



Analysis and Evaluation of Feature Selection and Feature Extraction Methods

Rubén E. Nogales^{1,2} · Marco E. Benalcázar¹

Received: 19 April 2023 / Accepted: 14 August 2023
© The Author(s) 2023

Abstract

Hand gestures are widely used in human-to-human and human-to-machine communication. Therefore, hand gesture recognition is a topic of great interest. Hand gesture recognition is closely related to pattern recognition, where overfitting can occur when there are many predictors relative to the size of the training set. Therefore, it is necessary to reduce the dimensionality of the feature vectors through feature selection techniques. In addition, the need for portability in hand gesture recognition systems limits the use of deep learning algorithms. In this sense, a study of feature selection and extraction methods is proposed for the use of traditional machine learning algorithms. The feature selection methods analyzed are: maximum relevance and minimum redundancy (MRMR), Sequential, neighbor component analysis without parameters (NCAsp), neighbor component analysis with parameters (NCAp), Relief-F, and decision tree (DT). We also analyze the behavior of feature selection methods using classification and recognition accuracy and processing time. Feature selection methods were fed through seventeen feature extraction functions, which return a score proportional to its importance. The functions are then ranked according to their scores and fed to machine learning algorithms such as Artificial Neural Networks (ANN), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Decision Tree (DT). This work demonstrates that all feature selection methods evaluated on ANN provide better accuracy. In addition, the combination and number of feature extraction functions influence the accuracy and processing time.

Keywords Hand gesture recognition · Feature selection · Leap motion controller · Feature extraction · Machine learning

1 Introduction

Physical movement can express emotions, attitudes, and intentions non-verbally and is known as gestures. In addition, hand gestures are other forms of physical movements used to express feelings. Hand gesture recognition (HGR) is the process of detecting and interpreting hand movements and gestures made by a user. This technology is used in various fields, such as computer vision, human–computer interaction, and gaming. The recognition is typically done through image processing and machine learning algorithms,

which analyze the camera's visual information to identify and classify the gestures. The process of hand gesture recognition involves several steps, including image acquisition, pre-processing, feature extraction, and classification. In [1], the authors present a gesture recognition system based on extracting frames from the 3D positions and velocities of the fingers. These data feed a Long Short-Term Memory (LSTM) and a Convolutional Neural Network (CNN), the authors store data from six gestures.

In [2] presents an architecture they call KTSL, based on the combination of hand position, direction, and shape to represent the meaning of sign language. The Kinect device captures the data. This data feeds Hidden Markov Model classification algorithms and Support Vector Machines. In [3] presents a static and dynamic gesture recognition system based on Dynamic Time Warping. To define the motion trajectory, they also present a composition model called strokelets. In [4], they propose automatic feature extraction. In addition, they propose four manual extraction techniques such as angular point number, angular point series number

✉ Rubén E. Nogales
ruben.nogales@epn.edu.ec

¹ Artificial Intelligence and Computer Vision Research Lab, Departamento de Informática y Ciencias de la Computación (DICC), Escuela Politécnica Nacional, Av. Ladrón de Guevara E11-253, 170525 Quito, Ecuador

² Facultad de Ingeniería en Sistemas Electrónica e Industrial, Universidad Técnica de Ambato, Ambato, Ecuador

percentage, subarea percentage, and aspect ratio. These features are fed to the Hidden Markov Model.

In this context, HGR is not a trivial problem because it is viewed as a pattern recognition problem [5, 6]. Also, the HGR problem is challenging to solve using mathematical or statistical models. Because to use mathematical models is necessary to know the complete problem compartment [7]. In contrast, statistical models need to know all variables and their compartment. For this reason, it is feasible to try to solve the HGR problem using deep learning methods or machine learning algorithms.

In this context, deep learning methods can address the HGR problem because these methods extract features automatically. However, deep learning methods have become complex to use because HGR systems need portability. In addition, deep learning methods require a large computational load and programming of complex models. The problem of the computational load has been solved by using GPUs. And the portability problem is solved using portable GPUs. However, portable GPUs require high power consumption, which is against portability.

In this sense, we need to return to traditional machine learning algorithms such as artificial neural network (ANN), support vector machine (SVM), K-nearest neighbors (KNN), decision trees (DT), among others. Machine learning algorithms typically operate in multidimensional environments, where data is represented as high-dimensional feature vectors. These feature vectors capture various data characteristics and represent the data in a format that the machine learning algorithm can analyze. By working in a multidimensional environment, machine learning algorithms can capture complex relationships and patterns within the data that can be used for tasks such as classification, regression, clustering, and dimensionality reduction. The machine learning models use feature selection and feature extraction as dimensionality reduction technics. Feature extraction is a process that transforms the original dataset into a reduced number of features. In contrast, feature selection finds the element that gives us the most information about the problem. According to [9], the performance of a machine learning model is directly related to the number of input variables. According to [10] the excess of variables can reduce the model performance because there may be a high correlation between variables. In the same sense, for the best description of the problem, it is necessary to identify the variables with the most significant description of the problem. The definition of the variables with the highest descriptive load for the problem is not a trivial task. This is the reason why a lot of feature selection methods have been in development. These methods use some selection criteria, such as redundancy or data relevance, among others.

Feature selection methods are grouped into the filter, wrapper, and embedded methods. Filter methods present

a score based on correlation, mutual information, or the classifier's performance on a single variable. This method has no relation to the learning phase. Wrapper methods present the relevance of a subset of features that best predict how a machine learning model performs. Finally, embedded methods incorporate the selection of features into the training phase in algorithms such as DT [11].

In [13–15], the authors select the features retrieved by the leap motion controller. They present them as feature selection. These features are the position and the orientation of the fingers and the hands. In addition, the authors use functions such as the mean, the standard deviation, the correlation, the Shannon entropy, the kurtosis and the skewness as feature extraction. On the other hand, [16] presents a feature extraction of fingertip angles, fingertip distance, fingertip height, fingertip position, and over these features use f -value, Sequential feature selection, random forest feature for its study. On the basis of this study, [17] mentions that the reduction of the number of features to a reasonable number is necessary, then they sort the more significant features with the f -score algorithm. The f -score calculates the ratio of the variance between classes and the variance within classes. Also, [18] uses the Gaussian Mixture Model to select features. However, they mention structured sparsity inducing and genetic algorithms as feature selection algorithms for images. Feature selection has been widely used in the medical field, especially in the field of genetics [19, 20]. Also, [21] uses some feature selection algorithms to define the best features for the description of Parkinson's disease (PCA, SVM, consistency, J48, filtered subset evaluation, information gain, gain ratio, chi-square) of Weka.

In this context, and within the framework of the filtering methods of the feature selection, we analyze the method of the maximum relevance minimum redundancy (MRMR) and the sequential. In the same sense, for wrapper methods, we analyzed neighbors component analysis with lambda parameters (NCAp) and neighbors component analysis without parameters (NCAsp). Finally, for embedded methods, we analyze the Relief-F and Decision Tree (DT) methods for hand gesture classification and recognition. These methods are applied to a data set consisting of M observations and a set of 17 features. The features are obtained by applying 17 feature extraction functions to a raw data set retrieved from the Leap motion controller. Also, we use machine learning algorithms such as ANN, SVM, K-NN, and DT for validating the performance of feature selection methods. Each algorithm defines the accuracy of classification and recognition, the time processing of classification tests, and the accuracy difference between training and testing.

This analysis proposal is presented because, in the field of hand gesture recognition using signals from the leap motion controller and machine learning algorithms, no guidance

could be found in the scientific literature to decide on a priori feature selection methods to achieve high accuracy. In addition, many proposed methods are just a starting point for other studies.

