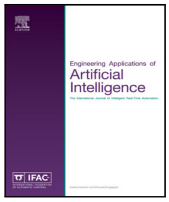
Engineering Applications of Artificial Intelligence 111 (2022) 104822

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Diagnosing Alzheimer's disease from on-line handwriting: A novel dataset and performance benchmarking

Nicole D. Cilia [a,b,1], Giuseppe De Gregorio [c,1], Claudio De Stefano [a,1], Francesco Fontanella [a,*,1], Angelo Marcelli [c,d,1], Antonio Parziale [c,d,1]

[a] Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Italy
[b] Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands
[c] Department of Electrical and Information Engineering and Applied Mathematics, University of Salerno, Italy
[d] CINI Laboratory of Artificial Intelligence and Intelligent Systems, Unit of Salerno, Italy

ARTICLE INFO

Keywords:
Neurodegenerative diseases
Health data
Alzheimer's disease prediction
Handwriting analysis
Classification and combining strategies

ABSTRACT

Neurodegenerative diseases are caused by the progressive degeneration of nerve cells that affect motor skills and cognitive abilities with increasing severity. Unfortunately, there is no cure for this type of disease and their impact can only be slowed down with specific pharmacological and rehabilitative therapies. Early diagnosis, therefore, remains the primary means to delay brain damage and improve the quality of life of people affected. Neurodegenerative diseases also affect movement fine control. Consequently, the analysis of handwriting dynamics can represent an effective tool to support an early diagnosis of these diseases. While many methods have been proposed in the literature based on the use of a wide range of handwriting tasks, researchers have not yet defined a universally accepted standard experimental protocol to collect data. Furthermore, although some databases containing handwriting data have been produced, only a few of them were designed specifically for research on neurodegenerative diseases, and, in most cases, they involve a small number of participants performing a few tasks. Here, we introduce the DARWIN (Diagnosis AlzheimeR WIth haNdwriting) dataset to overcome these drawbacks, which contains handwriting samples from people affected by Alzheimer's and a control group. The dataset includes data from 174 participants, acquired during the execution of handwriting tasks, performed according to a protocol specifically designed for the early detection of Alzheimer's. We report the results of the experiments performed to evaluate the effectiveness of the proposed tasks and features in capturing the distinctive aspects of handwriting that support the diagnosis of Alzheimer's disease.

1. Introduction

Neurodegenerative diseases (NDs in the following) are incurable and debilitating conditions caused by progressive degeneration of nerve cells. They may affect movements and/or mental skills, with Alzheimer's and Parkinson's diseases (respectively PD and AD in the following) being the most common among them.

PD mainly affects the motor system, and the most common symptoms are tremor, rigidity, slowness of movement, and difficulty with walking. As the disease worsens, cognitive symptoms, usually referred as Parkinson's disease dementia, become common. The motor symptoms of the disease result from the death of cells in the substantia nigra, a region of the midbrain, leading to a dopamine deficit (Kalia and Lang, 2015). AD produces a slow and progressive decline in mental functions such as memory, thought, judgment, and learning abilities. In the early stages of the disease, the predominant symptom is the episodic memory impairment that is indicative of ventromedial temporal lobe dysfunction (Armstrong et al., 2013). After that, it is typically followed by progressive amnesia and deterioration in other cognitive domains, showing pathological involvement of more widespread neural systems.

There is no cure for these diseases and the decline can only be somehow managed during their progression. Because of worldwide lifespan lengthening, it is expected that the incidence of NDs will dramatically increase in the coming decades. This creates a critical need for the improvement of the approaches currently used for diagnosing them as early as possible. As cognitive and motor functions are both involved in planning and execution of movements, and because handwriting requires a precise and properly coordinated control of the body (Precup et al., 2020), the analysis of handwriting dynamics might provide a cheap and non-invasive method for evaluating the disease

* Corresponding author.
E-mail addresses: nicoledalia.cilia@unicas.it (N.D. Cilia), gdegregorio@unisa.it (G. De Gregorio), destefano@unicas.it (C. De Stefano), fontanella@unicas.it (F. Fontanella), amarcelli@unisa.it (A. Marcelli), anparziale@unisa.it (A. Parziale).
[1] All authors contributed equally to the work.

https://doi.org/10.1016/j.engappai.2022.104822
Received 12 October 2021; Received in revised form 20 January 2022; Accepted 5 March 2022
Available online 21 March 2022

progression (Impedovo et al., 2018). Furthermore, it has been observed that the application of machine learning methods to motor function has shown promise in decreasing the time taken to perform clinical assessments (Myszczynska et al., 2020a; Albu et al., 2019). To this aim, cheap and widely used graphic tablets can be used to administer handwriting tests, which include simple and easy-to-perform handwriting/drawing tasks (Impedovo et al., 2018), and to record kinematic and dynamic information of the performed movements. For this reason, researchers are showing an increasing interest in developing and using machine learning based methodologies to support both the diagnosis and the treatment of NDs, and several methods have been proposed for the diagnosis of both AD (Tanveer et al., 2020) and PD (Pereira et al., 2019).

