–145.

[7] P. K. Sharma and S. R. Sharma, "Evolution of hand gesture recognition: A review," *Int. J. Eng. Comput. Sci.*, vol. 4, no. 4, pp. 9962–9965, 2015.

[8] Y. Liu, L. Jiang, H. Liu, and D. Ming, "A systematic analysis of hand movement functionality: Qualitative classification and quantitative investigation of hand grasp behavior," *Frontiers Neurorobot.*, vol. 15, Jun. 2021, Art. no. 658075.

[9] T. Yoshikawa, V. Losing, and E. Demircan, "Machine learning for human movement understanding," *Adv. Robot.*, vol. 34, no. 13, pp. 828–844, Jul. 2020.

[10] B. Mahesh, "Machine learning algorithms—A review," *Int. J. Sci. Res.*, vol. 9, no. 1, pp. 381–386, 2020.

[11] S. M. Mathews, "Explainable artificial intelligence applications in NLP, biomedical, and malware classification: A literature review," in *Proc. Comput. Conf. Intell. Comput.* Springer. 2019, pp. 1269–1292.

[12] W. Thongking, B. Sindhupakorn, P. Mitsomwang, and J. Tathanuch, "Analysis and classification of abnormal vertebral column by convolutional neural network algorithm," *Suranaree J. Soc. Sci.*, vol. 16, no. 1, pp. 1–13, 2022.

[13] B. Su and E. M. Gutierrez-Farewik, "Gait trajectory and gait phase prediction based on an LSTM network," *Sensors*, vol. 20, no. 24, p. 7127, Dec. 2020.

[14] V. B. Srinivasan, M. Islam, W. Zhang, and H. Ren, "Finger movement classification from myoelectric signals using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Kuala Lumpur, Malaysia, Dec. 2018, pp. 1070–1075.

[15] M. Yu, G. Li, D. Jiang, G. Jiang, B. Tao, and D. Chen, "Hand medical monitoring system based on machine learning and optimal EMG feature set," *Pers. Ubiquitous Comput.*, pp. 1–17, Aug. 2019. [Online]. Available: https://link.springer.com/article/10.1007/s00779-019-01285-2#citeas

[16] D. Ramírez-Martínez, M. Alfaro-Ponce, O. Pogrebnyak, M. Aldape-Pérez, and A.-J. Argüelles-Cruz, "Hand movement classification using Burg reflection coefficients," *Sensors*, vol. 19, no. 3, p. 475, Jan. 2019.

[17] S. Ahmed, D. Wang, J. Park, and S. H. Cho, "UWB-gestures, a public dataset of dynamic hand gestures acquired using impulse radar sensors," *Sci. Data*, vol. 8, no. 1, p. 102, Apr. 2021.

[18] A. Wiranata, Y. Ohsugi, A. Minaminosono, Z. Mao, H. Kurata, N. Hosoya, and S. Maeda, "A DIY fabrication approach of stretchable sensors using carbon nano tube powder for wearable device," *Frontiers Robot. AI*, vol. 8, Nov. 2021, Art. no. 773056.

[19] R. Chereshnev and A. Kertesz-Farkas, "HuGaDB: Human gait database for activity recognition from wearable inertial sensor networks," in *Computers and Society*. HSE Publications, 2017.

[20] A. Sharmaa, A. Mittal, S. Singha, and V. Awatramani, "Hand gesture recognition using image processing and feature extraction techniques," in *Proc. Int. Conf. Smart Sustain. Intell. Comput. Appl. (ICITETM)*. 2020, pp. 181–190.

[21] A. Korik, R. Sosnik, N. Siddique, and D. Coyle, "Imagined 3D hand movement trajectory decoding from sensorimotor EEG rhythms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 4591–4596.

[22] A. Chaudhary, J. L. Raheja, K. Das, and S. Raheja, "Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey," *Int. J. Comput. Sci. Eng. Surv.*, vol. 2, no. 1, pp. 122–133, Feb. 2011.

[23] K. Mebarkia and A. Reffad, "Multi optimized SVM classifiers for motor imagery left and right hand movement identification," *Australas. Phys. Eng. Sci. Med.*, vol. 42, no. 4, pp. 949–958, Dec. 2019.

[24] P. Viana, V. Fujii, L. Lima, G. Ouriques, G. Oliveira, R. Varoto, and A. Cliquet Jr., "An artificial neural network for hand movement classification using surface electromyography," in *Proc. 12th Int. Joint Conf. Biomed. Eng. Syst. Technol.*, 2019, pp. 185–192.

[25] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.

[26] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Netw.*, vol. 115, pp. 100–123, Jul. 2019.

[27] K. Nakajima and I. Fischer, *Reservoir Computing: Theory, Physical Implementations, and Applications*. Singapore: Springer, 2021.

[28] A. Polydoros, L. Nalpantidis, and V. Krüger, "Advantages and limitations of reservoir computing on model learning for robot control," in *Proc. IROS Workshop Mach. Learn. Planning Control Robot Motion*, Hamburg, Germany, 2015.

[29] A. A. Ferreira, T. B. Ludermir, and R. R. B. de Aquino, "An approach to reservoir computing design and training," *Expert Syst. Appl.*, vol. 40, no. 10, pp. 4172–4182, Aug. 2013.

[30] T. L. Carroll, "Optimizing reservoir computers for signal classification," *Frontiers Physiol.*, vol. 12, Jun. 2021, Art. no. 685121.

[31] D. Kudithipudi, Q. Saleh, C. Merkel, J. Thesing, and B. Wysocki, "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing," *Frontiers Neurosci.*, vol. 9, p. 502, Feb. 2016.