In addition, the importance of the present work lies in the attempt to achieve a high recognition accuracy because, in practical terms, the output of recognition systems is used as input for other systems, such as home automation, human–computer interaction, in the medical field, among others. It can also be used as a classification system in rehabilitation. The classification system would label a healthy hand or a hand with problems. In the same context, these systems could run on embedded hardware with low computational power. However, reducing dimensionality sacrifices accuracy but gains computational power.

2 Objectives

- Define the best set of feature extraction functions to achieve maximum accuracy for hand gesture classification and recognition using feature selection methods such as MRMR, Sequential, NCAp, NCAsp, Relief-F, and DT.
- To analyze the behavior of feature selection methods using classification, recognition accuracy, and processing time with different machine learning algorithms.

The rest of the paper is organized as follows: Sect. 3 presents the overview of work, the dataset building, features extractions and feature selection functions, and machine learning algorithms. The experimentation and result section presents the combination of feature extraction functions according to the ranking of feature selection methods for each machine learning algorithm and shows comparative tables of experimentation results. Finally, the paper presents conclusions and discussion.

3 Methodology

In this section, we describe the general overview of the work, building the dataset, feature extraction functions, feature selection functions, and machine learning algorithms.

3.1 General Work Overview

This work uses the spatial positions and directions retrieved by the Leap Motion Controller (LMC). The LMC represents the position of the fingertips at time t using the matrix.

$$\mathbf{P}_t = [p_{(1,t)}^{(x)}, p_{(1,t)}^{(y)}, p_{(1,t)}^{(z)}, \dots, p_{(5,t)}^{(x)}, p_{(5,t)}^{(y)}, p_{(5,t)}^{(z)}]_t^{(leap)},$$

being $[p_{(i,t)}^{(x)}, p_{(i,t)}^{(y)}, p_{(i,t)}^{(z)}]$ the vector with the spatial positions of the i th finger with respect to the sensor coordinate axes.

The directions of the fingertips at time t are represented using the matrix:

$$\mathbf{D}_t = [d_{(1,t)}^{(x)}, d_{(1,t)}^{(y)}, d_{(1,t)}^{(z)}, \dots, d_{(5,t)}^{(x)}, d_{(5,t)}^{(y)}, d_{(5,t)}^{(z)}]_t^{(leap)},$$

being $[d_{(i,t)}^{(x)}, d_{(i,t)}^{(y)}, d_{(i,t)}^{(z)}]$ the vector with the directions of the i th finger with respect to the sensor coordinate axes.

In this paper, we use six feature selection methods. The methods used are MRMR, Sequential, NCAp, NCAsp, Relief-F, and DT. In the present paper, the movement of the hand is represented by the sum of the values of the spatial positions. In this sense, the data are structured as a matrix of $M \times N$, where M is the total observations and N is the features that describe the movement.

Next, we define 17 feature extraction functions. Initially, the functions have a random order. These statistical functions describe the central tendency and scatter measures, among others. Each feature extraction function is applied to the matrix data, and the matrix $M \times 17$ is obtained. In this sense, each data obtained by the feature extraction function represents a predictor.

Next, the feature selection methods such as MRMR, Sequential, NCAp, NCAsp, Relief-F, and DT are applied over the matrix $M \times 17$, where each column represents a feature extraction function. Each method is applied over the matrix $M \times 17$, as each column represents a feature extraction function. This process returns a matrix where the number represents the function used to extract the feature and its score. We sorted the functions according to their scores. We fed the machine learning algorithm. The way to feed machine learning algorithms is to add one feature extraction function at a time. In the first step, we use the feature extractor function with the highest score. Then we add the first and second function with the better score, and so on. Each feature is added according to the score report until the algorithms are fed with the combination of all 17 feature extractor functions. The machine learning algorithms used are ANN, SVM, KNN, and DT. All processes are shown in Fig. 1.

3.2 Data Set Building

The dataset was built using 56 people voluntaries of Universidad Técnica de Ambato between students and teachers, women, and men, aged 18–46 years old. None of the volunteers had injuries to the right upper extremity. In addition, information such as ethnicity, occupation, and e-mail are requested for each participant. The number of lumens perceived in the environment in which the data is collected is also entered. Also, for each proposed gesture, the position, direction, and velocity in X , Y , and Z of the palm and each of the fingers are recorded. The device used to acquire data was

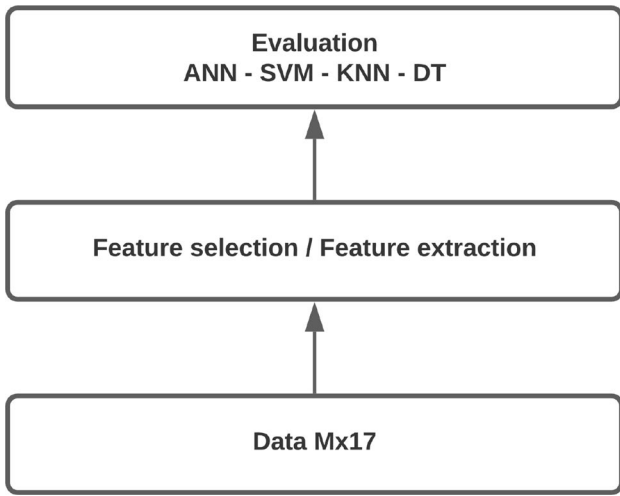


Fig. 1 General evaluation scheme for feature selection and feature extraction functions

LMC. We use this device because it is small and cheaper. It is specialized to track the hand. The LMC has three LED sensors and two depth cameras. Also, this device retrieves the spatial positions, directions, and velocity of hands and fingers according to the coordinate axis, which center is the center of the device [23]. Figure 2 shows the data acquisition process using the Leap Motion Controller.

Besides, each user develops five gestures. These gestures are Open Hand, Fist, Wave In, Wave Out, and Pinch, as shown in Fig. 3. The user can perform the hand movement representing the gesture at any time during the 5 s [24].

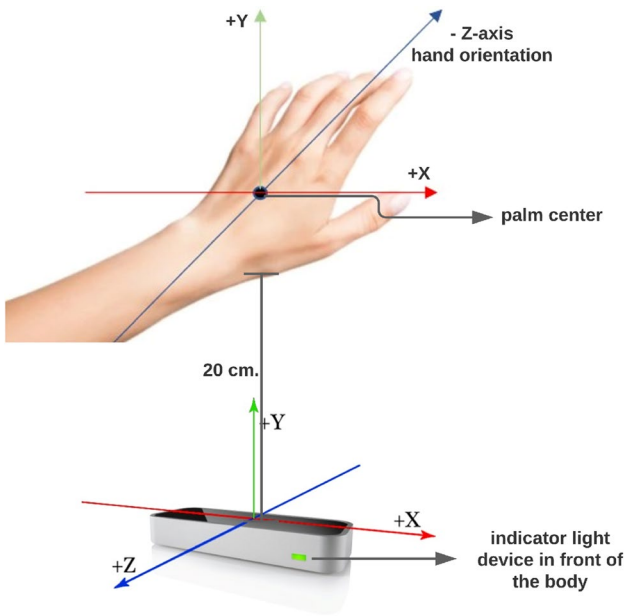


Fig. 2 Data acquisition using Leap Motion Controller



Fig. 3 Types of hand gestures

Each user repeats 30 times each gesture. In this sense, the dataset comprises 1680 observations of each gesture, 8400 observations total. The LMC has a sampling frequency of 200 Hz/s. But as our dataset saves images too, the sampling frequency reduces to 70 Hz. In this context, the dataset has 8400×70 . Each dataset instance has five fingers and three channels X , Y , and Z data [23].