An effective evaluation of handwriting alterations requires the definition of general criteria for carrying out tests able to highlight the first ND signs. The methods proposed in the literature make use of a wide range of tasks, features and classifiers (Vessio, 2019; Pozna and Precup, 2014). However, previous studies did not use sufficiently large datasets or an agreed set of features. To overcome these drawbacks, we have defined a protocol for handwriting data collection and suggested a set of features to be extracted. Both are based on findings in neuroscience and motor control regarding the role played by different brain areas of the brain in learning and executing handwriting and drawing tasks (Cilia et al., 2018) and how their malfunctioning is reflected in the execution of the tasks (Senatore and Marcelli, 2019a). The protocol is comprised of 25 tasks, with different levels of complexity and targeting different areas of the brain. The handwriting/drawing movements performed to execute each task are then described by using 18 features. Our preliminary results on the dataset showed that different tasks complement each others in such a way that the combination of the information brought by the execution of each of them led to a multi-classifier achieving state-of-the-art performance (De Gregorio et al., 2021).

In this paper, we present in detail the procedure we have used for recruiting the participants, both AD patients and healthy people, and for collecting the handwriting data making up the DARWIN dataset (Diagnosis AlzheimeR WIth haNdwriting). We also describe a large set of experiments we have performed for validating the rationale behind the protocol design, the ability of the proposed feature set to capture the distinctive aspects of the movements performed while executing the task, and to what extent they can be exploited by a machine learning based system to effectively discriminate between AD patients and healthy people. To this purpose, we have systematically evaluated how the parameters characterizing the systems we have developed affect the overall performance, and performed a statistical analysis of the results, with the ultimate goal of providing solid evidence to the viability of handwriting analysis as a tool to support the diagnosis of AD.

The main contributions of the paper can be summarized as follows:

- a novel dataset containing handwriting data for the prediction of AD. The dataset is the largest publicly available in terms of the number of participants and the number of tasks performed.[2] This data will contribute to: (i) overcome the lack of data, one of the major limitations of the previous studies; (ii) favor a fair performance comparison of existing and future methodologies and tools for AD prediction via handwriting analysis;
- the results of a large set of experiments, designed and performed with the aim of: (i) evaluating the effectiveness of the proposed features set to discriminate between AD patients and healthy people when exploited by well-known and widely-used classifiers; (ii) evaluating to what extent the different tasks envisaged by the protocol contribute valuable diagnostic information; (iii) providing researchers working in the field with some baseline results on the data proposed, to favor comparison between different approaches to the automatic diagnosis of AD.

[2] The dataset is available at the following page: http://webuser.unicas.it/fontanella/darwin.

The remainder of the paper is organized as follows. Section 2 discusses the related work, whereas Section 3 describes how the data have been collected and the features extracted. Section 4 presents the architectures we have adopted for performance benchmarking, whereas Section 5 outlines the classification models used to validate our protocol as well as the set of features extracted. Section 6 describes the experiments we have performed and reports the results we have obtained. Concluding remarks and outline of possible future investigations are eventually left to Section 7.

2. Related work

In the last few years Machine learning based tools are demonstrating their ability to solve a wide spectrum of real-world problems (Jain et al., 2019; Borlea et al., 2021). However, to be used effectively, these tools need benchmark data which allows a fair comparison of the solutions viable for a given problem.

As mentioned in the Introduction, most of the publicly available datasets contain handwriting data from PD patients, and most of them were collected from small groups of participants. The Parkinson's Disease Handwriting Database (PaHaW) consists of handwritten words in the Czech language. They were collected from 37 participants affected by PD and a control group made of 38 people (P. Drotár et al., 2013). The authors of this study selected words that allowed participants to write without lifting the pen from the writing surface. Data were acquired using a tablet overlaid with a white template paper and a conventional ink pen.

The HandPD dataset contains data of handwriting/drawing tasks, collected from 92 participants (18 healthy controls and 74 PD patients). It contains images of the handwriting data produced by the participants while tracing four copy of spirals and meanders (Pereira et al., 2016a). The new version of the dataset, called NewHandPD contains data from 66 participants (35 healthy people and 31 patients). Each individual was requested to execute twelve drawing tasks, four of them related to spirals and four related to meanders as in the HandPD dataset, two tasks involving circular movements (one circle in the air and another on the paper), and left and right-handed diadochokinesis. During the execution of the tasks, the handwritten dynamics was recorded by using the BiSP smart pen (Pereira et al., 2016b).