[32] Y. D. Mammedov, E. U. Olugu, and G. A. Farah, "Weather forecasting based on data-driven and physics-informed reservoir computing models," *Environ. Sci. Pollution Res.*, vol. 29, pp. 24131–24144, Apr. 2022.

[33] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "An efficient approach for neural network architecture," in *Proc. 25th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2018, pp. 745–748.

[34] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.

[35] A. Goudarzi, A. Shabani, and D. Stefanovic, "Product reservoir computing: Time-series computation with multiplicative neurons," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Killarney, Ireland, Jul. 2015, pp. 1–8.

[36] W. Thongking, A. Wiranata, A. Minaminosono, Z. Mao, and S. Maeda, "Soft robotic gripper based on multi-layers of dielectric elastomer actuators," *J. Robot. Mechatronics*, vol. 33, no. 4, pp. 968–974, Aug. 2021.

[37] A. Wiranata and S. Maeda, "A deformable linear dielectric elastomer actuator," *Key Eng. Mater.*, vol. 884, pp. 430–436, May 2021.

[38] A. Wiranata, A. M. A. Haidar, T. Murakami, A. Minaminosono, Z. Mao, and S. Maeda, "Dynamic characteristics of a dielectric elastomer actuator fabricated using a stretchable CNT powder electrode," in *Proc. 32nd Int. Symp. Micro-NanoMechatronics Hum. Sci.*, 2021, pp. 1–6.

[39] A. Minaminosono, H. Shigemune, Y. Okuno, T. Katsumata, N. Hosoya, and S. Maeda, "A deformable motor driven by dielectric elastomer actuators and flexible mechanisms," *Frontiers Robot. AI*, vol. 6, p. 1, Feb. 2019.

[40] X. Ji, X. Liu, V. Cacucciolo, M. Imboden, Y. Civet, A. E. Haitami, S. Cantin, Y. Perriard, and H. Shea, "An autonomous untethered fast soft robotic insect driven by low-voltage dielectric elastomer actuators," *Sci. Robot.*, vol. 4, no. 37, Dec. 2019, Art. no. eaaz6451.

THONGKING WITCHUDA received the B.Sc. and M.Eng. degrees from the Suranaree University of Technology, Thailand, in 2017 and 2021, respectively. She is currently pursuing the Ph.D. degree with the Shibaura Institute of Technology, Tokyo, Japan. Her research interests include depth image processing, machine learning, soft robotics, conductive fiber material, and soft material.

ARDI WIRANATA received the S.T. (Bachelor of Engineering) and M.Eng. degrees from the Department of Mechanical and Industrial Engineering, University of Gadjah Mada, in 2015 and 2016, respectively, and the Ph.D. degree from the Shibaura Institute of Technology, Toyosu, Japan, in 2022. He is currently a Lecturer with the Department of Mechanical and Industrial Engineering, University of Gadjah Mada. During his Ph.D., he received the Arimoto Shiro Memorial Prize. His research interests include soft robotics (including stretchable sensor and actuators), the IoT, DIY laboratory equipment, and wearable device. In addition, he is in charge of chanical tensile test equipment for stretchable conductive materials with the Engineering Simulation (including finite element analysis) Division, PSE-UGM.

SHINGO MAEDA (Member, IEEE) received the Ph.D. degree from Waseda University, in 2008.

From 2008 to 2009, he was an Assistant with Waseda University, where he was an Assistant Professor, from 2009 to 2011. From 2011 to 2014, he was an Assistant Professor with the Shibaura Institute of Technology, where he was an Associate Professor, from 2014 to 2019. From 2015 to 2016, he was a Visiting Professor with the Bio Robotics Institute, Scuola Superiore Sant'Anna, Italy. From 2021 to 2022, he was a Professor with the Shibaura Institute of Technology. Since 2022, he has been a Professor with the Tokyo Institute of Technology. His research interests include soft robotics and soft materials. He is a member of the Robotics Society of Japan (RSJ) and the Japan Society of Mechanical Engineers (JSME).

CHINTHAKA PREMACHANDRA (Senior Member, IEEE) was born in Sri Lanka. He received the B.Sc. and M.Sc. degrees from Mie University, Tsu, Japan, in 2006 and 2008, respectively, and the Ph.D. degree from Nagoya University, Nagoya, Japan, in 2011. From 2012 to 2015, he was an Assistant Professor with the Department of Electrical Engineering, Faculty of Engineering, Tokyo University of Science, Tokyo, Japan. From 2016 to 2017, he was an Assistant Professor with the Department of Electronic Engineering, School of Engineering, Shibaura Institute of Technology, Tokyo, where he was an Associate Professor, from 2018 to 2022. In 2022, he was promoted to a Professor with the Department of Electronic Engineering, Graduate School of Engineering, Shibaura Institute of Technology, where he is currently the Manager of the Image Processing and Robotic Laboratory. His research interests include AI, UAV, image processing, audio processing, intelligent transport systems (ITS), and mobile robotics.

He is a member of IEICE, Japan; SICE, Japan; RSJ, Japan; and SOFT, Japan. He received the IEEE Sensors Letters Best Paper Award from the IEEE Sensors Council in 2022 and the IEEE Japan Medal from the IEEE Tokyo Section in 2022. He also received the FIT Best Paper Award and the FIT Young Researchers Award from IEICE and IPSJ, Japan, in 2009 and 2010, respectively. He has served as a steering committee member and an editor for many international conferences and journals. He is the Founding Chair of the International Conference on Image Processing and Robotics (ICIPRoB), which is technically co-sponsored by IEEE. He is currently serving as an Associate Editor for IEEE Robotics and Automation Letters (R-AL) and *IEICE Transactions on Information and Systems*.