3.3 Feature Extraction Functions

The accuracy of models directly depends on the number of features. And in the proposed approach, we define 17 functions for feature extraction. These functions are based on statistical measures of central tendency, scatter, amplitude, wavelength, etc. The following describes the feature extraction functions used.

Variance (VAR) measures the signal amplitude and the power, Root Mean Square (RMS) is a meaningful way of calculating the average of values over a period of time, Mean Absolute Value (MAV) define the average of the summation of absolute value of signal, Enhanced Mean Absolute Value (EMAV) is an extension of MAV define a p value this value is used to select a region of the signal, Modified Mean Absolute Value (MMAV) is an extension of MAV this assign the weight window function, Modified Mean Absolute Value 2 (MMAV2) is another extension of MAV feature by assigning the continuous weight window function, Difference Absolute Standard Deviation Value (DASDV) is the square root of the average of the difference between the squared contiguous values, Enhanced Wavelength (EWL) is an extension of WL define a p value this value is used to select a region of the signal, Average Amplitude Change (AAC) measures the average change of the signal amplitude, Wavelength (WL) can be calculated by simplifying the cumulative length of waveform summation, Slope Sign Change (SSC) determines the number of times in which the number of wave form changes sign, Detector Log (DL) is good at estimating the exerted force, Pulse Percentage Rate (MYOP) this function is adapted by leap motion controller signal, amplitude Willinson (WAMP) acts as an indicator of the firing of motor unit potentials. Simple Square Integral (SSI) is defined as the summation of square values of signal amplitude, Standard Deviation (SD), Mean Value (MV) [25].

3.4 Feature Selection Functions

The feature selection methods used in the development of this paper are described in this section. Feature selection methods are necessary when working with machine learning, as described in the previous sections. In general, machine learning models present large feature vectors. The large size of the feature vectors is associated with a large dimensionality of the problem. In this context, feature selection is used as a dimensionality reduction technique. This process allows to rank the features according to their maximum relevance by Pattern Recognition community in the last decades [1, 18] and graph isomorphism is its most restrictive form requiring the mapping between the nodes of the two graphs while preserving node adjacency as well as non-adjacency.

3.4.1 Filter Methods

The **minimal-redundancy maximum-relevance** algorithm belongs to the family of wrapper methods. The algorithm finds an optimal set of mutually and maximally exclusive features and can effectively represent the response variable. The algorithm quantifies redundancy and relevance. Also, the algorithm observes the dependence of a pair of features, and the dependence of a variable against the response variable [19]. The algorithm for maximizing relevance looks at the dependence of a variable against the response variable, as presented in Eq. (1) [26].

$$V_s = \frac{1}{|S|} \sum_{x \in S} I(x, y) \quad (1)$$

where V_s represents the maximization value, $|S|$ represents the set of features or predictor variables, (x, y) represents the predictor variable and the response variable, respectively.

While for the minimization, the dependence between the pair of predictor variables is observed, as presented in Eq. (2) [26].

$$W_s = \frac{1}{|S|^2} \sum_{x, z \in S} I(x, z) \quad (2)$$

where W_s represents the value of minimization of the dependence between the predictor variables, and (x, z) represents the predictor variables.

The algorithm returns an index and an associated score. The score defines the importance of the predictor variable. Also, the algorithm uses a heuristic algorithm to determine the score.

In this context, the MRMR algorithm is fed with data extracted from the functions in the following order: MAV, EMAV, MMAV, MMAV, MMAV2, VAR, RMS, DASDV, SD, MV, ACC, WL, EWL, LD, SSC, MYOP, WA, SSI.

Once processed, the algorithm returns the score of the most significant predictor variables, as shown in Fig. 4.

The order of the variables corresponds to VAR, SSC, EWL, SD, WA, WL, LD, DASDV, EMAV, MYOP, MAV, ACC, MMAV, SSI, MMAV2, MV, RMS.

Sequential selects a subset of features from the data matrix \mathbf{X} that best predicts the data in \mathbf{y} . Where \mathbf{X} is a matrix data and \mathbf{y} are the classes. This method sequentially selects features until there is no improvement in prediction. The prediction of Sequential is based on a function that defined the criterion that permits selecting the best subset of features. The method executes two processes: first split the dataset in training and testing, and second, the method executes a cross-validation process. In this case, the function sums the values returned by the function and divides that sum by the total number of test observations. It then uses that mean value to evaluate each candidate feature subset. In this sense, the method views the number of misclassified observations for classification.

After computing the mean criterion values for each candidate feature subset, the method chooses the candidate feature subset that minimizes the mean criterion value. This process continues until adding more features does not decrease the criterion.

In this paper, the method is fed with the data matrix $M \times 17$, the values of the features correspond to the values of the randomly ordered feature selection functions. Table 1 presents in the first column the initial functions, while the second column presents the ordered feature selection functions based on the score returned by the method.

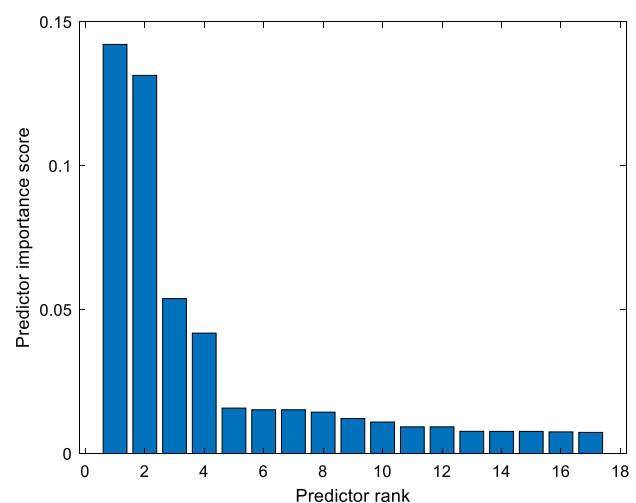


Fig. 4 Order of feature extraction functions after running feature selection functions with the algorithm MRMR

Table 1 Score of feature extraction functions using Sequential method

Initial order of feature extractor functions	Order according to the score after applying the Sequential feature selection method	Score
MAV	MYOP	0.000768282
EMAV	LD	0.000723214
MMAV	EWL	0.000658022
MMAV2	WL	0.00061593
VAR	MAV	0.000597222
RMS	SSC	0.000588435
DASDV	MMAV	0.000586876
SD	EMAV	0.000584325
MV	MV	0.000583475
SSC	VAR	0.000583333
WL	MMAV2	0
EWL	RMS	0
LD	DASDV	0
AAC	SD	0
MYOP	AAC	0
WA	WA	0
SSI	SSI	0

3.4.2 Wrapper Methods

The **neighbor component analysis** algorithm is a non-parametric method used to perform feature selection and belongs to the wrapper method family. It learns the weights from the original features input data space $\mathbf{X} = \{(\mathbf{x}_i, y_i), \dots, (\mathbf{x}_n, y_n)\}$ using a diagonal adaptation of NCAp. Also, it uses the distance metric to find a linear transformation of the features that maximize the average classification accuracy.

The process of NCAp is similar to KNN where k is equal to one. KNN is based on a distance metric between near neighbors. In this sense, NCAp also has a distance metric, and for measuring the distance, NCA does select a \mathbf{x}_i randomly as a reference point to others features vectors \mathbf{x}_j . It obtains $d_{ij} = (\mathbf{x}_i, \mathbf{x}_j)$, where d represents the distance. Moreover, NCA does use different parameters for adjusting the classification accuracy value. The parameters are regularization value (λ), optimization function, fit method, standardized process [27].

