

Article

A Many-Objective Simultaneous Feature Selection and Discretization for LCS-Based Gesture Recognition

Martin J.-D. Otis *  and Julien VandewynckelLAR.i Lab, University of Quebec at Chicoutimi, Saguenay, QC G7H 2B1, Canada;
julien.vandewynckel1@uqac.ca

* Correspondence: martin_otis@uqac.ca

Abstract: Discretization and feature selection are two relevant techniques for dimensionality reduction. The first one aims to transform a set of continuous attributes into discrete ones, and the second removes the irrelevant and redundant features; these two methods often lead to be more specific and concise data. In this paper, we propose to simultaneously deal with optimal feature subset selection, discretization, and classifier parameter tuning. As an illustration, the proposed problem formulation has been addressed using a constrained many-objective optimization algorithm based on dominance and decomposition (C-MOEA/DD) and a limited-memory implementation of the warping longest common subsequence algorithm (WarpingLCSS). In addition, the discretization sub-problem has been addressed using a variable-length representation, along with a variable-length crossover, to overcome the need of specifying the number of elements defining the discretization scheme in advance. We conduct experiments on a real-world benchmark dataset; compare two discretization criteria as discretization objective, namely Ameva and ur-CAIM; and analyze recognition performance and reduction capabilities. Our results show that our approach outperforms previous reported results by up to 11% and achieves an average feature reduction rate of 80%.



Citation: Otis, M.J.-D.; Vandewynckel, J. A Many-Objective Simultaneous Feature Selection and Discretization for LCS-Based Gesture Recognition. *Appl. Sci.* **2021**, *11*, 9787. <https://doi.org/10.3390/app11219787>

Academic Editor: Keun-Chang Kwak

Received: 9 September 2021

Accepted: 15 October 2021

Published: 20 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: many-objective optimization; evolutionary computation; discretization; feature selection; variable-length problem; longest common subsequence

1. Introduction

Gestures are composed of multiple body-part motions and can form activities [1]. Hence, gesture recognition offers a wide range of applications, including inter alia, fitness training, human robot and computer interaction, security, and sign language recognition. Likewise, gesture recognition is employed in ambient assisted living systems for tackling burgeoning and worrying public healthcare problems, such as autonomous living for people with dementia and Parkinson's disease. Although a large amount of work has been conducted on image-based sensing technology, camera and depth sensors are limited to the environment in which they are installed. Moreover, they are sensitive to obstructions in the field of vision, variation in luminous intensity, reflection, etc. In contrast, wearable sensors and mobile devices are more suitable for monitoring ambulatory activities and physiological signals.

In a supervised context, a wide range of action or gesture recognition techniques has been explored using wearable sensors. k-Nearest Neighbor (k-NN) might be the most straightforward classifier to utilize since it does not learn but searches the closest data in the training data using a given distance function. Even though conventional k-NN achieves good performance, it suffers from lack of ability to deal with these problems: low attribute and sample noise tolerance, high-dimensional spaces, large training dataset requirements, and imbalances in the data. Yu et al. [2] recently proposed a random subspace ensemble framework based on hybrid k-NN to tackle these problems, but the classifier has not yet been applied to a gesture recognition task. Hidden Markov Model (HMM) is the most

traditional probabilistic method used in the literature [3,4]. However, computing transition probabilities necessary for learning model parameters requires a large amount of training data. HMM-based techniques may also not be suitable for hard real-time (synchronized clock-based) systems due to its latency [5]. Since data sets are not necessarily large enough for training, Support Vector Machine (SVM) is a classical alternative method [6–8]. SVM is, nevertheless, very sensitive to the selection of its kernel type and parameters related to the latter. There are novel dynamic Bayesian networks often used to deal with sequence analysis, such as recurrent neural networks (e.g., LSTMs) [9] and deep learning approach [10], which should become more popular in the next years.

Dynamic Time Warping (DTW) is one of the most utilized similarity measures for matching two time-series sequences [11,12]. Often reproached for being slow, Rakthananon et al. [13] demonstrated that DTW is quicker than Euclidean distance search algorithms and even suggests that the method can spot gestures in real time. However, the recognition performance of DTW is affected by the strong presence of noise, caused by either segmentation of gestures during the training phase or gesture execution variability.

The longest common subsequence (LCSS) method is a precursor to DTW. It measures the closeness of two sequences of symbols corresponding to the length of the longest subsequence common to these two sequences. One of the abilities of DTW is to deal with sequences of different lengths, and this is the reason why it is often used as an alignment method. In [14], LCSS was found to be more robust in noisy conditions than DTW. Indeed, since all elements are paired in DTW, noisy elements (i.e., unwanted variation and outliers) are also included, while they are simply ignored in the LCSS. Although some image-based gesture recognition applications can be found in [15–17], not much work has been conducted using non-image data. In the context of crowd-sourced annotations, Nguyen-Dinh et al. [18] proposed two methods, entitled SegmentedLCSS and WarpingLCSS. In the absence of noisy annotation (mislabeling or inaccurate identification of the start and end times of each segment), the two methods achieve similar recognition performances on three data sets compared with DTW- and SVM-based methods and surpass them in the presence of mislabeled instances. Extensions were recently proposed, such as a multimodal system based on WarpingLCSS [19], S-SMART [20], and a limited memory and real-time version for resource constrained sensor nodes [21]. Although the parameters of these LCSS-based methods should be application-dependent, they have so far been empirically determined and a lack of design procedure (parameter-tuning methods) has been suggested.

In designing mobile or wearable gesture recognition systems, the temptation of integrating many sensing units for handling complex gesture often negates key real-life deployment constraints, such as cost, power efficiency, weight limitations, memory usage, privacy, or unobtrusiveness [22]. The redundant or irrelevant dimensions introduced may even slow down the learning process and affect recognition performance. The most popular dimensionality reduction approaches include feature extraction (or construction), feature selection, and discretization. Feature extraction aims to generate a set of features from original data with a lower computational cost than using the complete list of dimensions. A feature selection method selects a subset of features from the original feature list. Feature selection is an NP-hard combinatorial problem [23]. Although numerous search techniques can be found in the literature, they fail to avoid local optima and require a large amount of memory or very long runtimes. Alternatively, evolutionary computation techniques have been proposed for solving feature selection problem [24]. Since the abovementioned LCSS technique directly utilizes raw or filtered signals, there is no evidence on whether we should favour feature extraction or selection. However, these LCSS-based methods impose the transformation of each sample from the data stream into a sequence of symbols. Therefore, a feature selection coupled with a discretization process could be employed. Similar to feature selection, discretization is also an NP-hard problem [25,26].

In contrast to the feature selection field, few evolutionary algorithms are proposed in the literature [25,27]. Indeed, evolutionary feature selection algorithms have the dis-

advantage of high computational cost [28] while convergence (close to the true Pareto front) and diversity of solutions (set of solutions as diverse as possible) are still two major difficulties [29].

Evolutionary feature selection methods focus on maximizing the classification performance and on minimizing the number of dimensions. Although it is not yet clear whether removing some features can lead to a decrease in classification error rate [24], a multiple-objective problem formulation could bring trade-offs. Discretization attribute literature aims to minimize the discretization scheme complexity and to maximize classification accuracy. In contrast to feature selection, these two objectives seem to be conflicting in nature [30].

A multi-objective optimization algorithm based on Particle swarm optimization (heuristic methods) can provide an optimal solution. However, an increase in feature quantities increases the solution space and then decreases the search efficiency [31]. Therefore, Zhou et al. 2021 [31] noted that particle swarm optimisation may find a local optimum with high dimensional data. Some variants are suggested such as competitive swarm optimization operator [32] and multiswarm comprehensive learning particle swarm optimization [33], but tackling many-objective optimization is still a challenge [29].

Moreover, particle swarm optimization can fall into a local optimum (needs a reasonable balance between convergence and diversity) [29]. Those results are similar to filter and wrapper methods [34] (more details about Filter and wrapper methods can be found in [31,34]). Yang et al. 2020 [29] suggest to improve computational burdens with a competition mechanism using a new environment selection strategy to maintain the diversity of population. Additionally, to solve this issue, since mutual information can capture nonlinear relationships included in a filter approach, Sharmin et al. 2019 [35] used mutual information as a selection criteria (joint bias-corrected mutual information) and then suggested adding simultaneous forward selection and backward elimination [36].

Deep neural networks such as CNN [37] are able to learn and select features. As an example, hierarchical deep neural networks were included with a multiobjective model to learn useful sparse features [38]. Due to the huge number of parameter, a deep learning approach needs a high quantity of balanced samples, which is sometimes not satisfied in real-world problems [34]. Moreover, as a deep neural network is a black box (non-causal and non-explicable), an evaluation of the feature selection ability is difficult [37].

Currently, feature selection and data discretization are still studied individually and not fully explored [39] using many-objective formulation. To the best of our knowledge, no studies have tried to solve the two problems simultaneously using evolutionary techniques for a many-objective formulation. In this paper, the contributions are summarized as follows:

1. We propose a many-objective formulation to simultaneously deal with optimal feature subset selection, discretization, and parameter tuning for an LM-WLCSS classifier. This problem was resolved using the constrained many-objective evolutionary algorithm based on dominance (minimisation of the objectives) and decomposition (C-MOEA/DD) [40].
2. Unlike many discretization techniques requiring a prefixed number of discretization points, the proposed discretization subproblem exploits a variable-length representation [41].
3. To agree with the variable-length discretization structure, we adapted the recently proposed rand-length crossover to the random variable-length crossover differential evolution algorithm [42].
4. We refined the template construction phase of the microcontroller optimized Limited-Memory WarpingLCSS (LM-WLCSS) [21] using an improved algorithm for computing the longest common subsequence [43]. Moreover, we altered the recognition phase by reprocessing the samples contained in the sliding windows in charge of spotting a gesture in the steam.

5. To tackle multiclass gesture recognition, we propose a system encapsulating multiple LM-WLCSS and a light-weight classifier for resolving conflicts.

The main hypothesis is as follows: using the constrained many-objective evolutionary algorithm based on dominance, an optimal feature subset selection can be found. The rest of the paper is organized as follows: Section 2 states the constrained many-objective optimization problem definition, exposes C-MOEA/DD, highlights some discretization works, presents our refined LM-WLCSS, and reviews multiple fusion methods based on WarpingLCSS. Our solution encoding, operators, objective functions, and constraints are presented in Section 3. Subsequently, we present the decision fusion module. The experiments are described in Section 4 with the methodology and their corresponding evaluation metrics (two for effectiveness, including Cohen's kappa, and one for reduction). Finally, our system is evaluated and the results are discussed in Section 5.

2. Preliminaries and Background

In this section, we first briefly provide some basic definitions on the constrained many-objective optimization problem. We then describe a recently proposed optimization algorithm based on dominance and decomposition, entitled C-MOEA/DD. Additionally, we review evolutionary discretization techniques and successors of the well-known class-attribute interdependence maximization (CAIM) algorithm. Afterward, we expose some modifications on the different key components of the limited memory implementation of the WarpingLCSS. Finally, we review some fusion methods based on WarpingLCSS to tackle the multi-class gesture problem and recognition conflicts.