The ParkinsonHW dataset (Isenkul et al., 2014), contains handwriting data from 77 participants (62 PD patients and 15 healthy people). They performed the following tasks:

- static spiral test (SST): three Archimedean spirals were displayed on the tablet screen and participants were asked to retrace the spiral;
- dynamic spiral test (DST): the Archimedean spiral to retrace appeared and disappeared at a given time intervals;
- stability test on a certain point (STCP): a red point was displayed in the middle of the tablet screen and participants were asked to hold the pen on the point, but without touching the screen.

The data also include the images of the spirals drawn by the PD patients.

The availability of these datasets has favored the development of many studies, using different combinations of features and classifiers, and the reported results, although not always directly comparable, allows for a reliable estimation of the performance of the state-of-the-art methods (P. Drotár et al., 2013; Drotár et al., 2016; Pereira et al., 2019; Myszczynska et al., 2020b; Parziale et al., 2021; Cavaliere et al., 2020; Senatore and Marcelli, 2019b; Parziale et al., 2019). On the contrary, the only public dataset including samples from AD patients is the ISUNIBA dataset (Pirlo et al., 2015b). The dataset contains handwritten traits collected from 29 AD patients and a control group made up of 12 people. Each participant was asked to write the word "mamma" (i.e. Italian of "mom") for a given number of times. The authors chose that word for two reasons: (i) it is one of the first words babies learn to speak; (ii) it has been observed that it is repeated with high frequency by people in an advanced state of AD.

Table 1
List of tasks performed. Task categories are: memory and dictation (M), Graphic (G), and Copy (C).

#	Description	Category
1	Signature drawing	M
2	Join two points with a horizontal line, continuously for four times	G
3	Join two points with a vertical line, continuously for four times	G
4	Retrace a circle (6 cm of diameter) continuously for four times	G
5	Retrace a circle (3 cm of diameter) continuously for four times	G
6	Copy the letters 'l', 'm' and 'p'	C
7	Copy the letters on the adjacent rows	C
8	Write cursively a sequence of four lowercase letter 'l', in a single smooth movement	C
9	Write cursively a sequence of four lowercase cursive bigram 'le', in a single smooth movement	C
10	Copy the word "foglio"	C
11	Copy the word "foglio" above a line	C
12	Copy the word "mamma"	C
13	Copy the word "mamma" above a line	C
14	Memorize the words "telefono", "cane", and "negozio" and rewrite them	M
15	Copy in reverse the word "bottiglia"	C
16	Copy in reverse the word "casa"	C
17	Copy six words (regular, non regular, non words) in the appropriate boxes	C
18	Write the name of the object shown in a picture (a chair)	M
19	Copy the fields of a postal order	C
20	Write a simple sentence under dictation	M
21	Retrace a complex form	G
22	Copy a telephone number	C
23	Write a telephone number under dictation	M
24	Draw a clock, with all hours and put hands at 11:05 (Clock Drawing Test)	G
25	Copy a paragraph	C

3. Data collection and feature extraction

Our handwriting data were collected according to the acquisition protocol described in Cilia et al. (2018). The protocol includes 25 tasks, belonging to the following categories (see Table 1):

- Graphic tasks: tested participant's ability in writing elementary traits; they include joining some points and drawing geometrical figures;
- Copy tasks: evaluated participant's abilities in repeating complex graphic gestures, which have semantic meaning such as letters, words and numbers;
- Memory tasks: tested the changes in writing process previously memorized or associated with objects shown in a picture;
- Dictation tasks: investigated how handwriting varies when the working memory is used.

The dataset contains data from 174 participants: 89 AD patients and 85 healthy people.

Participants were recruited using standard clinical tests, namely, Mini-Mental State Examination (MMSE), Frontal Assessment Battery (FAB), and Montreal Cognitive Assessment (MoCA). These tests use questionnaires to assess cognitive skills covering many areas, ranging from orientation in time and place to registration recall. We also used the following exclusion criteria: (i) taking psychotropic medication or any other drugs influencing cognitive abilities; (ii) too compromised cognitive abilities, according to the evaluation of medical experts.

To avoid any bias, participants were recruited in such a way that the two groups matched in terms of age, level of education, type of work (manual or intellectual) and gender (see Table 2). All of them read and signed an informed consensus form describing the purpose of the data collection and detailing the data protection policy governing the storage and use of their data.

3.1. Data acquisition

To acquire the data, we used a Wacom's Bamboo tablet equipped with a pen that allowed participants to in ink on A4 white paper sheets placed on it. For each task, the tablet sampled the x–y coordinates of the pen tip movements at a frequency of 200 Hz. The coordinates can be subdivided into two categories: "on-paper" and "in-air". The first are recorded when the pen tip touches the paper, whereas the second are recorded when the pen tip is lifted from the piece of paper within a maximum distance of 3 cm. In the first case, the pressure exerted by the pen tip on the paper was also sampled.