In this work, for finding the best λ value, we generate an array of 50 values equally spaced. The array values are from 0 with steps of 3. Each value multiplies the standard deviation of the total observation number. And it is split for total observations numbers. Then, for each value of the array running the NCA algorithm. And to avoid bias, we use cross-validation with k -fold equal to five. Also, this function use as an optimization function stochastic gradient descent. Finally, we present the order of the most significant feature extractor functions reported according to Table 2.

Also, A second model is generated using NCAp only with the observations and classes as parameters. This

Table 2 Score of feature extraction functions using NCAp with parameters

Ranking of feature extraction functions		Feature extractor functions sorted according to score	
Functions	Score	Functions	Score
MAV	1.489	LD	3.819
EMAV	2.869	EMAV	2.869
MMAV	2.413	MMAV2	2.820
MMAV2	2.820	MMAV	2.413
VAR	0.029	SD	2.256
RMS	0.045	DASDV	2.229
DASDV	2.229	MYOP	1.865
SD	2.256	ACC	1.760
MV	1.489	WL	1.665
ACC	1.760	EWL	1.665
WL	1.665	MAV	1.489
EWL	1.665	MV	1.489
LD	3.819	VAR	0.045
SSC	2.24×10^{-58}	SSI	0.029
MYOP	1.865	RMS	0.029
WA	6.06×10^{-15}	SSC	6.06×10^{-15}
SSI	0.029	WA	2.24×10^{-58}

model takes as input only the matrix of observations (data-SetFeatures) and their respective labels \mathbf{y} . This process obtains a score of the most significant predictor functions. Table 3.

Table 3 Score of feature extraction functions using NCAsp without parameters

Ranking of feature extraction functions		Feature extractor functions sorted according to score	
Functions	Score	Functions	Score
MAV	9.23×10^{-4}	LD	9.202
EMAV	6.97×10^{-3}	MYOP	8.740
MMAV	1.18×10^{-3}	WL	3.577
MMAV2	2.97×10^{-3}	EWL	3.381
VAR	3.12×10^{-09}	SSI	2.246
RMS	2.74×10^{-09}	EMAV	6.97×10^{-3}
DASDV	2.76×10^{-07}	MMAV2	2.97×10^{-3}
SD	2.65×10^{-3}	SD	2.65×10^{-3}
MV	9.236×10^{-4}	MMAV	1.18×10^{-3}
ACC	4.56×10^{-10}	MAV	9.23×10^{-4}
WL	3.577	MV	9.236×10^{-4}
EWL	3.381	DASDV	2.76×10^{-07}
LD	9.202	VAR	3.12×10^{-09}
SSC	3.26×10^{-41}	RMS	2.74×10^{-09}
MYOP	8.740	ACC	4.56×10^{-10}
WA	2.83×10^{-159}	SSC	3.26×10^{-41}
SSI	2.246	WA	2.83×10^{-159}

3.4.3 Embedded Methods

Another algorithm used in this work for feature selection is **Relief-F**. This algorithm ranks the features by finding the weights of the most significant predictors. Relief-F works similarly to the kNN algorithm. In this sense, this algorithm depends on the k value. We will work with two values from k , with k equal to three and with k equal to five. It is necessary to mention that when k tends to one, the values of the weights are not reliable [28].

Relief-F initially starts with weights in zero. Then, randomly select an observation, and concerning this observation, find the nearest observation according to the k value and its classes. This process is iterative.

The Eq. (3) in [26] updates the weights when the class of the selected observation and the predicted observation class are the same.

$$w_j^i = w_j^{i-1} - \frac{\Delta_j(\mathbf{x}_r, \mathbf{x}_q)}{m} \cdot d_{rq} \tag{3}$$

where \mathbf{x}_r , and \mathbf{x}_q are feature vectors labels in the same class.

Similarly, it is necessary to update the weights when the features vectors \mathbf{x}_r , and \mathbf{x}_q belong to classes with different labels. According to Eq. (4) [26].

$$w_j^i = w_j^{i-1} + \frac{p_{y_q}}{1 - p_{y_r}} \cdot \frac{\Delta_j(\mathbf{x}_r, \mathbf{x}_q)}{m} \cdot d_{rq} \tag{4}$$

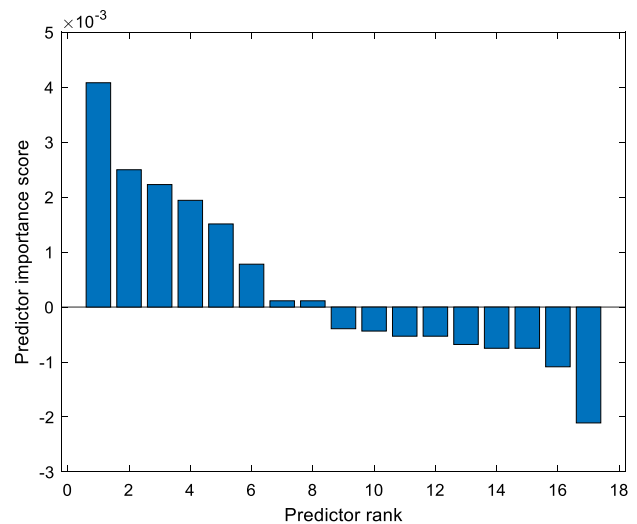


Fig. 5 Ranking of most important feature using Relief-F with $k = 5$

where w_j^i represent the weights of the predictor for i th iteration. p_{y_r} represent a priori probability of the class which \mathbf{x}_r belongs. While p_{y_q} represent a priori probability of the class which \mathbf{x}_q belongs. Besides, m represents the iterations number given for updating the weights. Finally, $\Delta_j(\mathbf{x}_r, \mathbf{x}_q)$ is a difference in the predictors' value between the observations \mathbf{x}_r , and \mathbf{x}_q .

This study works with k equal to five. The algorithm is fed with the feature extractor functions without a specific order. After applying the algorithm, we obtain the following ranking. LD, SSC, EWL, MYOP, DASDV, MMAV, ACC, WL, EMAV, SD, MAV, MV, RMS, SSI, VAR, MMAV2, WA. It is shown in Fig. 5.

Using **DT**, the definition of induction tells us that this task consists of extracting implicit general knowledge from particular observations and experiences.

In DT learning, the hypothesis space is the set of all predictor attributes or features. The DT induction task consists of finding the tree that best fits the available, already classified, example data. For each class, a branch is found in the tree that satisfies the conjunction of attribute values represented by the branch.

When generating a DT a crucial element is the method of attribute selection, which determines which criteria are used to generate the different branches of the tree, which determine the classification into the different classes.

Attribute selection is based on the calculation of the entropy value and the information gain. Table 4 shows the order of the feature extraction functions based on the selection of the most relevant attribute using DT.

Table 4 Order of feature extraction functions using decision tree

Initial order of feature extractor functions	Order according to the score after applying the Sequential feature selection method
MAV	MYOP
EMAV	LD
MMAV	WL
MMAV2	EWL
VAR	DASDV
RMS	SD
DASDV	MMAV2
SD	MAV
MV	SSC
SSC	VAR
WL	MMAV
EWL	EMAV
LD	WA
AAC	AAC
MYOP	MV
WA	RMS
SSI	SSI

3.5 Machine Learning Algorithms

For proving the score of feature selection functions reported by studied algorithms. This study uses four machine learning algorithms, two parametric algorithms, *ANN* and *SVM*. And two non-parametric algorithms such as *KNN* and *DT*.