2.1. Constrained Many-Objective Optimization

Since artificial intelligence and engineering applications tend to involve more than two and three objective criteria [40], the concept of many objective optimization problems must be introduced beforehand. Literally, they involve many objectives in a conflicted and simultaneous manner. Hence, a constrained many-objective optimization problem may be formulated as follows:

$$\begin{aligned}
 & \text{minimize} && \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_m(\mathbf{x})]^T \\
 & \text{subject to} && g_j(\mathbf{x}) > 0, \quad j = 1, \dots, J \\
 & && h_k(\mathbf{x}) = 0, \quad k = 1, \dots, K \\
 & && \mathbf{x} \in \Omega
 \end{aligned} \tag{1}$$

where $\mathbf{x} = [x_1, \dots, x_n]^T$ is a n -decision variable candidate solution taking its value in the bonded space Ω . A solution respecting the J inequality ($g_j(\mathbf{x}) > 0$) and K equality constraints ($h_k(\mathbf{x}) = 0$) is qualified as attainable. These constraints are included in the objective functions and are detailed in our proposed method in Section 3.3. $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ associates a candidate solution to the objective space \mathbb{R}^m through m conflicting objective functions. The obtained results are thus alternative solutions but have to be considered equivalent since no information is given regarding the relevance of the others.

A solution \mathbf{x}^1 is said to dominate another solution \mathbf{x}^2 , written as $\mathbf{x}^1 \prec \mathbf{x}^2$ if and only if

$$\begin{aligned}
 & \forall i \in \{1, \dots, m\} : f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \\
 & \bigwedge \exists j \in \{1, \dots, m\} : f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)
 \end{aligned} \tag{2}$$

2.2. C-MOEA/DD

MOEA/DD is an evolutionary algorithm for many-objective optimization problems, drawing its strength from MOEA/D [44] and NSGA-III [45]. As it combines both the dominance-based and decomposition-based approaches, it implies an effective balance between the convergence and diversity of the evolutionary process. Decomposition is a popular method to break down a multiple objective problem into a set of scalar optimization subproblems. Here, the authors use the penalty-based boundary intersection approach,

but they highlight that any approach could be applied. Subsequently, we briefly explain the general framework of MOEA/DD and expose its requisite modifications for solving constrained many-objective optimization problems.

At first, a procedure generates N solutions to form the initial parent solutions and creates a weight vector set, W , representing N unique subregions in the objective space. As the current problem does not exceed six objectives, only the one layer weight generation algorithm was used. The T closest weights for each solution are also extracted to form a neighborhood set of weight vectors, E . The initial population, P , is then divided into several non-domination levels using the fast non-dominated sorting method employed in NSGA-II.

In the MOEA/DD main while-loop, a common process is applied for each weight vector in E until the termination criterion is reached. It consists of randomly choosing k -mating parents in the neighboring subregions of the weight vector considered. When no solution exists in the selected subregions, they are randomly selected in the current population. These k -solutions are then altered using genetic operators. For each offspring, an intricate update mechanism is applied on the population.

First, the associated subregion of the offspring is identified. The considered offspring is then merged with the population in a temporary container, P' . Next, the non-domination level structure of P' is updated. It is worthy to note that an ingenious method was employed to avoid full non-dominated sorting of P' . Since the population must preserve its size throughout the run of MOEA/DD, three cases may arise. When all solutions are non-dominated, the worst solution of the most crowded weight vector is deleted from the population. This function has been denominated *LocateWorst*. When there are multiple non-domination levels, the deletion of one solution depends on the number within the last non-domination level, F_l . On the one hand, there is only one solution in F_l , and the density of the associated subregion is investigated so as not to incorrectly alter the population diversity. *LocateWorst* is called in the case where the density contains only one element. When the most crowded subregion associated with each solution in F_l contains more than one element, the solution owning the largest scalarized value within it is deleted. Otherwise, *LocateWorst* is called so as not to delete isolated subregions.

Since MOEA/DD is designed to solve unconstrained many-objective optimization problems, Li et al. [40] also provided an extension for handling constrained many-objective optimization problems, which requires three modifications. First, a constraint violation value, $CV(x)$, henceforth accompanies each solution x . It is determined as follows:

$$CV(x) = \sum_{j=1}^J \langle g_j(x) \rangle + \sum_{k=1}^K |h_k(x)| \quad (3)$$

where the function $\langle \alpha \rangle$ returns the absolute value of α if $\alpha < 0$ and returns 0 otherwise. Second, while the abovementioned update procedure is maintained for feasible solutions, the survey of the infeasible ones is dictated by their association with an isolated subregion. More precisely, a second chance of survival is granted to these infeasible solutions, and the solution with the largest CV or the one that is not associated with an isolated subregion is eliminated from the next population. Finally, the selection for reproduction procedure becomes a binary tournament, where two solutions are initially randomly picked, and the solution with the smallest CV is favoured or a random choice is applied in the case of equality.

2.3. Discretization

The discretization process aims to transform a set of continuous attributes into discrete ones. Although there is a substantial number of discretization methods in the literature, Garcia et al. [26] recently carried out extensive testing of the 30 most representative and newest discretization techniques in supervised classification. Amongst the best performing algorithms, FUSINTER, ChiMerge, CAIM, and Modified Chi2 obtained the highest average

accuracies; it is possible to add Zeta and MDLP to this list if the Cohen's kappa metric is considered. In the authors' taxonomy, the evaluation measures for comparing solutions were broken down into five families: information, statistics, rough set, wrapper, and binning. Subsequently, we review few evolutionary approaches to solve discretization problems and succeeding methods of CAIM.

In [46], a supervised method called Evolutionary Cut Points Selection for Discretization (ECPD) was introduced. The technique exploits the fact that boundary points are suitable candidates for partitioning numerical attributes. Hence, a complete set of boundary points for each attribute is first generated. A CHC model [47] then searches the optimal subset of cut points while minimizing the inconsistency. Later on, the evolutionary multivariate discretizer (EMD) was proposed on the same basis [27]. The inconsistency was substituted for the aggregate classification error of an unpruned version of C4.5 and a Naive Bayes. Additionally, a chromosome length reduction algorithm was added to overcome large numbers of attributes and instances in datasets. However, the selection of the most appropriate discretization scheme relies on the weighted-sum of each objective functions, where a user-defined parameter is provided. This approach is thus limited even though varying parameters of a parametric scalarizing approach may produce multiple different Pareto-optimal solutions. In [25], a multivariate evolutionary multi-objective discretization (MEMOD) algorithm is proposed. It is an enhanced version of EMD, where the CHC has been replaced by the well-known NSGA-II, and the chromosome length reduction algorithm hereafter exploits all Pareto solutions instead of the best one. The following objective functions have been considered: the number of cut points currently selected, the average classification error produced by a CART and Naive Bayes, and the frequency of the selected cut points.

As previously exposed, CAIM stands out due to its performance amongst the classical techniques. Some extensions have been proposed, such as Class-Attribute Contingency Coefficient [48], Autonomous Discretization Algorithm (Ameva) [49], and ur-CAIM [30]. Ameva has been successfully applied in activity recognition [50] and fall detection for people who are older [51]. The technique is designed for achieving a lower number of discretization intervals without prior user specifications and maximizes a contingency coefficient based on the χ^2 statistics. The Ameva criterion is formulated as follows:

$$Ameva(k) = \frac{\chi^2}{k(l-1)} \quad (4)$$

where k and l are the number of discrete intervals and the number of classes, respectively. The ur-CAIM discretization algorithm enhances CAIM for both balanced and imbalanced classification problems. It combines three class-attribute interdependence criteria in the following manner:

$$ur-CAIM = CAIM_N * CAIR * (1 - CAIU) \quad (5)$$

where $CAIM_N$ denotes the CAIM criterion scaled into the range [0,1]. CAIR and CAIU stand for Class-Attribute Interdependence Redundancy and Class-Attribute Interdependence Uncertainty, respectively. In the ur-CAIM criterion, the CAIR factor has been adapted to handle unbalanced data.

2.4. Limited-Memory Warping LCSS Gesture Recognition Method

SegmentedLCSS and WarpingLCSS, introduced by [18], are two template matching methods for online gesture recognition using wearable motion sensors based on the longest common subsequence (LCS) algorithm. Aside from being robust against human gesture variability and noisy gathered data, they are also tolerant to noisy labeled annotations. On three datasets (10–17 classes), both methods outperform DTW-based classifiers with and without the presence of noisy annotations. WarpingLCSS has a smaller runtime complexity, about one order of magnitude, than SegmentedLCSS. In return, a penalty parameter, which

is application-specific, has to be set. Since each method is a binary classifier, a fusion method must be established, which will be discussed and illustrated in detail later.

A recently proposed variant of the WarpingLCSS method [21], labeled LM-WLCSS, allows the technique to run on a resource constrained sensor node. A custom 8-bit Atmel AVR motion sensor node and a 32-bit ARM Cortex M4 microcontroller were successfully used to illustrate the implementation of this method on three different everyday life applications. On the assumption that a gesture may last up to 10 s and given that the sample rate is 10 Hz, the chips are capable of recognizing, simultaneously and in real-time, 67 and 140 gestures, respectively. Furthermore, the extremely low power consumption used to recognize one gesture (135 μ W) might suggest an ASIC (Application-Specific Integrated Circuit) implementation.

In the following subsections, we review the core components of the training and recognition processes of an LM-WLCSS classifier, which will be in charge of recognizing a particular gesture. All streams of sensor data acquired using multiple sensors attached to the sensor node are pre-processed using a specific quantization step to convert each sample into a sequence of symbols. Accordingly, these strings allow for the formation of a training data set essential for selecting a proper template and computing a rejection threshold. In the recognition mode, each new sample gathered is quantized and transmitted to the LM-WLCSS and then to a local maximum search module, called SearchMax, to finally output if a gesture has occurred or not. Figure 1 describes the entire data processing flow.

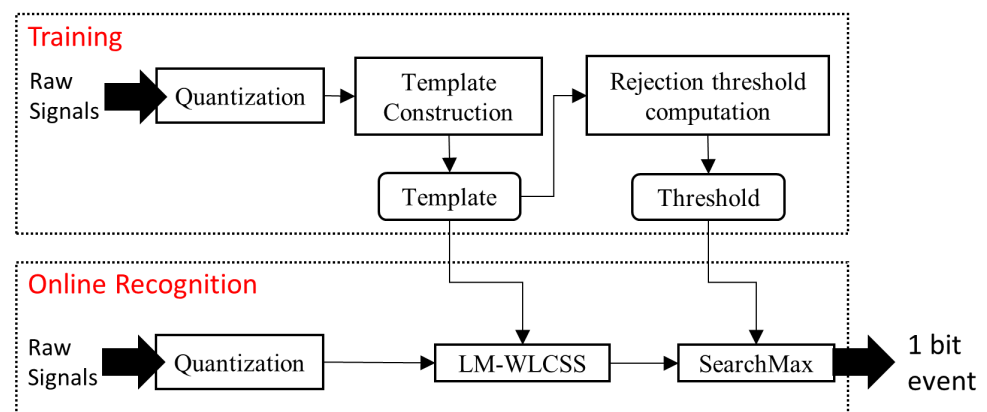


Figure 1. A binary classifier based on the Limited-Memory Warping LCSS [21].