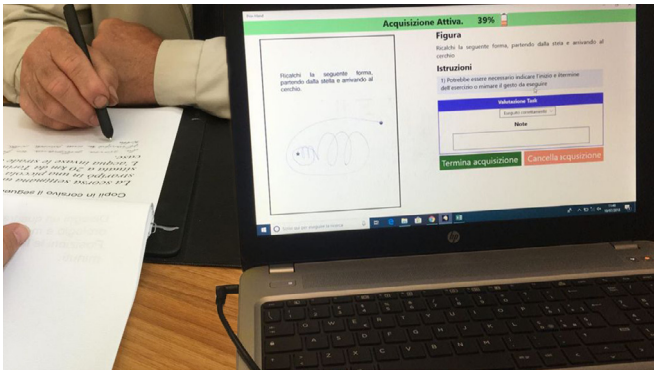


Fig. 1. Example of data acquisition. The acquisition software records the movements performed by the participant on the paper thanks to the use of a graphic tablet.

During data acquisition, the participant was sitting behind a table in a comfortable position, with the tablet set atop the table and in front of the participant. On top of the tablet, there was a block of 25 paper sheets, once for each task, stapled together and fixed on the tablet by using the slot in the tablet cover. On each sheet were printed the description of the task to be executed as well as any other data needed, depending on the task. The tablet was connected to a PC, for administering the test. During the test, the data acquisition software shows on the monitor the format of the paper sheet for each specific task and displays in real-time the movements as captured by the tablet, as shown in Fig. 1. Once a task has been executed, the participant turns the page, read the assignment and proceeds, and so does the operator by using the GUI of the tool. After the test completion, the raw data were processed to extract and store in the dataset the features described in the following.

3.2. Feature extraction

For each task, from the raw data, i.e. (x,y)-coordinates, pressure and timestamp, we extracted 18 features, detailed in the following.

Table 2
Average demographic data of participants. Standard deviations are shown in parentheses.

	Age	Education	#Women	#Men
Patients	71.5 (9.5)	10.8 (5.1)	46	44
Control group	68.9 (12)	12.9 (4.4)	51	39

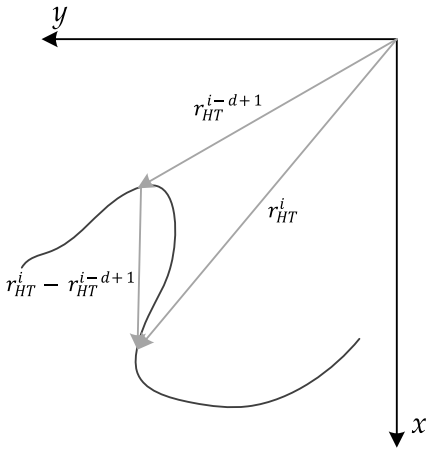


Fig. 2. An example of computation of the GMRTP feature.

Total Time (TT): Total time spent to perform the entire task.

Air Time (AT): Time spent to perform in-air movements.

Paper Time (PT): Time spent to perform on-paper movements.

Mean Speed on-paper (MSP): Average speed of on-paper movements. Speed is the variation of displacement with respect to time.

Mean Speed in-air (MSA): Average speed of in-air movements.

Mean Acceleration on-paper (MAP): Average acceleration of on-paper movements. Acceleration is the variation of speed with respect to time.

Mean Acceleration in-air (MAA): Average acceleration of in-air movements.

Mean Jerk on-paper (MJP): Average jerk of on-paper movements. Jerk is the variation of acceleration with respect to time.

Mean Jerk in-air (MJA): Average jerk of in-air movements.

Pressure Mean (PM): Average of the pressure levels exerted by the pen tip.

Pressure Var (PV): Variance of the pressure levels exerted by the pen tip.

GMRT on-paper (GMRTP): Generalization of the Mean Relative Tremor (MRT) as defined in Pereira et al. (2015) and computed for on-paper movements. MRT measures the amount of tremor in drawing spirals and meanders. The feature is equal to the average distance between the ith sample of the written trace and another one taken d samples before:

$$\frac{1}{n-d}\sum_{i=d}^{n}|r_{HT}^{i} - r_{HT}^{(i-d+1)}|$$

where n is the number of sample points, d is the displacement of the sample points used to compute the radius difference, and r^i is the ith spiral radius of the handwritten trace, i.e. the distance between the ith point and the center of the spiral. To generalize MRT, it is crucial to define a reference system that is valid for a generic drawing or writing task. Since all tasks of the data set were acquired by using a standard sheet of paper, we took the top right corner of the sheet as origin of the reference system. Therefore, in GMRT r^i is the distance between the ith point of the trace and the origin of the reference system. As suggested in Pereira et al. (2015), we set $d = 10$. An example of computation of the GMRTP is shown in Fig. 2.