ANN is a mathematical model used for predicting systems performance. *ANN* was developed according to human brain functionality. Additionally, its architecture is similar to biological human neuron layers. Also, it is a non-deterministic algorithm due to the random weight's initialization. The *ANN* is widely used by it has high parallelism, fault and noise tolerance, and its capabilities of learning and generalization. [29].

The *ANN* architecture is formed by an input layer, one or more hidden layers, and output layers. The input layer receives the data retrieved from the environment. The hidden layers are composed of neurons or nodes. Each node has an activation function and processes the sum of the multiplied inputs by the weights. The output layer reports class 0 or class 1 if the *ANN* is binary, but if *ANN* is multiclass reports an array of probabilities, where each probability value corresponds to one category.

SVM is another algorithm of machine learning. It is used for classification or regression and belongs to the supervised learning family. The researchers use *SVM* because exist many hyperplanes that could split the classes. In this sense, *SVM* introduces the concept of a maximum margin classifier. Also, *SVM* is a deterministic algorithm and would

avoid local minima. Additionally, it is a binary algorithm, and for it to be considered multiclass is necessary to apply techniques such as one vs. one or one vs. all. In the same sense, *SVM* faces two issues when the classes are linearly separable and non-linearly separable [30].

SVM uses linear functions such as $\mathbf{w}\mathbf{x}_i + b \geq 1$. where \mathbf{w} and \mathbf{x} are vectors. The vectors closer to the hyperplane is called support vector, and the line touching the support vectors is considered the decision boundary. The distance between the two decision boundaries of a hyperplane is called the margin.

KNN is called a lazy algorithm. It is a non-parametric method because it does not involve parameter adjustment or estimation. Also, it is a deterministic algorithm. *KNN* identifies dynamically k observations of a dataset that are similar to a new observation. To identify similarities *KNN* uses a distance metric. The distances metrics could be Euclidean distance, Minkowski, or Mahalanobis distance between others. Also, it uses the Dynamic Time Warping (*DTW*) method [31, 32].

In this sense, *KNN* defines the neighbor number according to the k value. Besides, *KNN* represents a higher similarity when the distance between samples is small, so they are highly likely to belong to the same label [33].

DT is an algorithm used in machine learning for classification. *DT* divides the dataset into training and testing instances. Also, this algorithm selects one attribute from a set of training instances. The training is based on the calculation of the entropy and the gain of information. Next, build the *DT* using the training instance and the chosen attribute. Where each internal node tests an attribute x_i , each branch assigns an attribute value, and each leaf gives a class. The accuracy of the *DT* is measured with the testing instances. To classify a new input, traverse the tree from root to leaf and assigns the labeled [34].

4 Experiments

The experimentation was carried out in Alienware computer with six cores, and twelve logical processors of 3.4 GHz core i7 of sixth generations, 32 Gb of RAM, and Windows 10. The design of the experiments is based on the dataset described in the previous section. In all experiments, we measure the accuracy of training and testing of classification and testing of recognizing the hand gestures. The gestures were described in the last section. Also, we measure the time of processing.

In this sense, the acquired data are organized into cells $\{1,5\}$, where each cell contains each finger's X , Y , and Z channels, and each channel contains 70-time instants. The 70-time instants are limited: if less than 70 frames are obtained during the sampling time, an interpolation is

performed with the extracted data completed to 70 frames. In the same way, if the frames obtained exceed 70 frames, they are delimited, eliminating redundant data with an equal division, observing that the shape of the original signal does not change. Besides, the data on each channel are normalized between 0 and 1. In addition, the signal is filtered using a Butterworth filter with a sampling frequency of 70 Hz and a cutoff frequency of 18 Hz. This cutoff frequency makes the signal smooth without losing its original shape.

For experimentation, we use the signal composed of spatial positions and directions of the fingers. The algorithms for classification and recognition only use data from the three fingers, thumb, middle and pink. Also, each of these fingers presents data of three channels X , Y , and Z . Besides, we use cross-validation with a k fold equal to five. The technique used to extract the data and feed it to the classifiers is window splitting. The signal is divided into windows of 20 with a stride of 15, and each data window is delivered to the classifier, which returns a label. Finally, we have a vector of labels. And by majority vote, it returns the label the most time is repeated. Additionally, before giving the data to the algorithms, the data are mixed randomly. In this context, the algorithms used are ANN, SVM, KNN, and DT.

In this work, we use a feedforward ANN with two hidden layers. The first hidden layer uses ReLU as an activation function with 25 neurons. The second hidden layer use log-sig as an activation function and 15 neurons. The input of ANN is the number of features according to the number of combinations of feature selection functions. As optimization function use cross-entropy a gradient descends as a technic of weights adjust. Additionally, ANN uses 2000 iterations and a regularization factor of $1.0e1$.

Also, the SVM classifier needs to set up its hyperparameters as the kernel and scale. To this work, the kernel is gaussian, and the scale is of order ten. In the same sense, KNN was set up with k equal to three and Euclidean distance as a metric. Besides, DT use information gain technic for building the tree. Also, this algorithm for training uses 100 levels of deep.

In this context, we evaluate the combination feature extraction functions according to the score given by MRMR, Sequential, NCAp with adjusting parameters such as cross-validation, regularization, and the optimization function to sgd. Also, this work evaluates NCAsp algorithm without parameters. Also, it evaluates the Relief-F feature selection algorithm with k equal to five. Finally, we evaluate DT using entropy and gain information.

The experimentation consisted of ordering the feature extractor functions according to the score provided by each feature selection method. Then, the dataset is divided into windows. The windows are 20 with jumps of 15. To each window is applied the function with the highest score reported, and at the end, a vector of labels is obtained. The

label selected is by majority vote. Then the same process is executed, but with the feature selection function with the highest score and the second one, and so on, until all feature selection functions are combined. This process is repeated 10 times, and finally, a 10×17 matrix is obtained, where each cell corresponds to the accuracy. From where the average of the 10 experimentations is obtained, and then the maximum accuracy value is reported. Obtained the index of the maximum value that corresponds to the number of combinations of feature selector functions, as presented in the following pseudocode.

Algorithm 1

```

1  Fs = [VAR, SSC, EWL, SD, WA, WL, LD, DASDV,
        EMAV, MYOP, MAV, ACC, MMAV, SSI,
        MMAV2, MV, RMS];
2  Win = 20;
3  Step = 15;
4  Ds = data;
5  i = 1;
6  Alg = [ANN, SVM, KNN, DT]
7  for repetition = 1 :10
8      until (i = 1 : length(Fs))
9          for (j = 1 : (size(ds,2))/Win)
10             acc(j) = Alg(Fs(1:i)(ds(Win)));
11             end
12             acc(i,j) = mode(acc)
13             end
14  End

```

Table 5 shows the summary of the experiments. It shows the number of combinations represented by the variable idx. The maximum accuracy and the standard deviation of each model trained, tested, and recognized in each algorithm are also shown. In the same sense, the time taken to complete is shown.

Figure 6 shows the testing accuracy grouped by each method and evaluated for each proposed classification algorithm. In addition, the number of combinations of feature extractor functions with which the methods reach their maximum value is presented.

As can be seen, the difference in the reported accuracies is very small, and the standard deviation values overlap. This tells us that there is no significant difference between the methods in the algorithms evaluated. This leads to performing a hypothesis testing statistic to determine significant differences and define the method, the combination of feature selection functions, and the algorithm that best-performing return for the hand gesture recognition problem with Leap Motion Controller signals.