2.4.1. Quantization Step (Training Phase)

At each time, t , a quantization step assigns an n -dimensional vector,

$$x(t) = [x_1(t) \dots x_n(t)], \quad (6)$$

representing one sample from all connected sensors as a symbol. In other words, a prior data discretization technique is applied on the training data, and the resulting discretization scheme is used as the basis of a data association process for all incoming new samples. Specifically to the LM-WLCSS, Roggen et al. [21] applied the K-means algorithm and the nearest neighbor. Despite the fact that K-means is widely employed, it suffers from the following disadvantages: the algorithm does not guaranty the optimality of the solution (position of cluster centers) and the optimal number of clusters assessed must be considered the optimum. In this paper, we investigate the use of the Ameva and ur-CAIM coefficients as a discretization evaluation measure in order to find the best suitable discretization

scheme. The nearest neighbor algorithm is preserved, where the squared Euclidean distance was selected as a distance function. More formally, a quantization step is defined as follows:

$$Q_c(x(t)) = \underset{i=1, \dots, |\mathcal{L}_c|}{\operatorname{argmin}} \frac{\|x(t) - L_{ci}\|^2}{\max_{j,k=1, \dots, |\mathcal{L}_c|} \|L_{cj} - L_{ck}\|^2} \tag{7}$$

where $Q_c(\cdot)$ assigns to the sample $x(t)$ the index of a discretization point L_{ci} chosen from the discretization scheme \mathcal{L}_c associated with the gesture class c . Therefore, the stream is converted into a succession of discretization points.

2.4.2. Template Construction (Training Phase)

Let s_{ci} denote the sequence i , i.e., the quantized gesture instance i , belonging to the gesture class training data set S_c . Hence, $S_c \subset S$, where S is the training data set. In the LM-WLCSS, the template construction of a gesture class c simply consists of choosing the first motif instance in the gesture class training data set. Here, we adopt the existing template construction phase of the WarpingLCSS. A template \bar{s}_c , representing all gestures from the class c , is therefore the sequence that has the highest LCS among all other sequences of the same class. It results in the following:

$$\bar{s}_c = \underset{s_{ci} \in S_c}{\operatorname{argmax}} \sum_{j \in |S_c|, j \neq i} l(s_{ci}, s_{cj}) \tag{8}$$

where $l(\cdot, \cdot)$ is the length of the longest common subsequence.

The LCS problem has been extensively studied, and it has an exponential raw complexity of $\mathcal{O}(2^n)$. A major improvement, proposed in [52], is achieved by dynamic programming in a runtime of $\mathcal{O}(nm)$, where n and m are the lengths of the two compared strings. In [43], the authors suggested three new algorithms that improve the work of [53], using a van Emde Boas tree, a balanced binary search tree, or an ordered vector. In this paper, we use the ordered vector approach, since its time and space complexities are $\mathcal{O}(nL)$ and $\mathcal{O}(R)$, where n and L are the lengths of the two input sequences and R is the number of matched pairs of the two input sequences.

2.4.3. Limited-Memory Warping LCSS

LM-WLCSS instantaneously produces a matching score between a symbol $s_c(i)$ and a template \bar{s}_c . When one identical symbol encounters the template \bar{s}_c , i.e., the i th sample and the first j th sample of the template are alike, a reward R_c is given. Otherwise, the current score is equal to the maximum between the two following cases: (1) a mismatch between the stream and the template, and (2) a repetition in the stream or even in the template. An identical penalty D , the normalized squared Euclidean distance between the two considered symbols $d(\cdot, \cdot)$ weighted by a fixed penalty P_c , is thus applied. Distances are retrieved from the quantizer since a pairwise distance matrix between all symbols in the discretization scheme has already been built and normalized. In the original LM-WLCSS, the decision between the different cases is controlled by tolerance ϵ . Here, this behavior has been nullified due to the exploration capacity of the metaheuristic to find an adequate discretization scheme. Hence, modeled on the dynamic computation of the LCS score, the matching score $M_c(j, i)$ between the first j symbols of the template \bar{s}_c and the first i symbols of the stream W stem from the following formula:

$$M_c(j, i) = \begin{cases} 0, & \text{if } i = 0 \text{ or } j = 0 \\ M_c(j - 1, i - 1) + R_c, & \text{if } W(i) = \bar{s}_c(j) \\ \max \begin{cases} M_c(j - 1, i - 1) - D, \\ M_c(j - 1, i) - D, \\ M_c(j, i - 1) - D, \end{cases} & \text{otherwise} \end{cases} \tag{9}$$

where $D = P_c * d(W(i), \bar{s}_c(j))$. It is easily determined that the higher the score, the more similar the pre-processed signal is to the motif. Once the score reaches a given acceptance threshold, an entire motif has been found in the data stream. By updating a backtracking variable, B_c , with the different lines of (9) that were selected, the algorithm enables the retrieving of the start-time of the gesture.

2.4.4. Rejection Threshold (Training Phase)

The computation of the rejection threshold, ω_c , requires computing the LM-WLCSS scores between the template and each gesture instance (expected chosen template) contained in the gesture class c . Let $\mu^{(c)}$ and $\sigma^{(c)}$ denote the resulting mean and standard deviation of these scores. It follows

$$\omega_c = \mu^{(c)} - h_c * \sigma^{(c)}, \quad (10)$$

where h_c is a real positive in the range $]0, \frac{\mu^{(c)}}{\sigma^{(c)}}[$.

2.4.5. Searchmax (Recognition Phase)

A SearchMax function is called after every update of the matching score. It aims to find the peak in the matching score curve, representing the beginning of a motif, using a sliding window without the necessity of storing that window. More precisely, the algorithm first searches the ascent of the score by comparing its current and previous values. In this regard, a flag is set, a counter is reset, and the current score is stored in a variable called Max. For each following value that is below Max, the counter is incremented. When Max exceeds the pre-computed rejection threshold, ω_c , and the counter is greater than the size of a sliding window WF_c , a motif has been spotted. The original LM-WLCSS SearchMax algorithm has been kept in its entirety. WF_c , therefore, controls the latency of the gesture recognition and must be at least smaller than the gesture to be recognized.

2.4.6. Backtracking (Recognition Phase)

When a gesture has been spotted by SearchMax, retrieving its start-time is achieved using a backtracking variable. The original implementation as a circular buffer with a maximal capacity of $|\bar{s}_c| * WB_c$ has been maintained, where $|\bar{s}_c|$ and WB_c denote the length of the template \bar{s}_c and the length of the backtracking variable B_c , respectively. However, we add an additional behavior. More precisely, WF_c elements are skipped because of the required time for SearchMax to detect local maxima, and the backtracking algorithm is applied. The current matching score is then reset, and the WF_c previous samples' symbols are reprocessed. Since only references to the discretization scheme \mathcal{L}_c are stored, re-quantization is not needed.

2.5. Fusion Methods Using WarpingLCSS

WarpingLCSS is a binary classifier that matches the current signal with a given template to recognize a specific gesture. When multiple WarpingLCSS are considered in tackling a multi-class gesture problem, recognition conflicts may arise. Multiple methods have been developed in literature to overcome this issue. Nguyen-Dinh et al. [18] introduced a decision-making module, where the highest normalized similarity between the candidate gesture and each conflicting class template is outputted. This module has also been exploited for the SegmentedLCSS and LM-WLCSS. However, storing the candidate detected gesture and reprocessing as many LCSS as there are gesture classes might be difficult to integrate on a resource constrained node. Alternatively, Nguyen-Dinh et al. [19] proposed two multimodal frameworks to fuse data sources at the signal and decision levels, respectively. The signal fusion combines (summation) all data streams into a single dimension data stream. However, considering all sensors with an equal importance might not give the best configuration for a fusion method. The classifier fusion framework aggregates the similarity scores from all connected template matching modules, and each

one processes the data stream from one unique sensor, into a single fusion spotting matrix through a linear combination, based on the confidence of each template matching module. When a gesture belongs to multiple classes, a decision-making module resolves the conflict by outputting the class with the highest similarity score. The behavior of interleaved spotted activities is, however, not well-documented. In this paper, we decided to deliberate on the final decision using a light-weight classifier.

3. Proposed Method

In this section, we present an evolutionary algorithm for feature selection, discretization, and parameter tuning for an LM-WLCSS-based method. Unlike many discretization techniques requiring a prefixed number of discretization points, the proposed algorithm exploits a variable-length structure in order to find the most suitable discretization scheme for recognizing a gesture using LM-WLCSS. In the remaining part of this paper, our method is denoted by MOFSD-GR (Many-Objective Feature Selection and Discretization for Gesture Recognition).

3.1. Solution Encoding and Population Initialization

A candidate solution x integrates all key parameters required to enable data reduction and to recognize a particular gesture using the LM-WLCSS method.

As previously noted, the sample at time t is an n -dimensional vector $x(t) = [x_1(t) \dots x_n(t)]$, where n is the total number of features characterizing the sample. Focusing on a small subset of features could significantly reduce the number of required sensors for gesture recognition, save computational resources, and lessen the costs. Feature selection has been encoded as a binary valued vector $p_c = \{p_j\}_{j=1}^n \in [0, 1]^n$, where $p_j = 0$ indicates that the corresponding features is not retained whereas $p_j = 1$ signifies that the associated feature is selected. This type of representation is very widespread across literature.

The discretization scheme $\mathcal{L}_c = (L_1, L_2, \dots, L_m)$ is represented by a variable-length vector, where m is a positive integer uniformly chosen in the range $[K_c^{lower}, K_c^{upper}] = [10, 70]$. The upper limit of this decision variable is purposely larger than necessary to improve diversity. These limits are selected by trial and error. Each discretization point $L_i = (z_1, z_2, \dots, z_n) \in [0, 1]^n$, $i \in \{1, \dots, m\}$, is a n -dimensional point uniformly chosen in the training space of the gesture c .

Amongst the abovementioned LM-WLCSS parameters, only the SearchMax window length WF_c , the penalty P_c , and the coefficient h_c of the threshold have been included into the solution representation.

1. WF_c controls the latency of the recognition process, i.e., the required time to announce that a gesture peak is present in the matching score. WF_c is a positive integer uniformly chosen in the interval $[WF_c^{lower}, WF_c^{upper}] = [5, 15]$. By fixing the reward R_c to 1, the penalty P_c is a real number uniformly chosen in the range $[0, 1]$; otherwise, gestures that are different from the selected template would be hardly recognizable.
2. The coefficient h_c of the threshold is strongly correlated to the reward R_c and the discretization scheme \mathcal{L}_c . Since it cannot easily be bounded, its value is locally investigated for each solution.
3. The backtracking variable length WB_c allows us to retrieve the start-time of a gesture. Although a too short length results in a decrease in recognition performance of the classifier, its choice could reduce the runtime and memory usage on a constrained sensor node. Since its length is not a major performance limiter in the learning process and it can easily be rectified by the decider during the deployment of the system, it was fixed to three times the length of the longest gesture occurrence in c in order to reduce the complexity of the search space.

Hence, the decision vector x can be formulated as follows:

$$x = (p_c, \mathcal{L}_c, P_c, WF_c, h_c). \quad (11)$$

3.2. Operators

In C-MOEA/DD, selected solutions produce one or more offspring using any genetic operators. In this paper, for each selected parent solution pair $\{x_1, x_2\}$, a crossover generates two children $\{x'_1, x'_2\}$ that are mutated afterwards. In the following subsections, these two operators are explained.