GMRT in-air (GMRTA): Generalization of the Mean Relative Tremor computed on in air movements.

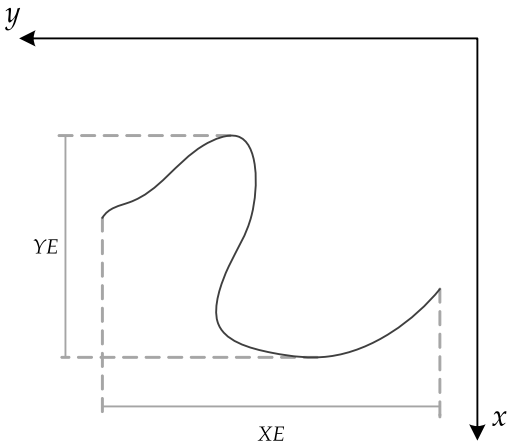


Fig. 3. An example of computation of the XE and YE features.

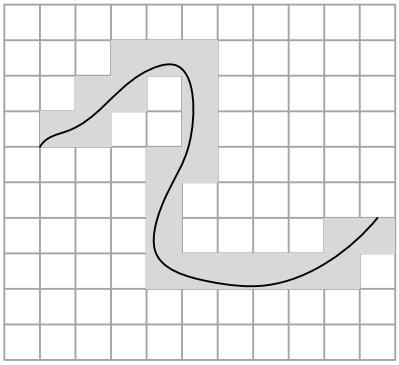


Fig. 4. An example of computation of the Dispersion Index feature. $DI = 21/110 = 0.191$.

Mean GMRT (GMRT): Average of GMRTP and GMRTA.

Pendowns Number (PWN): Counts the total number of pendowns recorded during the execution of the entire task (e.g., a continuous uninterrupted line present a pendowns number equal to 1).

Max X Extension (XE): Maximum extension recorded along the X axis. The maximum extension of a component along an axis is calculated considering the difference between its farthest/nearest points to the origin on the considered axis (see Fig. 3).

Max Y Extension (YE): Maximum extension recorded along the Y axis. Computed the same as the XE feature, but taken into account the y axis (see Fig. 3).

Dispersion Index (DI): The Dispersion Index measures how the handwritten trace is "dispersed" on the entire piece of paper; in other words, it measures how much of the sheet is covered. To calculate the index, the sheet is ideally divided into TB fixed-size boxes of 3×3 pixels, and then it is computed the number CB of boxes containing a fragment of handwriting/drawing. Eventually, DI is given by the ratio between CB and TB. An example of computation of the DI feature is shown in Fig. 4.

4. Rationale of the experiments and benchmark architectures

As already mentioned, the test we have designed comprises 25 tasks, targeting to different extents the brain areas involved in fine motor planning and execution that may be affected by AD. Some of them have been already considered in previous studies, whereas others have been derived from recent findings in neuroscience and motor control of handwriting.

As different tasks can elicit different handwriting alterations, in our previous work (De Gregorio et al., 2021) we considered two scenarios.

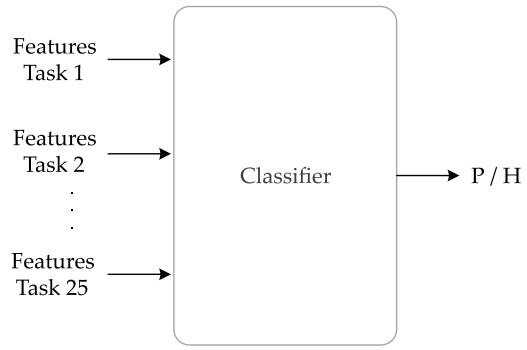


Fig. 5. Single classifier working on features set obtained by merging each task features value.

The first evaluated to what extent the test as a whole is capable of providing valuable information for the purpose concerned. Thus, for each participant, the features extracted from each task were combined into a unique feature vector and used to build a classifier capable of discriminating between AD patients and healthy people, as depicted in Fig. 5. The second, on the contrary, was designed to evaluate to what extent the information brought by each task contributes to the final decision. To this aim, the feature vectors extracted from each task were used to build a task-specific classifier, and the final classification was achieved by combining their outputs (see Fig. 6).