Figure 7 shows how all the methods evaluated in ANN have the highest accuracy value. In addition, ANNs present variability due to the stochastic resetting of the weights.

Table 5 Summary of the maximum training, testing, and recognition accuracy of feature selection methods evaluated in classification algorithms

		Training			Testing			Recognition			Time
		idx	Acc	std	idx	Acc	std	idx	Acc	std	
MRMR	ANN	17	98.3321	0.18	13	92.7588	1.4542	9	82.2712	1.8195	82.4106
	SVM	17	98.9481	0.1346	9	91.4133	1.6187	9	79.9152	2.4562	13.1187
	KNN	8	91.6	1.0897	8	91.6	1.0897	9	77.0909	1.9012	0.8654
	DT	16	89.903	0.9492	16	89.903	0.9492	16	75.2879	2.295	0.3029
NCA with parameters	ANN	17	98.3413	0.1854	10	92.7358	1.4824	13	83.3348	2.314	80.0652
	SVM	17	98.9481	0.1346	10	91.5273	1.4287	12	80.1212	3.0044	12.2362
	KNN	12	91.0182	1.8826	12	91.0182	1.8826	13	77.6667	2.662	0.6827
	DT	13	89.903	1.8007	13	89.903	1.8007	17	75.3636	2.3192	0.2627
NCA without parameters	ANN	17	98.3189	0.1822	4	92.2394	1.6819	15	81.6742	2.5675	90.9038
	SVM	17	98.9481	0.1346	9	91.3212	1.2372	12	79.8636	3.0508	15.7562
	KNN	4	90.1212	1.1411	4	90.1212	1.1411	4	75.0455	3.0813	0.6953
	DT	11	89.903	1.8129	11	89.903	1.8129	17	75.3333	2.2674	0.2641
Relieff	ANN	17	98.3319	0.1306	13	92.8776	1.7726	13	83.403	2.5082	88.0529
	SVM	17	98.9481	0.1346	10	91.1576	1.7419	12	80.3788	2.5404	12.5256
	KNN	3	91.3939	0.782	3	91.3939	0.782	13	76.5909	2.7871	0.6607
	DT	13	89.8424	1.3255	13	89.8424	1.3255	13	76.1364	3.1359	0.2379
Sequentials	ANN	16	98.4711	0.1135	14	93.0788	1.5789	14	83.5152	1.8538	67.5646
	SVM	17	98.9481	0.177	15	90.7758	1.6168	15	79.8182	3.1022	12.1294
	KNN	15	90.7515	0.8944	15	90.7515	0.8944	14	76.6515	2.7656	0.595
	DT	14	89.8667	1.143	14	89.8667	1.143	16	75.3939	2.4061	0.1911
DT	ANN	16	98.5037	0.1199	6	93.0545	1.5064	16	83.2273	2.3045	44.7923
	SVM	17	98.9481	0.1346	7	91.3818	1.6461	15	79.8636	3.2937	6.8531
	KNN	13	90.8242	0.7886	13	90.8242	0.7886	15	76.6212	2.3268	0.5914
	DT	15	89.8424	1.2179	15	89.8424	1.2179	15	75.3788	2.291	0.1858

Standard deviation, number of feature combinations, and processing time

Accuracy and Number of Feature Extractor Functions Combined per Classifier and Method

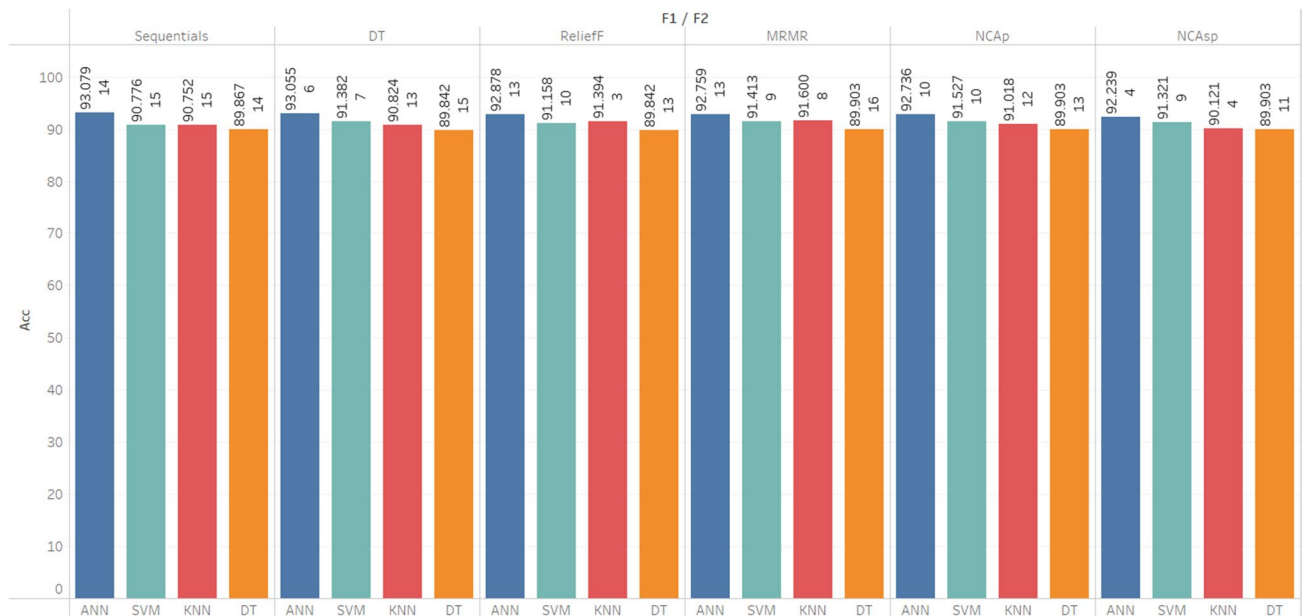


Fig. 6 Summary of the maximum training, testing, and recognition accuracy of feature selection methods evaluated in classification algorithms. Standard deviation, number of feature combinations, and processing time

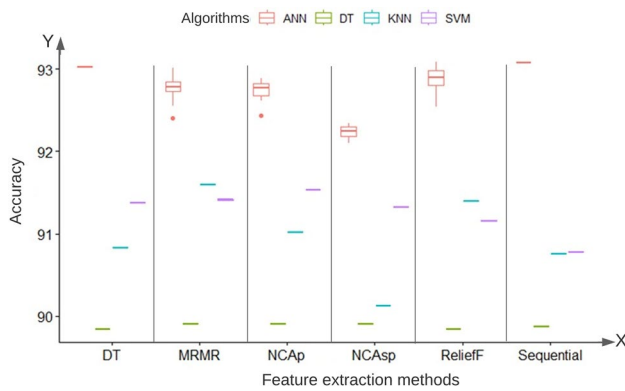


Fig. 7 The boxplot shows the variability of the data and the algorithms with the highest accuracy values

We filter the methods evaluated by ANN for the following analysis. The filtering of the methods evaluated in each algorithm is shown in Fig. 8.

The differences in the accuracies that are shown are small. The number of combinations of feature selection functions used to achieve these accuracies would be the analysis factor to demonstrate the difference (Fig. 8).

However, the statistical analysis is presented. In fact, there is no significant difference between the Sequential and DT

methods, which are the most accurate. However, Sequential uses a combination of 14 feature selector functions, while DT uses six functions. This leads to analyzing the response time of the algorithms with each of the methods (Fig. 9).

In this sense the processing time is a decision factor when choosing a feature selection method in the context of hand gesture recognition using infrared signals emitted by the Leap Motion Controller (Table 6).