3.2.1. Crossover Operation

The classical uniform crossover is used for the selected feature vector. In this paper, we adapted the recently proposed rand-length crossover for the random variable-length crossover differential evolution algorithm [42] to crossover two discretization schemes. More precisely, offspring lengths are firstly randomly and uniformly selected from the range $[K_c^{lower}, \min(|x_1^{\mathcal{L}_c}| + |x_2^{\mathcal{L}_c}|, K_c^{upper})]$, where $x_i^{\mathcal{L}_c}$ indicates the discretization scheme (to be used for the gesture class c) associated with the solution x_i and $|\cdot|$ indicates the number of elements in this designated discretization scheme. For the current value of $i \in [1, \min_{i \in \{1,2\}} |x_i^{\mathcal{L}_c}|]$, three cases might occur. When both parent solutions contain a discretization point at the index i , the simulated binary crossover (SBX) is applied to each dimension of the two points. When one of the parent solution discretization scheme is too short, both children inherit from the parent having the longest discretization scheme. Otherwise, a new discretization point is uniformly chosen in the training space for each children solution. All newly created discretization points are randomly assigned to children solution. The pseudo-code of the rand-length crossover for discretization scheme procedure is given in Algorithm 1.

Since LM-WLCSS penalties are encoded as real-values, the SBX operator is also applied to the decision variable P_c . In contrast, SearchMax window lengths are integers; thus, we incorporate the weighted average normally distributed arithmetic crossover (NADX) [54]. It induces a greater diversity than uniform crossover and SBX operators while still proposing values near and between the parents. Despite the length of the backtracking variable having been fixed, the NADX operator could be considered.

When selecting features, the discretization schemes or LM-WLCSS penalties, and SearchMax window lengths of children solutions are different from those of parent solutions, and their coefficients, h_c , of the threshold must be undefined because the resulting LM-WLCSS classifier from the solution is altered.

3.2.2. Mutation Operation

All decision variables are equiprobably modified. The uniform bit flip mutation operator is applied to the selected feature binary vector. Each discretization point in the discretization scheme is also equiprobably altered. Specifically, when a discretization point has been identified for a modification, all of its features are mutated using the polynomial mutation operator. For all of the remaining decision variables, the polynomial mutation is applied whether decision variables are encoded as integers or real numbers.

Algorithm 1: Rand-length crossover for discretization schemes.

Input: discretization schemes $\{\mathcal{L}_c^1, \mathcal{L}_c^2\}$ of two parent solutions $\{x_1, x_2\}$
Output: discretization schemes $\{\mathcal{L}_c^{1'}, \mathcal{L}_c^{2'}\}$ for two offspring solutions $\{x_1', x_2'\}$

- 1 $N_{off1} \leftarrow \text{random}(K_c^{lower}, \min(|\mathcal{L}_c^1| + |\mathcal{L}_c^2|, K_c^{upper}))$
- 2 $N_{off2} \leftarrow \text{random}(K_c^{lower}, \min(|\mathcal{L}_c^1| + |\mathcal{L}_c^2|, K_c^{upper}))$
- 3 **for** $i=1$ **to** $\max(N_{off1}, N_{off2})$ **do**
- 4 Sample c_1, c_2
- 5 **if** $i > |\mathcal{L}_c^1|$ **then**
- 6 **if** $i \leq |\mathcal{L}_c^2|$ **then**
- 7 $c_1 \leftarrow c_2 \leftarrow L_{ci}^2$
- 8 **else**
- 9 **for** $j=1$ **to** n **do**
- 10 $c_1(j) \leftarrow$ random point in the training space of the gesture c
- 11 $c_2(j) \leftarrow$ random point in the training space of the gesture c
- 12 **end**
- 13 **end**
- 14 **else**
- 15 **if** $i > |\mathcal{L}_c^2|$ **then**
- 16 $c_1 \leftarrow c_2 \leftarrow L_{ci}^1$
- 17 **else**
- 18 **for** $j=1$ **to** n **do**
- 19 $\{c_1(j), c_2(j)\} \leftarrow \text{SBX}(L_{ci}^1(j), L_{ci}^2(j))$
- 20 **end**
- 21 **end**
- 22 **end**
- 23 $u \leftarrow \text{random}(0, 1)$
- 24 **if** $u \leq 0.5$ **then**
- 25 **if** $i < N_{off1}$ **then** $L_{ci}^{1'} \leftarrow c_1$
- 26 **if** $i < N_{off2}$ **then** $L_{ci}^{2'} \leftarrow c_2$
- 27 **else**
- 28 **if** $i < N_{off1}$ **then** $L_{ci}^{1'} \leftarrow c_2$
- 29 **if** $i < N_{off2}$ **then** $L_{ci}^{2'} \leftarrow c_1$
- 30 **end**
- 31 **end**
- 32 **return** $\{\mathcal{L}_c^{1'}, \mathcal{L}_c^{2'}\}$

3.3. Objective Functions

The quality of a candidate solution is measured by the objective functions. In order to find the best solution for recognizing a particular gesture using LM-WLCSS, five functions have been considered:

$$\text{minimize } F(x) = [-f_1(x), -f_2(x), -f_3(x), -f_4(x), f_5(x)]^T \quad (12)$$

where

$$f_1(x) = F1score = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (13)$$

$$f_2(x) = \frac{1}{|\bar{s}_c| |S_c|} \sum_{y \in S_c, y \neq \bar{s}_c} l(\bar{s}_c, y) \quad (14)$$

$$f_3(x) = Aमेवा(\mathcal{L}_c) \quad (15)$$

$$f_4(x) = - \sum_{e \in T_c} \frac{p(e) \log(p(e))}{\log(|T_c|)} \quad (16)$$

$$f_5(x) = \frac{\sum_{y \in p_c} [y = 1]}{n} \quad (17)$$

subject to

$$|T_c| \geq 3 \quad (18)$$

$$\omega_c \geq 0 \quad (19)$$

where T_c is the set of distinct discretization points in the elected template \bar{s}_c , $|T_c|$ is the number of distinct elements in the latter, and $[\cdot]$ denotes the Iverson bracket.

Let us firstly define the basic terms generated by a confusion matrix: tp (true positives) is the number of correctly identified samples, fp (false positives) refers to the incorrectly identified samples, tn (true negatives) is the number of correctly rejected samples, and fn (false negatives) refers to the incorrectly rejected samples. In (13), f_1 measures how well the trained binary classifier performs on the testing data set. Although the accuracy is widely acknowledged, it cannot be used as exclusive performance recognition indicator, since the classifier could have exactly zero predictive power [55]. We alternatively selected the F1 score, defined as the harmonic mean of precision and recall, where $\text{precision} = \frac{tp}{tp+fp}$ and $\text{recall} = \frac{tp}{tp+fn}$.

The objective function f_2 , in (14), directly comes from the template construction during the training phase of the binary classifier. It is the average sum of the longest common subsequence between the elected template \bar{s}_c and the other quantized gesture instances in the gesture class training data set. The higher the score is, the more the template represents the gesture class c .

The Ameva criterion, determined by the objective function f_3 in (15), expresses the quality of the discretization scheme component of the solution. Its highest values are attained when all samples from a specific class are quantized to a unique discretization point (the other discretization points have no associated samples). Additionally, the criterion favours a low number of discretization points. Since there are only two classes in this problem, i.e., the samples from the gesture class c represents the positive class, and all others examples are negatives; it might be possible to encounter similarities in the different gesture executions for both classes. As a result, negative examples might be quantized into the same discretization points defining the class template \bar{s}_c , and the Ameva criterion might try to create unnecessary discretization points. To overcome this issue, a constraint on the template, defined in (18), imposes that the latter must be defined by at least three distinct discretization points. Additionally, in (16), the objective function f_4 counters this conflicting situation and measures heterogeneity by the normalized entropy of the elected template \bar{s}_c included between $[0, 1]$. Lower appearance of a discretization point in the template is thus penalized. The Ameva criterion may be interchanged with ur-CAIM or any other discretization criterion.

In (17), the last objective function indicates the average number of selected features in the current solution, as we need to reduce the number of features.

Algorithm 2 presents the pseudo-code of the evaluation procedure of a candidate solution x . First and foremost, a quantizer Q_c is created using the discretization scheme \mathcal{L}_c and the feature selection vector p_c . An LM-WLCSS classifier can thus be trained on the training dataset. Although the objective function f_5 is completely independent of the classifier construction, an infeasible solution situation may be encountered due to the negativity of the rejection threshold ω_c , as stated in (19). In contrast, evaluation procedure continues, and from the elected class template T_c and the rejection threshold, it follows the objective function f_3 . As previously mentioned, the decision variable h_c must be locally investigated. When the coefficient of variation $\frac{\mu^{(c)}}{\sigma^{(c)}}$ is different from zero, the procedure increments the value of h_c from 0 to $\frac{\mu^{(c)}}{2 \times \sigma^{(c)}}$ with a step of $\frac{\mu^{(c)}}{2 \times 10 \times \sigma^{(c)}}$ because a high amplitude of the coefficients can nullify the rejection threshold. For each coefficient value, the previously constructed LM-WLCSS classifier is not retained. Only updating the SearchMax threshold, clearing the circular buffer (variable B_c), and resetting the matching score are necessary. Here, the greater objective function f_1 obtained value (i.e., the best-obtained classifier performance) and its associated h_c are preserved, and the evaluated solution x and objective function $F(x)$ are updated in consequence.

3.4. Multi-Class Gesture Recognition System

Whenever a new sample $x(t)$ is acquired, each of the required subset of the vector is transmitted to the corresponding trained LM-WLCSS classifier to be specifically quantized and instantaneously classified. Each binary decision, forming a decision vector $d(t)$, is sent to a decision fusion module to eventually yield which gesture has been executed. Among all of the aggregation schemes for binarization techniques, we decided to deliberate on the final decision through a light-weight classifier, such as neural networks, decision trees, logistic regressions, etc. Figure 2 illustrates the final recognition flow.

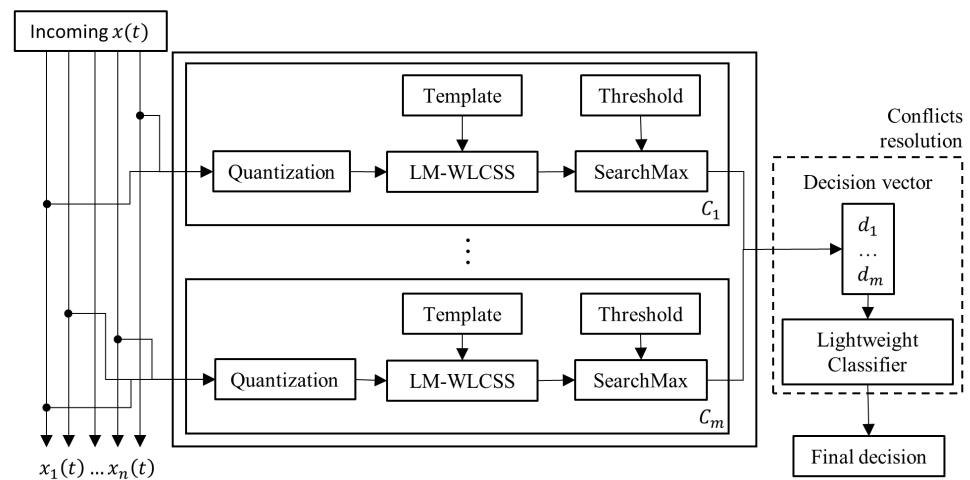


Figure 2. A multiclass gesture recognition system including multiple binary classifiers based on LM-WLCSS.