As the purpose of the experiments is primarily that of validating the protocol we adopted for data collection and the features we extract during the execution of the tasks, the architectures of Figs. 5 and 6 were implemented by adopting different classifiers, so as to mitigate the effect of strengths and weaknesses of each classifier on performance. In the case of the architecture in Fig. 6, the same classifier was adopted to build the 25 task-specific classifiers to be combined, whereas in the second case the top-performing classifiers were combined.

To evaluate whether it would be possible to reduce the complexity of the test (and thus the time needed to administer it), for each classifier we sorted the tasks in descending order according to the accuracy shown by the classifier. Then, by computing the cumulative accuracy incrementally, we were able to select the subset of tasks corresponding to the best performance for each type of classifier. In all the implementations of the architecture depicted in Fig. 6, the outputs of the classifiers were combined using the majority vote rule.

5. Classification models

To validate our assumptions about the protocol used for collecting the handwriting data, as well as to evaluate to what extent the set of features we have adopted is capable of capturing the distinctive aspects of the two populations of participants, we have performed a set of classification experiments, as will be described in Section 6.

To mitigate as much as possible the bias on the performance due to the classifier, the experiments have been performed by using different classifiers, widely used in the literature: Decision Tree, Random Forest, Logistic Regression, K-Nearest Neighbor, Linear Discriminant Analysis, Gaussian Naive Bayes, Support Vector Machine, Multilayer Perceptron, and Learning Vector Quantization.

We chose these models since they: (i) are widely-used and standard implementations are available for each of them; (ii) represent different paradigms of classification algorithms; (iii) exhibit good performance on a large variety of classification problems. Table 3 shows their computation complexity ($\mathcal{O}$ notation), as well as the acronyms used in the remainder of the paper.

In order to make the paper self-consistent, the following subsections describe the main characteristics of each classifier, and provide the references for the readers interested in further details. The classifiers were implemented in Python, using the functionalities of the Scikit-Learn library (Pedregosa et al., 2011).[3]

Table 3
Acronyms and computational complexity ($\mathcal{O}$ notation) of the training phase of the classification models used. N and M represent the number of training samples and the number of features, respectively. As for the other quantities involved, they are represented as follows:
L: #trees making up the ensemble (RF);
E: #epochs (MLP and LVQ)
P: #neurons (MLP and LVQ).

Model	Acronym	Complexity
Random forest	RF	$LNMlog_2N$
Logistic regression	LR	NM
K-Nearest Neighbor	KNN	1
Linear Discriminant Analysis	LDA	NM^2
Gaussian Naive Bayes	GNB	NM
Support Vector Machine	SVM	N^2M
Decision Tree	DT	$NMlog_2N$
Multilayer Perceptron	MLP	$ENMP$
Learning Vector Quantization	LVQ	$ENMP$

5.1. Decision tree

A Decision Tree (DT) is a decision support tool with a tree graph structure (Breiman et al., 1984). In a DT, internal nodes represent attribute tests, where each branch yields the outcome of the test, whereas leaf nodes represent class labels. The paths from the root node to the leaves represent classification rules. For the tree learning, we used the C4.5 algorithm. This algorithm builds a decision tree with a top down approach, by using the concept of information entropy. Given a training set S, it breaks down S into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. At each node of the tree, C4.5 chooses the attribute that most effectively splits the corresponding subset. The splitting criterion is the normalized information entropy gain which measures how much the subsets are homogeneous, in terms of class labels, with respect to the split set. The algorithm then recurs on the smaller subsets. The algorithm creates a leaf node when one of the following base cases occur:

- all samples in the set belong to the same class. The leaf node is labeled with that class.
- the number of instances in the set is below a certain threshold. The leaf node is labeled with the more represented class in the set.
- None of the features provide any information gain. The leaf node is labeled with the most represented class in the set.

The pseudocode of C4.5 algorithm is the following:

1. Check for the above base cases.
2. For each attribute a, find the normalized information gain ratio from splitting on a.
3. Let a_b be the attribute with the highest normalized information gain.
4. Create a decision node that splits on a_b.
5. Recurse on the subset obtained by splitting on a_b, and add those nodes as children of the node.

5.2. Random forest

The Random Forest algorithm (RF) builds an ensemble of L tree-based classifiers combining two well-known strategies for inducing

[3] The code developed to implement the nine classification schemes used in the experiments detailed in Section 6 is available at the following GitHub repository: https://github.com/Natural-Computation-Lab/DARWIN-Dataset-Baseline-Performance.git.