5 Conclusion

This paper analyzes feature selection methods such as MRMR, Sequential, NCAp, NCAsp, Relief-F, and DT. These methods report a score that represents the significance of the features. In addition, we propose a hand gesture recognition model to validate the significance of the features. The model consists of five modules such as: data acquisition, pre-processing, feature extraction, classification, and post-processing. In the feature extraction module, the proposed methods are applied. While in the classification module, ANN, SVM, K-NN, and DT are used. The data set used for this study consists of five gestures. These gestures are open hand, fist, wave in, wave out, and pinch. The dataset contains 1680 observations for each gesture,

Accuracy and combination number of feature extractor functions per classifier and method

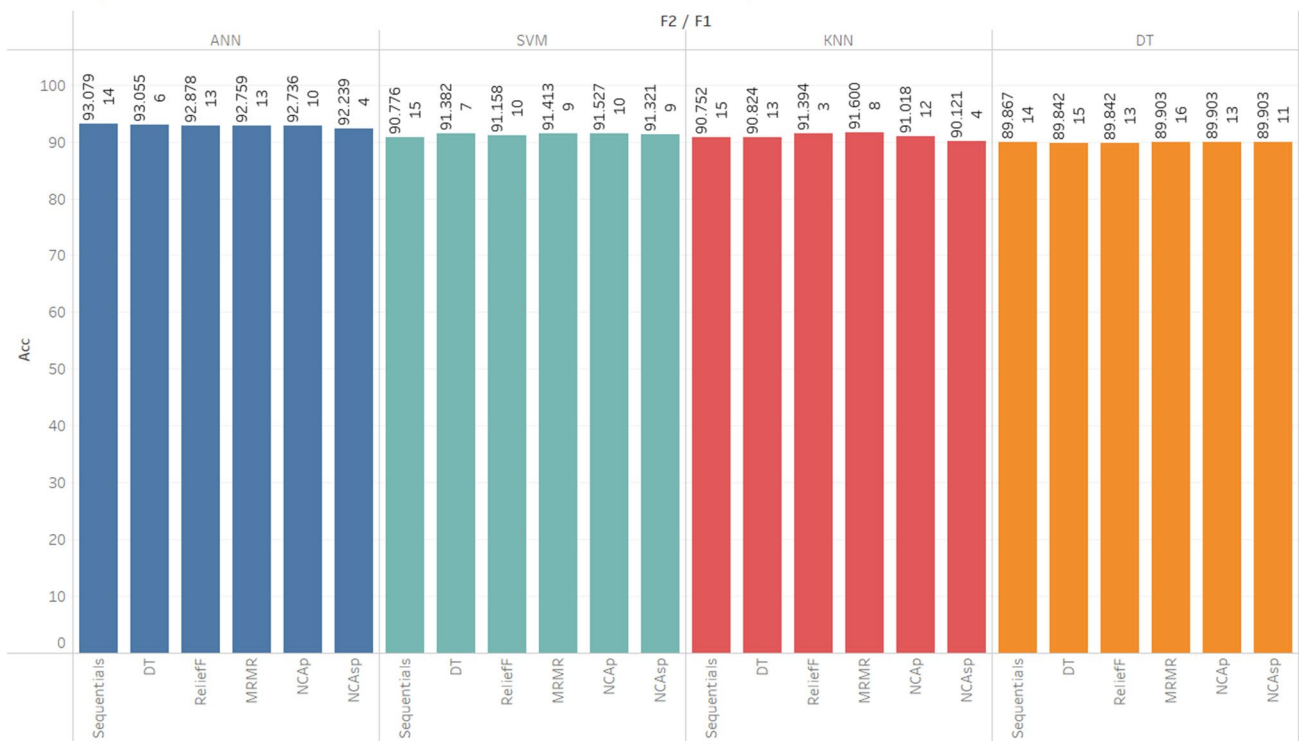


Fig. 8 The figure presents the evaluation of the accuracy of the feature selection methods, with the number of combinations of features grouped by the classification algorithm

Fig. 9 The figure presents the evaluation of the accuracy of the feature selection methods, with the number of combinations of features grouped by the classification algorithm

	diff	lwr	upr	p adj
MRMR-DT	-0.27151212	-0.466678543	-0.07634570	0.0019954
NCAp-DT	-0.29454242	-0.489708846	-0.09937600	0.0006739
NCAsp-DT	-0.79090606	-0.995015990	-0.58679613	0.0000000
Relief-DT	-0.15272424	-0.347890664	0.04244218	0.2037165
Sequential-DT	0.04848788	-0.169714814	0.26669057	0.9851666
NCAp-MRMR	-0.02303030	-0.192049382	0.14598878	0.9984852
NCAsp-MRMR	-0.51939394	-0.698665745	-0.34012213	0.0000000
Relief-MRMR	0.11878788	-0.050231201	0.28780696	0.3093554
Sequential-MRMR	0.32000000	0.124833578	0.51516642	0.0001953
NCAp-NCAp	-0.49636364	-0.675635442	-0.31709183	0.0000000
Relief-NCAp	0.14181818	-0.027200898	0.31083726	0.1464297
Sequential-NCAp	0.34303030	0.147863881	0.53819673	0.0000620
Relief-NCAsp	0.63818182	0.458910012	0.81745362	0.0000000
Sequential-NCAsp	0.83939394	0.635284010	1.04350387	0.0000000
Sequential-Relief	0.20121212	0.006045699	0.39637854	0.0398463

Table 6 Presentation of the difference in processing time, due to the number of combined feature selection functions

Test of classification using ANN			
Methods	# Feature extraction	Testing	Time of processing sec
Sequential	14	93.079	67.5646
DT	6	93.055	44.7923

totaling 8400. The input data is an 8400×17 matrix. The 17 predictors are formed by computing the functions MAV, EMAV, MMAV, MMAV2, VAR, RMS, DASDV, SD, MV, ACC, WL EWL, LD, SSC, MYOP, WA, SSI. The result of the evaluation of the feature selection methods shows that all methods perform better with ANN. This is because the classification and recognition accuracy are the highest with respect to the other machine learning methods such as SVM, K-NN, and DT. However, the differences in the accuracy of the feature selection methods evaluated in an ANN are insignificant because the standard deviations overlap. In this context, a test statistic is generated to determine whether the feature selection methods differ statistically. The study shows that the Sequential feature selection method, with an accuracy of 93.079%, and DT, with an accuracy of 93.055%, do not have significant differences. In this context, the response time is evaluated because Sequential combines 14 feature extractor functions to achieve the maximum accuracy value. At the same time, DT combines six feature extractor functions. The reported time is measured in milliseconds. In addition, the time is measured after the user finishes executing the gesture until the algorithm returns a response or label. In this sense, the execution time of the combination of feature extractor functions presented by Sequential evaluated on

an ANN is 67.5646 milliseconds. While under the same conditions, the time reported by the combination of feature extractor functions presented by DT is 44.7923 milliseconds. The limitation of the presented work would be in the accuracy obtained. Future work would be directed to increase the accuracy value, probably working on post-processing techniques.

Acknowledgements The authors would like to express their gratitude to the Escuela Politécnica Nacional and its doctoral program in computer science, for having the best human resources for the development of its students. Thanks are also due to the Universidad Técnica de Ambato, for providing the facilities for continuous improvement.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by RN, and MB. The first draft of the manuscript was written by RN and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The study is funded by Escuela Politécnica Nacional.

Availability of Data and Materials The dataset is available.