Algorithm 2: Solution evaluation.

Input: solution x
Output: solution $F(x)$

- 1 Create a quantizer Q_c using the discretization scheme \mathcal{L}_c and the feature selection vector p_c
- 2 **if** $\omega_c \geq 0$ or $|T_c| \geq 3$ **then**
- 3 $F(x) \leftarrow [0, 0, 0, 0, \infty]$
- 4 **return** $F(x)$
- 5 **end**
- 6 Compute $f_3(x)$ and $f_5(x)$
- 7 Train a LM-WLCSS classifier using Q_c
- 8 Compute $f_2(x)$ and $f_4(x)$
- 9 **if** $\frac{\mu^{(c)}}{\sigma^{(c)}} = 0$ **then**
- 10 $h_c \leftarrow 0$
- 11 Compute $f_1(x)$
- 12 **else**
- 13 $hmax \leftarrow 0$
- 14 $f_1max \leftarrow 0$
- 15 **repeat**
- 16 Update the SearchMax threshold $\omega_c \leftarrow \mu^{(c)} - h_c * \sigma^{(c)}$
- 17 Clear the backtracking variable B_c and reset the matching score $M_c(j, 0) \leftarrow 0$, where $j = 1, \dots, |\bar{s}_c|$
- 18 $f_1 \leftarrow$ Compute $f_1(x)$
- 19 **if** $f_1 > f_1max$ **then**
- 20 $f_1max \leftarrow f_1$
- 21 $hmax \leftarrow h_c$
- 22 **end**
- 23 $h_c \leftarrow h_c + \frac{\mu^{(c)}}{2 \times 10 \times \sigma^{(c)}}$
- 24 **until** $h \geq \frac{\mu^{(c)}}{2 \times \sigma^{(c)}}$
- 25 $h_c \leftarrow hmax$
- 26 $f_1(x) \leftarrow f_1max$
- 27 **end**
- 28 $F(x) \leftarrow [-f_1(x), -f_2(x), -f_3(x), -f_4(x), f_5(x)]$
- 29 **return** $F(x)$

4. Experiments

In this section, we describe the experimental framework. First, we present the Opportunity dataset [56] as a benchmark for gesture recognition and dimensionality reduction. This dataset, available on the UCI machine learning repository (<https://archive.ics.uci.edu/ml/datasets/opportunity+activity+recognition> (accessed on 15 September 2021)), aims to propose a benchmark for human activity recognition algorithms or for specific stages of the activity recognition chain, such as dimensionality reduction, signal fusion, and classification. It includes multiple runs of a scripted two-part scenario performed by several subjects equipped with on-body sensors in a simulated studio flat, wherein numerous ambient and object sensors have been integrated. All raw sensor readings have 243 dimensions. The first part consists of an activity of daily living, allowing for a look at four abstraction levels of the activity recognition. The second one, denominated ‘drill run’, focuses on the number of instance daily gestures.

4.1. Benchmark Dataset

The different approaches used in the literature to report classification results on this particular benchmark are reviewed. Finally, we detail the key points of our experimental

setup, such as the required dataset partitioning imposed by our approach to avoid biases, general parameter settings, and performance metrics.

4.2. Experimental Setup

Three main ways have been adopted by gesture recognition literature to report classification results on the Opportunity dataset. First, in [57,58], the proposed method was tested on the challenging task B2 [58], where performance recognition must be reported on the testing set composed of ADL4 and ADL5 for Subjects 2 and 3. According to the challenge, the authors are free to include any remaining subsets into the training set. Missing values, due to packet-loss, have been replaced by linear interpolation. All on-body sensors have been exploited, resulting in an input space with 113 dimensions. Secondly, [58] also reported gesture recognition performances for each of the four subjects using an identical data preparation provided by the UCI repository. Although datasets have 113 dimensions, the methods used for handling missing data may reduce this number. Chen et al. [59] conducted a similar experimentation, but all types of sensors were included, i.e., 243 dimensions. Finally, in [18], a five-fold cross validation (in K -fold cross validation), a dataset \mathcal{D} is split into k mutually exclusive subsets, where the size of each fold is approximately equal. One of the partition \mathcal{D}_t , with $t \in \{1, 2, \dots, k\}$, is used for testing the classifier performance, and the remaining of the dataset, i.e., $\mathcal{D} \setminus \mathcal{D}_t$, consists of its training dataset. This process has to be repeated k -times and was performed on the 'drill run' subset of the Opportunity dataset using accelerometers on arms. Based on the same model validation technique, [19] evaluated the proposed methods on the 'drill run' of each subject using a five-fold cross validation. The experiments only employed 17 3D-sensors, and raw signals were down-sampled. In this work and the aforementioned one, there is no mention of methods for handling missing data.

In our proposed method, the whole training data stream must be quantized for each solution since the selected dimensions and discretization scheme vary. Due to the humongous Euclidean distance searches induced and limited experiment time requirements, we favour smaller datasets. Hence, for the sake of comparison, we reproduced the experiments of Nguyen-Dinh et al. [19] but without down-sampling raw signals. All 51 dimensions were scaled to unit size. We used the default method for handling missing values provided by the UCI repository. For each subject, Table 1 summarizes the number of repetitions (#inst) per gesture and their average length (avg) with standard deviation (SD). It follows that gestures have strong variability, especially 'CleanTable', 'DrinkfromCup', and 'ToggleSwitch', and the number of instances is inconstant. Additionally, this input dataset noticeably contains a very large portion of 'null classes' (40%).

Table 1. Number of instances and average gesture lengths per subject in the Gesture set of the Opportunity dataset.

Gesture Names	Subject 1			Subject 2			Subject 3			Subject 4		
	#inst	avg	SD	#inst	avg	SD	#inst	avg	SD	#inst	avg	SD
CleanTable	20	120.00	47.01	20	163.10	42.43	18	132.6	15.90	21	74.14	29.30
CloseDishwasher	20	86.85	11.03	19	89.05	11.44	18	85.67	7.86	21	59.57	15.15
CloseDoor1	21	102.95	9.55	20	110.35	9.31	18	126	8.64	21	85.14	10.43
CloseDoor2	20	101.70	20.54	20	121.05	10.47	18	135.8	7.43	21	83.00	9.17
CloseDrawer1	20	61.80	4.43	20	42.05	6.84	18	68.83	5.71	21	38.67	10.60
CloseDrawer2	20	63.35	5.05	20	43.60	7.60	18	75.44	7.40	21	43.86	9.38
CloseDrawer3	20	76.50	8.04	20	73.40	9.33	18	78.28	5.72	21	55.10	10.04
CloseFridge	20	76.25	5.84	20	73.20	7.57	19	84.79	13.37	21	56.00	12.94
DrinkfromCup	40	189.05	19.57	40	209.20	29.33	36	186.4	18.22	40	159.00	44.08
OpenDishwasher	20	89.75	5.70	21	97.19	14.03	18	90.33	7.34	21	65.81	12.05
OpenDoor1	20	91.75	11.09	20	101.55	14.72	18	130.6	10.86	21	79.81	10.94
OpenDoor2	20	103.10	5.66	20	101.10	18.01	18	145.2	14.64	21	77.24	11.53
OpenDrawer1	20	64.80	7.57	20	72.25	9.29	18	74.28	8.56	21	53.76	11.98
OpenDrawer2	20	68.75	5.46	20	56.30	8.32	18	76.56	5.80	21	47.57	12.34
OpenDrawer3	20	82.60	4.79	20	61.90	8.37	18	85.39	6.69	21	55.67	10.94
OpenFridge	20	75.50	6.43	20	82.50	11.28	19	100.2	11.19	21	57.71	6.69
ToggleSwitch	38	39.84	10.58	28	62.04	25.75	36	55.36	11.87	39	31.03	26.31

In this paper, we performed a five-fold cross-validation. The proposed framework for building a multi-class gesture recognition system based on LM-WLCSS, however, requires the partitioning of each training dataset, $\mathcal{Z} = \mathcal{D} \setminus \mathcal{D}_t$, into three mutually exclusive subsets, \mathcal{Z}_1 , \mathcal{Z}_2 , and \mathcal{Z}_3 , to avoid biased results. \mathcal{Z}_1 represents the training dataset used for all the base-level classifiers and contains 70% of \mathcal{Z} . The remaining data is equally split over \mathcal{Z}_1 and \mathcal{Z}_2 . Performance recognition is maximized over the test set \mathcal{Z}_2 . Once each binary classifier has been trained, predictions on the stream \mathcal{Z}_3 are obtained, transforming all incoming multi-modal samples into a succession of decision vectors. This newly created dataset, \mathcal{Z}_3' , allows us to resolve conflicts by training a light-weight classifier. Finally, the final performance of the system is assessed by using the testing dataset \mathcal{D}_t .

For our method, C-MOEA/DD parameters remain identical to the original paper [40]; hence, the penalty parameter in PBI $\theta = 5$, the neighborhood size $T = 20$, and the probability used to select in the neighborhood $\delta = 0.9$. For the reproduction procedure, the crossover probability is $p_c = 1.0$, and the distribution index for the SBX operators is $\eta_c = 30$. As stated before, mutation of a decision variable of a solution may occur with an equiprobability of occurrence $p_m = 1/6$, and when this decision variable is a vector, each element also has an equal probability to be altered. The polynomial mutation distribution index was fixed at $\eta_m = 20$. In this problem, we fixed the population size at 210, and the stopping criterion is reached when the number of evaluation exceeds 100,000.

4.3. Evaluation Metrics

The effectiveness of the proposed many-objective formulation is evaluated from the two following perspectives:

1. Effectiveness: Work based on WarpingLCSS and its derivatives mainly use the weighted F1-score F_w , and its variant $F_{w_{NoNull}}$, which excludes the *null class*, as primary evaluation metrics. F_w can be estimated as follows:

$$F_w = 2 * \sum_c \frac{N_c}{N_{total}} \frac{precision_c * recall_c}{precision_c + recall_c} \quad (20)$$

where N_c and N_{total} are, respectively, the number of samples contained in class c and the total number of samples. Additionally, we considered Cohen's kappa. This accuracy measure, standardized to lie on a -1 to 1 scale, compares an observed

accuracy Obs_{Acc} with an expected accuracy Exp_{Acc} , where 1 indicates the perfect agreement, and values below or equal to 0 represent poor agreement. It is computed as follows:

$$Kappa = \frac{Obs_{Acc} - Exp_{Acc}}{1 - Exp_{Acc}}. \quad (21)$$

2. Reduction capabilities: Similar to Ramirez-Gallego et al. [60], a reduction in dimensionality is assessed using a reduction rate. For feature selection, it designates the amount of reduction in the feature set size (in percentage). For discretization, it denotes the number of generated discretization points.