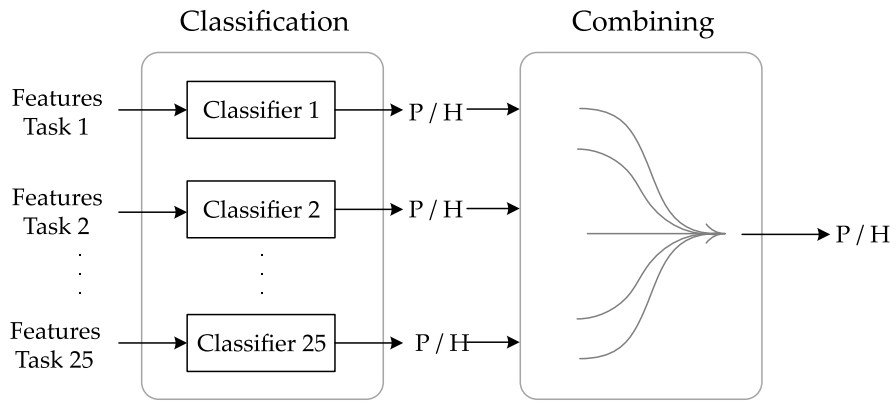


Fig. 6. Combined classifications for each handwriting task.

diversity in classifier ensembles (Breiman, 2001), namely bagging and random subspace. These strategies induce classifier diversity generating a different training dataset for each classifier making up the ensemble. Given a training set D consisting of N samples, each made of M features, RF builds the ith tree of the ensemble as follows:

1. Draw from D N samples at random with replacement (bagging strategy). The resulting set will be the training set of the tree.
2. Set a number $K \ll M$.
3. At each node, randomly draw K features from the set of available features (random subspace strategy).
4. Among the values of each of the K features drawn, choose the best binary split according to the Gini index. Select the feature with the best index value.
5. Grow the tree to its maximum size according to the stopping criterion chosen[4].
6. Let the tree unpruned.

5.3. Logistic regression

Logistic regression (LR) is a linear classification algorithm that uses the logistic function to model class probabilities (Yu et al., 2011). Given a training set of instance-label pairs $(\mathbf{x}_i, y_i)$ $i = \ldots, N$, where $\mathbf{x}_i \in \mathbb{R}^M$ and $y_i \in \{1, -1\}$, LR requires the solution of the following optimization problem:

$$\min_{\mathbf{w},c} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} log(exp(-y_i(\mathbf{x}_i^T\mathbf{w} + c)) + 1) \tag{1}$$

To solve this problem we used the lbfgs method. It approximates the Broyden–Fletcher–Goldfarb–Shanno algorithm (Fletcher, 1987), and it is an iterative method for solving unconstrained nonlinear optimization problems, recommended for small data-sets.

5.4. K-Nearest Neighbor

The K-Nearest Neighbor algorithm (KNN) is a well-known non parametric method that can be used for both classification and regression (Bishop, 2006). According to this approach, an unknown sample is labeled with the most common label among its k nearest neighbors in the training set. The rationale behind the k-NN classifier is that, given an unknown sample $\mathbf{x}$ to be assigned to one of the c_i classes of the problem at hand, the a-posteriori probabilities $p(c_i|\mathbf{x})$ of $\mathbf{x}$ may be estimated as follows:

$$p(c_i|\mathbf{x}) = n_i/K$$

where n_i is the number of samples among the K nearest neighbor of $\mathbf{x}$ belonging to the ith class.

[4] Node splitting usually is stopped when one of the following conditions occur: (i) The number of samples in the node to be split is below a given threshold; (ii) the samples in the node belong to the same class

5.5. Linear discriminant analysis

Linear discriminant analysis (LDA), is a generalization of the Fisher's linear discriminant method, which finds the linear combination of features that best characterizes or separates two or more classes (Duda et al., 2001). The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

Given a training set D of instance-label pairs $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^M$ and $y_i \in \{0, 1\}$, LDA assumes that the conditional probability density functions $p(\mathbf{x}|y = 0)$ and $p(\mathbf{x}|y = 1)$ are normally distributed with the same covariance matrix: $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma})$ and $(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$. LDA also assumes that $\boldsymbol{\Sigma}$ is full ranked. Under these assumptions, the Bayes optimal decision criterion to label an unseen sample $\mathbf{x}$ is given by the following equation:

$$\mathbf{w} \cdot \mathbf{x} > c$$

where

$$\begin{aligned} \mathbf{w} &= \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}) \\ c &= \mathbf{w} \cdot \frac{1}{2}(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1) \end{aligned} \tag{2}$$

From a geometrical perspective, $\mathbf{w}$ is a vector perpendicular to the hyperplane separating the two classes. The location of the hyperplane is defined by the threshold c

5.6. Support Vector Machines

Support Vector Machines (SVMs) are supervised learning methods based on the concept of decision planes (Chang and Lin, 2011). These planes linearly separates (in the feature space) objects belonging to different classes. Intuitively, given two classes to be discriminated in a given feature space, a good separation is achieved by the hyperplanes that have the largest distance to the nearest training points belonging to different classes; in general, the larger the margin, the lower the generalization error of the classifier.