Declarations

Conflict of Interest The authors declare that there is no conflict of interest. They also declare that the article has not been published in other journals or conferences.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Naguri, C.R., Bunesco, R.C.: Recognition of dynamic hand gestures from 3D motion data using LSTM and CNN architectures. In: Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2017, vol. 2018, pp. 1130–1133 (2018). <https://doi.org/10.1109/ICMLA.2017.00013>.
- Lee, G.C., Yeh, F., Hsiao, Y.: Kinect-based Taiwanese sign-language recognition system. *Multimed. Tools Appl.* **151**, 261–279 (2016). <https://doi.org/10.1007/s11042-014-2290-x>
- Dynamic, A.I., Warping, T.: Author's accepted manuscript an image-to-class dynamic time warping approach for both 3D static and trajectory hand gesture recognition. *Pattern Recogn.* (2016). <https://doi.org/10.1016/j.patcog.2016.01.011>
- Ding, Z., Chen, Y., Chen, Y., Wu, X.: Similar hand gesture recognition by automatically extracting distinctive features. *Int. J. Control Automat. Syst.* **15**(4), 1770–1778 (2017)
- Tang, A.O., Lu, K.E., Wang, Y., Huang, J.I.E., Li, H.: A real-time hand posture recognition system using deep neural networks. *ACM Trans. Intell. Syst. Technol.* **6**(2), 1–23 (2015)
- Normani, N., et al.: A machine learning approach for gesture recognition with a lensless smart sensor system. In: IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 136–139 (2018)
- Bataineh, M.H.: Artificial neural network for studying human performance. ProQuest Diss. Theses, vol. 1518568, p. 179 (2012) [Online]. Available <http://ezproxy.net.ucf.edu/login?>, <http://search.proquest.com/docview/1039557097?accountid=100035Cn>, http://sfx.fcla.edu/ucf?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&genre=dissertations+&+these&sid=ProQ:ProQuest+Dissertations
- Jia, J., Zhao, C., Yi, W.: Real-time hand gestures system based on leap motion static gestures. *Concurr. Comput. Pract. Exp.* **31**, e4898 (2018). <https://doi.org/10.1002/cpe.4898>
- Alpaydin, E.: Introduction to Machine Learning, 3rd., vol. ث قفتنى، no. ث قفتنى. The MIT Press, London (2020)
- Bishop, C.M.: *Neural Networks for Pattern Recognition*. OXFORD University Press, New York (2005)
- Destrero, A., Mosci, S., De Mol, C., Verri, A., Odone, F.: Feature selection for high-dimensional data. *Comput. Manage. Sci.* **6**(1), 25–40 (2009). <https://doi.org/10.1007/s10287-008-0070-7>
- Schulte, R.V., Prinsen, E.C., Hermens, H.J., Buurke, J.H.: Genetic algorithm for feature selection in lower limb pattern recognition. *Front. Robot. AI* (2021). <https://doi.org/10.3389/frobt.2021.710806>
- Fok, K.Y., Ganganath, N., Cheng, C.T., Tse, C.K.: A real-time ASL recognition system using leap motion sensors. In: Proc. 2015 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. Cyber C 2015, pp. 411–414 (2015). <https://doi.org/10.1109/CyberC.2015.81>
- Midarto Dwi Wibowo, I.N.: 2017 International Conference on Information & Communication Technology and System (ICTS), pp. 67–72 (2017)
- Guerra-Segura, E., Ortega-Pérez, A., Travieso, C.M.: In-air signature verification system using leap motion. *Expert Syst. Appl.* (2021). <https://doi.org/10.1016/j.eswa.2020.113797>
- Marin, G., Dominio, F., Zanuttigh, P.: Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimed. Tools Appl.* **75**(22), 14991–15015 (2016). <https://doi.org/10.1007/s11042-015-2451-6>
- Borysova, A.: Leap Motion Controller for South African Sign Language Recognition (2016) [Online]. Available https://projects.uct.ac.za/honsproj/cgi-bin/view/2017/borysova_kooverjee_versfeld.zip/supporting/final_paper_anna.pdf
- Sooai, A.G., et al.: Comparison of recognition accuracy on dynamic hand gesture using feature selection. In: 2018 Int. Conf. Comput. Eng. Netw. Intell. Multimedia, CENIM 2018—Proceeding, pp. 270–274 (2018). <https://doi.org/10.1109/CENIM.2018.8711397>
- Yu, L., Liu, H.: Redundancy based feature selection for microarray data. In: KDD-2004—Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 2, pp. 737–742 (2004). <https://doi.org/10.1145/1014052.1014149>
- Perez, R.: Una revisión de algoritmos de selección de atributos rma.pdf. *Revista Cubana de Ciencias Informáticas, Cuba*, p. 30 (2013) [Online]. Available: <http://rcci.uci.cu/>
- Butt, A.H., et al.: Objective and automatic classification of Parkinson disease with leap motion controller. *Biomed. Eng. Online* **17**(1), 1–21 (2018). <https://doi.org/10.1186/s12938-018-0600-7>
- Nogales, R.E., Benalcázar, M.E.: Hand gesture recognition using machine learning and infrared information: a systematic literature review. *Int. J. Mach. Learn. Cybern.* (2021). <https://doi.org/10.1007/s13042-021-01372-y>
- Nogales, R., Benalcázar, M.E., Toalumbo, B., Palate, A., Martínez, R., Vargas, J.: Construction of a dataset for static and dynamic hand tracking using a non-invasive environment. *Adv. Intell. Syst. Comput.* **1307**, 185–197 (2021). https://doi.org/10.1007/978-981-33-4565-2_12
- Nogales, R., Benalcázar, M.: Real-time hand gesture recognition using the leap motion controller and machine learning. In: IEEE Latin American Conference on Computational Intelligence (LACCI), pp. 1–6 (2019)
- Too, J., Abdullah, A.R., Saad, N.M.: Classification of hand movements based on discrete wavelet transform and enhanced feature extraction. *Int. J. Adv. Comput. Sci. Appl.* **10**(6), 83–89 (2019). <https://doi.org/10.14569/ijacsa.2019.0100612>
- The MathWorks Inc.: Optimization Toolbox version: 9.4 (R2022a). The MathWorks Inc., Massachusetts (2022)
- Yang, W., Wang, K., Zuo, W.: Neighborhood component feature selection for high-dimensional data. *J. Comput.* **7**(1), 162–168 (2012). <https://doi.org/10.4304/jcp.7.1.161-168>
- Robnik, M., Konenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. *Mach. Learn.* **53**(1–2), 23–69 (2003)
- Basheer, I.A., Hajmeer, M.: Artificial neural networks: fundamentals, computing, design, and application. *J. Microbiol. Methods* **43**(1), 3–31 (2000). [https://doi.org/10.1016/S0167-7012\(00\)00201-3](https://doi.org/10.1016/S0167-7012(00)00201-3)
- Jakkula, V.: Tutorial on Support Vector Machine (SVM). Sch. EECS, Washingt. State Univ., pp. 1–13 (2011) [Online]. Available <http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf>
- Zhang, Z.: Introduction to machine learning: K-nearest neighbors. *Ann. Transl. Med.* **4**(11), 218 (2016). <https://doi.org/10.21037/atm.2016.03.37>
- Algorithms, N.N.: Lecture 1 k-Nearest Neighbor Algorithms for Classification and Prediction, pp. 1–6 [Online]. Available: <https://ocw.mit.edu/courses/sloan-school-of-management/15-062-data-mining-spring-2003/lecture-notes/knn3.pdf>
- Zhu, Z.: K-Nearest Neighbors (KNN) Classification with Different Distance Metrics, pp. 1–14 (2020)
- Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **21**(3), 660–674 (1991). <https://doi.org/10.1109/21.97458>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.