5. Results and Discussion

The validation of our simultaneous feature selection, discretization, and parameter tuning for LM-WLCSS classifiers is carried out in this section. The results on performance recognition and dimensionality reduction effectiveness are presented and discussed. The computational experiments were performed on an Intel Core i7-4770k processor (3.5 GHz, 8 MB cache), 32 GB of RAM, Windows 10. The algorithms were implemented in C++. The Euclidean and LCSS distance computations were sped up using Streaming SIMD Extensions and Advanced Vector Extensions. Subsequently, the Ameva or ur-CAIM criterion used as an objective function f_3 (15) is referred to as MOFSD-GR_{Ameva} and MOFSD-GR_{ur-CAIM} respectively.

On all four subjects of the Opportunity dataset, Table 2 shows a comparison between the best-provided results by Nguyen-Dinh et al. [19], using their proposed classifier fusion framework with a sensor unit, and the obtained classification performance of MOFSD-GR_{Ameva} and MOFSD-GR_{ur-CAIM}. Our methods consistently achieve better F_w and $F_{wNoNull}$ scores than the baseline. Although the use of Ameva brings an average improvement of 6.25%, the F1 scores on subjects 1 and 3 are close to the baseline. The current multi-class problem is decomposed using a one-vs.-all decomposition, i.e., there are m binary classifiers in charge of distinguishing one of the m classes of the problem. The learning datasets for the classifiers are thus imbalanced. As shown in Table 2, the choice of ur-CAIM corroborates the fact that this method is suitable for unbalanced dataset since it improves the average F1 scores by over 11%.

Table 2. Average recognition performances on the Opportunity dataset for the gesture recognition task, either with or without the null class.

	[19]		MOFSD-GR					
			Ameva			ur-CAIM		
	F_w	$F_{wNoNull}$	F_w	$F_{wNoNull}$	Kappa	F_w	$F_{wNoNull}$	Kappa
Subject 1	0.82	0.83	0.84	0.83	0.81	0.90	0.91	0.88
Subject 2	0.71	0.73	0.82	0.81	0.79	0.89	0.90	0.87
Subject 3	0.87	0.85	0.89	0.87	0.85	0.93	0.93	0.91
Subject 4	0.75	0.74	0.85	0.83	0.81	0.87	0.87	0.84

Figure 3 illustrates the feature reduction rates produced by MOFSD-GR_{Ameva} and MOFSD-GR_{ur-CAIM} across all 17 gestures of the Opportunity dataset. The following analysis are made.

1. The ur-CAIM criterion consistently leads to a better reduction rate (close to 80% in mean). Therefore, from a design point of view, the effectiveness of sensors—and their ideal placements—to recognize a specific activity are more identified.
2. The Ameva criterion achieves a more stable standard deviation in the reduction rate across all subjects than the ur-CAIM criterion.
3. Since MOFSD-GR_{Ameva} achieves a better recognition rate than the baseline, its implied reduction capabilities are still acceptable (>40%).

Figures 3 and 4 depict the number of discretization points yielded by the two discretization strategies across all 17 gestures of the Opportunity dataset. From the results, the following assessment can be made.

1. As intended by the nature of Ameva, MOFSD-GR_{Ameva} yields a small number of cut points close to the constraint imposing that the template be made of at least three distinct discretization points (18). However, this advantage seems to limit the exploration capacity of C-MOEA/DD since only half of the original features are discarded.
2. In contrast, MOFSD-GR_{ur-CAIM} tends to generate larger discretization schemes than MOFSD-GR_{Ameva}. Since the ur-CAIM criterion aggregates two conflicting objectives (CAIM aimed to generate a lower number of cut points, and the pair CAIR and CAIU advocates a larger number), compromises are made.

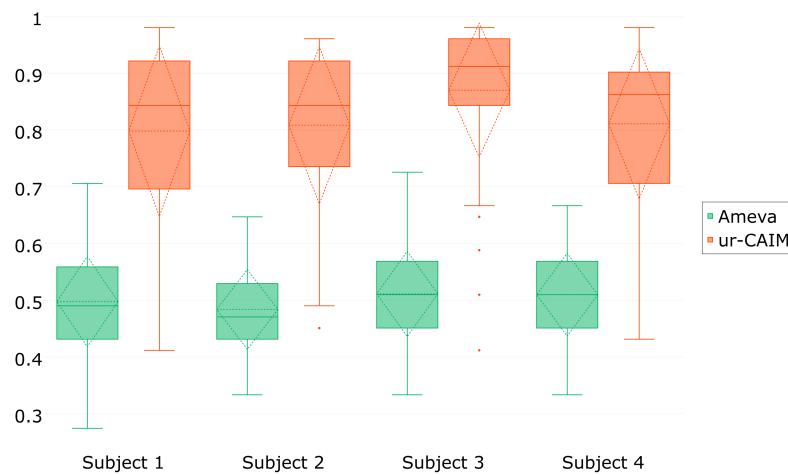


Figure 3. Box plot representation for feature selection (reduction rate in %).

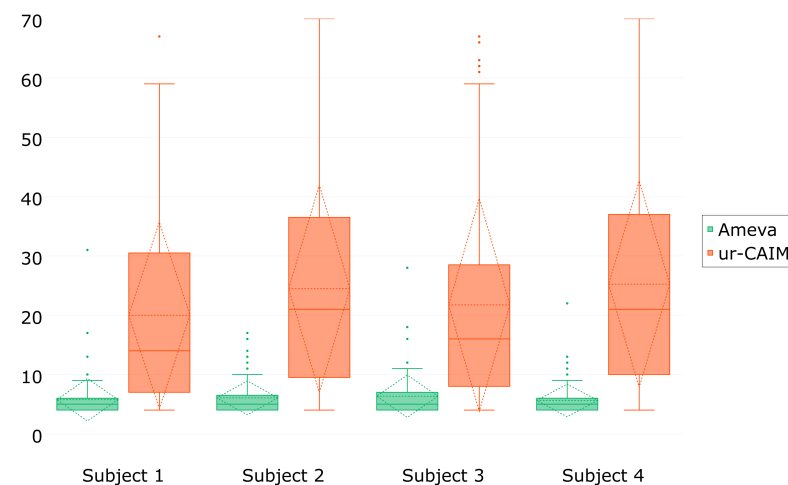


Figure 4. Box plot representation for discretization (number of cut points).

Tables 3 and 4 present more detailed results. They recapitulate the average, μ , and standard deviation, SD , of the number of cut points ($\#dp$) produced and features selected ($\#d$) by MOFSD-GR_{Ameva} and MOFSD-GR_{ur-CAIM}, respectively. Please note that no substantive conclusions could be drawn from the intersections between the following sets of selected features from (1) a particular subject, (2) a particular gesture, and (3) a particular gesture and fold due to the one-vs.-all decomposition approach used for this multi-class problem.

Table 3. Average cut points and selected features obtained by MOFSD-GR_{Ameva}.

Gesture Names	Subject 1				Subject 2				Subject 3				Subject 4			
	$\mu_{#d}$	$SD_{#d}$	$\mu_{#dp}$	$SD_{#dp}$	$\mu_{#d}$	$SD_{#d}$	$\mu_{#dp}$	$SD_{#dp}$	$\mu_{#d}$	$SD_{#d}$	$\mu_{#dp}$	$SD_{#dp}$	$\mu_{#d}$	$SD_{#d}$	$\mu_{#dp}$	$SD_{#dp}$
CleanTable	25.20	3.90	5.40	0.55	26.40	3.05	4.80	1.30	23.60	1.95	6.00	1.58	24.80	3.27	6.20	1.64
CloseDishwasher	27.00	6.67	5.20	1.79	24.60	5.08	4.60	0.89	21.60	5.13	5.20	1.64	22.20	3.56	5.80	1.30
CloseDoor1	22.60	7.50	5.60	2.07	27.00	1.22	4.80	1.30	24.20	4.49	6.00	2.92	22.00	2.92	5.60	2.51
CloseDoor2	24.60	2.41	4.00	0.00	28.20	2.59	4.60	0.89	22.20	1.92	6.20	1.92	25.80	4.60	4.20	0.45
CloseDrawer1	28.80	2.28	6.40	2.30	27.40	4.83	9.40	3.21	24.00	4.18	6.40	1.52	21.80	4.55	8.60	2.79
CloseDrawer2	25.00	2.65	7.60	3.21	28.80	3.03	6.20	1.48	23.60	2.61	6.00	2.35	21.60	3.71	7.00	3.74
CloseDrawer3	27.20	3.27	4.40	0.55	25.20	4.15	5.00	1.00	26.00	4.12	4.40	0.55	25.40	3.44	4.20	0.45
CloseFridge	26.00	2.55	4.60	0.89	26.60	3.21	5.20	1.10	26.40	3.21	6.20	2.17	27.40	2.51	4.40	0.55
DrinkfromCup	24.40	3.44	4.00	0.00	24.80	3.96	4.40	0.89	25.00	4.00	5.00	1.00	26.20	5.02	4.60	1.34
OpenDishwasher	24.60	3.36	4.60	0.89	24.20	4.21	4.20	0.45	27.00	3.39	5.00	0.00	26.00	2.12	4.80	0.84
OpenDoor1	27.80	5.26	7.20	5.54	28.80	2.77	7.60	5.27	23.20	3.56	5.60	1.82	25.20	1.10	4.60	0.89
OpenDoor2	29.20	2.39	4.40	0.89	25.60	3.29	4.60	0.89	23.20	3.56	4.80	1.10	23.80	1.64	4.40	0.55
OpenDrawer1	25.00	4.30	6.20	2.68	26.00	2.55	9.80	2.17	24.60	2.70	6.00	2.35	27.00	4.85	8.40	7.67
OpenDrawer2	24.00	3.08	6.80	1.30	24.00	3.39	5.80	1.92	25.40	2.19	9.00	5.15	26.20	4.82	5.00	1.00
OpenDrawer3	25.40	4.67	4.20	0.45	26.40	4.22	6.20	2.68	25.80	1.92	5.20	1.79	27.80	3.56	5.40	2.07
OpenFridge	25.20	4.09	5.40	0.89	27.20	4.87	8.80	5.72	27.00	4.69	8.80	5.07	27.00	1.41	5.20	2.17
ToggleSwitch	23.20	1.92	11.40	11.08	26.40	2.70	5.80	1.79	25.60	5.50	11.00	9.67	24.60	2.07	7.80	2.49
Mean	25.60	3.75	5.73	2.06	26.33	3.48	5.99	1.94	24.61	3.48	6.28	2.50	24.99	3.24	5.66	1.91

Table 4. Average cut points and selected features obtained by MOFSD-GR_{ur-CAIM}.

Gesture Names	Subject 1				Subject 2				Subject 3				Subject 4			
	$\mu_{#d}$	$SD_{#d}$	$\mu_{#dp}$	$SD_{#dp}$	$\mu_{#d}$	$SD_{#d}$	$\mu_{#dp}$	$SD_{#dp}$	$\mu_{#d}$	$SD_{#d}$	$\mu_{#dp}$	$SD_{#dp}$	$\mu_{#d}$	$SD_{#d}$	$\mu_{#dp}$	$SD_{#dp}$
CleanTable	13.20	8.64	33.00	22.99	9.00	7.11	14.80	9.04	7.60	7.70	11.60	5.68	11.20	9.83	15.60	21.03
CloseDishwasher	6.80	4.76	17.20	15.67	13.60	7.64	10.40	5.22	2.20	1.30	7.00	5.10	6.20	5.67	22.00	12.75
CloseDoor1	4.60	2.19	12.00	10.17	5.40	2.41	19.00	10.84	10.80	10.03	16.00	11.90	6.80	5.54	17.40	13.56
CloseDoor2	6.60	4.62	10.20	9.12	6.20	5.07	15.40	7.44	7.40	6.19	20.00	24.03	3.40	2.30	10.80	6.06
CloseDrawer1	22.40	5.98	30.60	16.47	16.80	9.26	36.60	25.17	14.00	4.85	41.40	19.05	14.20	7.40	46.80	15.51
CloseDrawer2	16.60	3.21	36.80	25.97	15.40	4.34	37.80	13.81	4.60	1.52	31.60	18.73	14.40	5.77	27.20	7.50
CloseDrawer3	5.40	4.51	7.40	4.77	4.20	1.48	23.40	23.20	5.80	4.97	14.00	11.64	10.60	10.33	22.40	18.19
CloseFridge	7.60	6.50	11.80	6.50	8.40	5.68	26.20	12.01	4.40	2.79	18.20	12.19	10.20	6.06	28.00	10.79
DrinkfromCup	6.80	4.44	12.40	5.86	8.80	10.13	10.40	10.26	3.60	1.52	13.20	5.54	14.00	8.15	13.80	19.16
OpenDishwasher	5.60	6.07	10.40	7.40	9.40	7.02	14.00	10.42	4.00	2.00	9.00	5.48	3.80	2.95	19.20	22.88
OpenDoor1	3.60	1.52	8.60	2.41	7.20	5.12	23.80	18.03	5.00	3.94	9.40	4.93	7.60	4.88	7.40	2.07
OpenDoor2	13.60	7.37	9.00	8.00	6.20	3.27	9.40	3.51	3.80	1.48	15.80	7.26	8.00	3.67	10.60	3.21
OpenDrawer1	11.60	4.93	25.80	5.26	9.40	7.47	36.20	14.11	16.60	10.90	43.80	23.64	11.20	5.12	30.60	17.16
OpenDrawer2	16.20	10.69	37.40	15.50	14.60	8.02	40.40	13.58	6.40	2.19	28.00	20.38	9.80	4.82	38.80	10.83
OpenDrawer3	10.40	7.83	23.20	22.42	8.00	5.00	22.20	18.31	3.20	2.17	8.60	5.86	6.20	5.07	34.40	19.24
OpenFridge	13.20	9.39	35.20	8.20	5.00	2.45	37.20	25.02	2.20	0.45	36.20	16.13	8.40	7.30	38.60	21.61
ToggleSwitch	13.80	9.26	31.80	11.14	17.80	7.66	29.20	18.21	12.00	3.39	35.60	19.82	17.40	6.66	30.60	16.02
Mean	10.47	5.99	20.75	11.64	9.73	5.83	23.91	14.01	6.68	3.96	21.14	12.79	9.61	5.97	24.36	13.97

6. Limitation of the Study

More experimental comparisons against other recent methods or applies on different activity datasets such as Nurse Care Activity Recognition Challenge [61] to demonstrate the effectiveness of the proposed algorithm could be added in this paper. Moreover, other performances metrics could be investigated such as f-measure or feature reduction rate. However, such metrics cannot determine the overall performance of a feature selection algorithm considering both feature selection and discretization. In such a case, other proposed metrics (e.g., score, pareto optimality, and stability) can be employed for an improved analysis.

An optimal solution considers constraints (both Equations (18) and (19) in our proposed method) and then could be a local solution for the given set of data and problem formulated in the decision vector (11). This solution still needs proof of the convergence toward a near global optimum for minimization under the constraints given in Equations (12) to (19). Our approach could be compared with other recent algorithms such as convolutional neural network [37], fuzzy c-mean [62], genetic algorithm [63], particle swarm optimisation [64], and artificial bee colony [28]. However some difficulties arise before comparing and analysing the results: (1) near optimal solution for all algorithms represent a compromise and are difficult to demonstrate, and (2) both simultaneous feature selection and discretization contain many objectives.

7. Conclusions and Future Works

In this paper, we proposed an evolutionary many-objective optimization approach for simultaneously dealing with feature selection, discretization, and classifier parameter tuning for a gesture recognition task. As an illustration, the proposed problem formulation was solved using C-MOEA/DD and an LM-WLCSS classifier. In addition, the discretization sub-problem was addressed using a variable-length structure and a variable-length crossover to overcome the need of specifying the number of elements defining the discretization scheme in advance. Since LM-WLCSS is a binary classifier, the multi-class problem was decomposed using a one-vs.-all strategy, and recognition conflicts were resolved using a light-weight classifier. We conducted experiments on the Opportunity dataset, a real-world benchmark for gesture recognition algorithm. Moreover, a comparison between two discretization criteria, Ameva and ur-CAIM, as a discretization objective of our approach was made. The results indicate that our approach provides better classification performances (an 11% improvement) and stronger reduction capabilities than what is obtainable in similar literature, which employs experimentally chosen parameters, k-means quantization, and hand-crafted sensor unit combinations [19].

In our future work, we plan to investigate search space reduction techniques, such as boundary points [27] and other discretization criteria, along with their decomposition when conflicting objective functions arise. Moreover, efforts will be made to test the approach more extensively either with other dataset or LCS-based classifiers or deep learning approach. A mathematical analysis using a dynamic system, such as Markov chain, will be defined to prove and explain the convergence toward an optimal solution of the proposed method. The backtracking variable length, B_c , is not a major performance limiter in the learning process. In this sense, it would be interesting to see additional experiments showing the effects of several values of this variable on the recognition phase and, ideally, how it affects the NADX operator.

Our ultimate goal is to provide a new framework to efficiently and effortlessly tackle the multi-class gesture recognition problem.

Author Contributions: Conceptualization, J.V.; methodology, J.V.; formal analysis, M.J.-D.O. and J.V.; investigation, M.J.-D.O. and J.V.; resources, M.J.-D.O.; data curation, J.V.; writing—original draft preparation, J.V. and M.J.-D.O.; writing—review and editing, J.V. and M.J.-D.O.; supervision,

M.J.-D.O.; project administration, M.J.-D.O.; funding acquisition, M.J.-D.O. All authors have read and agreed to the published version of the manuscript.

Funding: While performing this project, J.V. received a scholarship from REPARTI Strategic Network supported by Fonds québécois de la recherche sur la nature et les technologies (FRQ-NT). This work was supported by The Natural Sciences and Engineering Research Council of Canada (NSERC) under the grant number 418235-2012 and RGPIN-2018-06329 as well as by Fond de Recherche du Québec—Nature et Technologie (FRQ-NT) under the grant number 2016-PR-188869. We thank the REPARTI Center (strategic network) for its financial support coming from FRQ-NT.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the open access database used in this study.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset analysed in this study is available following this link: <https://archive.ics.uci.edu/ml/datasets/opportunity+activity+recognition> (accessed on 15 September 2021).

Acknowledgments: The authors thank Sophie Lasfargeas (University of Quebec at Chicoutimi) for her constructive comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Byrne, R.W.; Cartmill, E.; Genty, E.; Graham, K.E.; Hobaiter, C.; Tanner, J. Great ape gestures: intentional communication with a rich set of innate signals. *Anim. Cogn.* **2017**, *20*, 755–769. [[CrossRef](#)]
2. Yu, Z.; Chen, H.; Liu, J.; You, J.; Leung, H.; Han, G. Hybrid k -Nearest Neighbor Classifier. *IEEE Trans. Cybern.* **2016**, *46*, 1263–1275. [[CrossRef](#)]
3. Amma, C.; Georgi, M.; Schultz, T. Airwriting: a wearable handwriting recognition system. *Pers. Ubiquitous Comput.* **2014**, *18*, 191–203. [[CrossRef](#)]
4. Galka, J.; Masiar, M.; Zaborski, M.; Barczewska, K. Inertial Motion Sensing Glove for Sign Language Gesture Acquisition and Recognition. *IEEE Sens. J.* **2016**, *16*, 6310–6316. [[CrossRef](#)]
5. Lu, Z.; Chen, X.; Li, Q.; Zhang, X.; Zhou, P. A Hand Gesture Recognition Framework and Wearable Gesture-Based Interaction Prototype for Mobile Devices. *IEEE Trans. Hum.-Mach. Syst.* **2014**, *44*, 293–299. [[CrossRef](#)]
6. Benatti, S.; Casamassima, F.; Milosevic, B.; Farella, E.; Schönle, P.; Fateh, S.; Burger, T.; Huang, Q.; Benini, L. A Versatile Embedded Platform for EMG Acquisition and Gesture Recognition. *IEEE Trans. Biomed. Circuits Syst.* **2015**, *9*, 620–630. [[CrossRef](#)]
7. Geng, Y.; Chen, J.; Fu, R.; Bao, G.; Pahlavan, K. Enlighten Wearable Physiological Monitoring Systems: On-Body RF Characteristics Based Human Motion Classification Using a Support Vector Machine. *IEEE Trans. Mob. Comput.* **2016**, *15*, 656–671. [[CrossRef](#)]
8. Fukui, R.; Watanabe, M.; Shimosaka, M.; Sato, T. Hand shape classification in various pronation angles using a wearable wrist contour sensor. *Adv. Robot.* **2015**, *29*, 3–11. [[CrossRef](#)]
9. Cifuentes, J.; Boulanger, P.; Pham, M.T.; Prieto, F.; Moreau, R. Gesture Classification Using LSTM Recurrent Neural Networks. In Proceedings of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019; pp. 6864–6867.
10. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2019**, *119*, 3–11. [[CrossRef](#)]
11. Shokoohi-Yekta, M.; Hu, B.; Jin, H.; Wang, J.; Keogh, E. Generalizing DTW to the multi-dimensional case requires an adaptive approach. *Data Min. Knowl. Discov.* **2017**, *31*, 1–31. [[CrossRef](#)]
12. Dindo, H.; Presti, L.L.; Cascia, M.L.; Chella, A.; Dedić, R. Hankelet-based action classification for motor intention recognition. *Robot. Auton. Syst.* **2017**, *94*, 120–133. [[CrossRef](#)]
13. Rakthanmanon, T.; Campana, B.; Mueen, A.; Batista, G.; Westover, B.; Zhu, Q.; Zakaria, J.; Keogh, E. Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. *ACM Trans. Knowl. Discov. Data* **2013**, *7*, 10:1–10:31. [[CrossRef](#)]
14. Vlachos, M.; Kollios, G.; Gunopulos, D. Discovering similar multidimensional trajectories. In Proceedings 18th International Conference on Data Engineering, San Jose, CA, USA, 26 February–1 March 2002; pp. 673–684. [[CrossRef](#)]
15. Frolova, D.; Stern, H.; Berman, S. Most Probable Longest Common Subsequence for Recognition of Gesture Character Input. *IEEE Trans. Cybern.* **2013**, *43*, 871–880. [[CrossRef](#)]
16. Stern, H.; Shmueli, M.; Berman, S. Most discriminating segment—Longest common subsequence (MDSLCS) algorithm for dynamic hand gesture classification. *Pattern Recognit. Lett.* **2013**, *34*, 1980–1989. [[CrossRef](#)]
17. Nyirarugira, C.; Kim, T. Stratified gesture recognition using the normalized longest common subsequence with rough sets. *Signal Process. Image Commun.* **2015**, *30*, 178–189. [[CrossRef](#)]

18. Nguyen-Dinh, L.V.; Calatroni, A.; Tröster, G. Robust Online Gesture Recognition with Crowdsourced Annotations. *J. Mach. Learn. Res.* **2014**, *15*, 3187–3220.
19. Nguyen-Dinh, L.V.; Calatroni, A.; Troster, G. Towards a Unified System for Multimodal Activity Spotting: Challenges and a Proposal. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, Seattle Washington, WA, USA, 13–17 September 2014; ACM: New York, NY, USA, 2014; pp. 807–816. [[CrossRef](#)]
20. Hardegger, M.; Roggen, D.; Calatroni, A.; Troster, G. S-SMART: A Unified Bayesian Framework for Simultaneous Semantic Mapping, Activity Recognition, and Tracking. *ACM Trans. Intell. Syst. Technol.* **2016**, *7*, 34:1–34:28. [[CrossRef](#)]
21. Roggen, D.; Cuspinera, L.P.; Pombo, G.; Ali, F.; Nguyen-Dinh, L.V., Limited-Memory Warping LCSS for Real-Time Low-Power Pattern Recognition in Wireless Nodes. In *Wireless Sensor Networks: 12th European Conference, EWSN, Proceedings*; Springer International Publishing: Porto, Portugal, 2015; pp. 151–167. [[CrossRef](#)]
22. Chan, M.; Estève, D.; Fourniols, J.Y.; Escriba, C.; Campo, E. Smart wearable systems: Current status and future challenges. *Artif. Intell. Med.* **2012**, *56*, 137–156. [[CrossRef](#)]
23. Unler, A.; Murat, A. A discrete particle swarm optimization method for feature selection in binary classification problems. *Eur. J. Oper. Res.* **2010**, *206*, 528–539. [[CrossRef](#)]
24. Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Trans. Evol. Comput.* **2016**, *20*, 606–626. [[CrossRef](#)]
25. Tahan, M.H.; Asadi, S. MEMOD: a novel multivariate evolutionary multi-objective discretization. *Soft Comput.* **2017**, *22*, 1–23. [[CrossRef](#)]
26. Garcia, S.; Luengo, J.; Saez, J.A.; Lopez, V.; Herrera, F. A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 734–750. [[CrossRef](#)]
27. Ramírez-Gallego, S.; García, S.; Benítez, J.M.; Herrera, F. Multivariate Discretization Based on Evolutionary Cut Points Selection for Classification. *IEEE Trans. Cybern.* **2016**, *46*, 595–608. [[CrossRef](#)]
28. Wang, X.H.; Zhang, Y.; Sun, X.Y.; Wang, Y.L.; Du, C.H. Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size. *Appl. Soft Comput. J.* **2020**, *88*, 106041 [[CrossRef](#)]
29. Yang, W.; Chen, L.; Wang, Y.; Zhang, M. Multi-Many-Objective Particle Swarm Optimization Algorithm Based on Competition Mechanism. *Comput. Intell. Neurosci.* **2020**, *2020*, 5132803. [[CrossRef](#)]
30. Cano, A.; Nguyen, D.T.; Ventura, S.; Cios, K.J. ur-CAIM: improved CAIM discretization for unbalanced and balanced data. *Soft Comput.* **2016**, *20*, 173–188. [[CrossRef](#)]
31. Zhou, Y.; Kang, J.; Kwong, S.; Wang, X.; Zhang, Q. An evolutionary multi-objective optimization framework of discretization-based feature selection for classification. *Swarm Evol. Comput.* **2021**, *60*, 100770. [[CrossRef](#)]
32. Cheng, R.; Jin, Y. A competitive swarm optimizer for large scale optimization. *IEEE Trans. Cybern.* **2015**, *45*, 191–204. [[CrossRef](#)]
33. Yu, X.; Zhang, X. Multiswarm comprehensive learning particle swarm optimization for solving multiobjective optimization problems. *PLoS ONE* **2017**, *12*, e0172033. [[CrossRef](#)]
34. Zhou, Y.; Kang, J.; Guo, H. Many-objective optimization of feature selection based on two-level particle cooperation. *Inf. Sci.* **2020**, *532*, 91–109. [[CrossRef](#)]
35. Sharmin, S.; Shoyaib, M.; Ali, A.A.; Khan, M.A.H.; Chae, O. Simultaneous feature selection and discretization based on mutual information. *Pattern Recognit.* **2019**, *91*, 162–174. [[CrossRef](#)]
36. Roy, P.; Sharmin, S.; Ali, A.; Shoyaib, M. *Discretization and Feature Selection Based on Bias Corrected Mutual Information Considering High-Order Dependencies*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Singapore, 2020; Volume 12084, pp. 830–842. [[CrossRef](#)]
37. Lu, H.Y.; Zhang, M.; Liu, Y.Q.; Ma, S.P. Convolution Neural Network Feature Importance Analysis and Feature Selection Enhanced Model. *Ruan Jian Xue Bao/J. Softw.* **2017**, *28*, 2879–2890. [[CrossRef](#)]
38. Gong, M.; Liu, J.; Li, H.; Cai, Q.; Su, L. A multiobjective sparse feature learning model for deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 3263–3277. [[CrossRef](#)] [[PubMed](#)]
39. Tsai, C.F.; Chen, Y.C. The optimal combination of feature selection and data discretization: An empirical study. *Inf. Sci.* **2019**, *505*, 282–293. [[CrossRef](#)]
40. Li, K.; Deb, K.; Zhang, Q.; Kwong, S. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Trans. Evol. Comput.* **2015**, *19*, 694–716. [[CrossRef](#)]
41. Ryerkerk, M.L.; Averill, R.C.; Deb, K.; Goodman, E.D. Solving metameric variable-length optimization problems using genetic algorithms. *Genet. Program. Evolvable Mach.* **2017**, *18*, 247–277. [[CrossRef](#)]
42. Al-Dabbagh, M.D.; Al-Dabbagh, R.D.; Abdullah, R.R.; Hashim, F. A new modified differential evolution algorithm scheme-based linear frequency modulation radar signal de-noising. *Eng. Optim.* **2015**, *47*, 771–787. [[CrossRef](#)]
43. Zhu, D.; Wang, L.; Wu, Y.; Wang, X. A Practical $O(R \log \log n + n)$ time Algorithm for Computing the Longest Common Subsequence. *CoRR* **2015**, *44*, abs/1508.05553.
44. Zhang, Q.; Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [[CrossRef](#)]
45. Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [[CrossRef](#)]

46. García, S.; López, V.; Luengo, J.; Carmona, C.J.; Herrera, F. A Preliminary Study on Selecting the Optimal Cut Points in Discretization by Evolutionary Algorithms. *ICPRAM* **2012**, *2012*, 211–216.
47. Eshelman, L.J. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In *Foundations of Genetic Algorithms*; RAWLINS, G.J., Ed.; Elsevier: Amsterdam, The Netherlands, 1991; Volume 1, pp. 265–283. [[CrossRef](#)]
48. Tsai, C.J.; Lee, C.I.; Yang, W.P. A discretization algorithm based on Class-Attribute Contingency Coefficient. *Inf. Sci.* **2008**, *178*, 714–731. [[CrossRef](#)]
49. Gonzalez-Abril, L.; Cuberos, F.; Velasco, F.; Ortega, J. Ameva: An autonomous discretization algorithm. *Expert Syst. Appl.* **2009**, *36*, 5327–5332. [[CrossRef](#)]
50. Soria Morillo, L.M.; Alvarez-Garcia, J.A.; Gonzalez-Abril, L.; Ortega Ramirez, J.A. Discrete classification technique applied to TV advertisements liking recognition system based on low-cost EEG headsets. *Biomed. Eng. Online* **2016**, *15*, 75. [[CrossRef](#)]
51. Ángel Álvarez de la Concepción, M.; Morillo, L.M.S.; Álvarez García, J.A.; González-Abril, L. Mobile activity recognition and fall detection system for elderly people using Ameva algorithm. *Pervasive Mob. Comput.* **2017**, *34*, 3–13. [[CrossRef](#)]
52. Wagner, R.A.; Fischer, M.J. The String-to-String Correction Problem. *J. ACM* **1974**, *21*, 168–173. [[CrossRef](#)]
53. Iliopoulos, C.S.; Rahman, M.S. New efficient algorithms for the LCS and constrained LCS problems. *Inf. Process. Lett.* **2008**, *106*, 13–18. [[CrossRef](#)]
54. Laskany, G.S.; Trabia, M.B. A genetic algorithm with weighted average normally-distributed arithmetic crossover and twinkling. *Appl. Math.* **2012**, *3*, 1220–1235. [[CrossRef](#)]
55. Ben-David, A. A lot of randomness is hiding in accuracy. *Eng. Appl. Artif. Intell.* **2007**, *20*, 875–885. [[CrossRef](#)]
56. Roggen, D.; Calatroni, A.; Rossi, M.; Holleczeck, T.; Förster, K.; Troster, G.; Lukowicz, P.; Bannach, D.; Pirkl, G.; Ferscha, A.; et al. Collecting complex activity datasets in highly rich networked sensor environments. In Proceedings of the 2010 Seventh International Conference on Networked Sensing Systems (INSS), Kassel, Germany, 15–18 June 2010, pp. 233–240. [[CrossRef](#)]
57. Ordonez, F.J.; Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* **2016**, *16*, 115. [[CrossRef](#)]
58. Chavarriaga, R.; Sagha, H.; Calatroni, A.; Digumarti, S.T.; Tröster, G.; del R. Millán, J.; Roggen, D. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognit. Lett.* **2013**, *34*, 2033–2042. [[CrossRef](#)]
59. Chen, Y.L.; Wu, X.; Li, T.; Cheng, J.; Ou, Y.; Xu, M. Dimensionality reduction of data sequences for human activity recognition. *Neurocomputing* **2016**, *210*, 294–302. [[CrossRef](#)]
60. Ramirez-Gallego, S.; Krawczyk, B.; Garcia, S.; Wozniak, M.; Herrera, F. A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing* **2017**, *239*, 39–57. [[CrossRef](#)]
61. Inoue, S.; Lago, P.; Takeda, S.; Shamma, A.; Faiz, F.; Mairittha, N.; Mairittha, T. Nurse Care Activity Recognition Challenge. *IEEE Dataport* **2019**. [[CrossRef](#)]
62. Lin, H.Y. Feature clustering and feature discretization assisting gene selection for molecular classification using fuzzy c-means and expectation–maximization algorithm. *J. Supercomput.* **2021**, *77*, 5381–5397. [[CrossRef](#)]
63. Zhou, Y.; Zhang, W.; Kang, J.; Zhang, X.; Wang, X. A problem-specific non-dominated sorting genetic algorithm for supervised feature selection. *Inf. Sci.* **2021**, *547*, 841–859. [[CrossRef](#)]
64. Hu, Y.; Zhang, Y.; Gong, D. Multiobjective Particle Swarm Optimization for Feature Selection with Fuzzy Cost. *IEEE Trans. Cybern.* **2021**, *51*, 874–888. [[CrossRef](#)]