While the basic idea of the SVM applies to linear classifiers, they can be easily adapted to non-linear classification tasks by using the so-called "kernel trick", which implies the mapping of the original features into a higher dimensional feature space. Given a training set D of instance-label pairs $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^M$ and $y_i \in \{1, -1\}$ SVMs require the solution of the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to } y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ &\qquad \xi_i \geq 0 \end{aligned} \tag{3}$$

According to the above formulation, training samples are mapped into a higher dimensional space by the function ϕ. SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_i)$ is called the *kernel function*. Typical kernels are:

linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$.
polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^d, \ \gamma > 0$.
radial basis function (RBF): $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2)$, $\gamma > 0$
sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j)^d$.

Here γ, r, and d are the kernel parameters.

5.7. BayesIan network

A Bayesian network (BN) is a directed acyclic graph that encodes a joint probability distribution over a set of random variables. A node in the graph represents a variable and an arc between two nodes encodes the conditional dependencies between them. The structure of a BN is useful to factorize the joint probability in the product of local terms, i.e. the conditional probabilities. Such a factorization makes it easier and faster to compute the joint probability. Furthermore, the conditional dependencies that come out from the graph are useful for further analysis. When we use a BN as a classifier we consider the label c and each attribute (feature) of the set $A = \{x_1, \ldots, x_M\}$ as a random variable. Once learned, given an unseen sample $\mathbf{x} \in \mathbb{R}^M$, a BN assigns to $\mathbf{x}$ the label $\hat{c}$ that maximizes the posterior probability $p(c|x_1, \ldots, x_M)$. Furthermore, known the conditional dependencies between the class and the attributes, the label $\hat{c}$ can be computed using the following equation:

$$\hat{c} = \arg\max_c p(c \mid pa_c) \prod_{x_i \in O} p(x_i \mid pa_{x_i})$$

where pa_c are is the set of attributes x_i linked to c with an incoming arc, whereas the set O contains the attributes linked with c with an outgoing arc from c.

5.8. Gaussian Naive Bayes

A Naive Bayes classifier is a Bayesian network, relying on two simplifying assumptions. The first assumes that the predictive attributes are conditionally independent given the class, whereas the second assumes that no hidden or latent attributes influence the prediction process. These assumptions imply that in a naive Bayesian classifier the only possible arcs are those directed from the class node to the attribute (feature) nodes. Therefore, once the network is learned, an unseen sample $\mathbf{x}$ is assigned to the class $\hat{c}$ that maximizes the posterior probability $p(c|x_1, \ldots, x_m)$, according to the following equation:

$$\hat{c} = \arg\max_c p(c) \prod_{x_i \in O} p(x_i \mid c)$$

A Gaussian Naive Bayes (GNB) classifier deals with continuous data, and is based on the assumption that the continuous values are normally distributed Hastie et al. (2009). In practice, for a continuous attribute x, average and variance are computed for each class. Let μ_k and σ_k^2 the mean and the variance of the values in x associated to the class c_k, then the probability density of a value v given a class c_k $p(x = v \mid c_k)$ can be computed using the equation for a normal distribution parameterized by, μ_k and σ_k^2:

$$p(x = v \mid c_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

5.9. Multilayer Perceptron

A Multilayer Perceptron (MLP) is an information processing system made up of a number of simple, highly interconnected processing elements called *neurons* (Rumelhart et al., 1986). The output of the ith neuron is the activation function of a weighted sum of its input:

$$y_i = f_a(w_0 + \sum_j^{n_i} w_j \cdot x_j)$$

typical activation functions are: sigmoid, hyperbolic tangent, Rectified Linear Unit (ReLU).

NN topologies are usually organized in *layers*. The patterns are presented to the network via the "input layer", whereas the final answer is provided through an "output layer". Once the network topology has been chosen, a NN must be trained by providing as input a set of labeled samples. We used a feed-forward completely connected network, trained by using the back-propagation algorithm (Goodfellow et al., 2016).

5.10. Learning vector quantization

Learning vector quantization (LVQ) is a prototype-based supervised classification algorithm (Kohonen, 1995). In a feature space of dimension M, the solution of a LVQ system is represented by a set of prototypes (neurons):

$$W = \{\mathbf{w}_1, \ldots, \mathbf{w}_P\}, \quad \mathbf{w}_i \in \mathbb{R}^M$$

The LVQ training algorithm, for each data point, determines its nearest prototype according to a given distance measure. The position of this so-called winner prototype is then adapted, i.e. the winner is moved nearer if it correctly classifies the data point or moved away if it classifies the data point incorrectly.

Given a training set $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}, \quad \mathbf{x}_i \in \mathbb{R}^M$, the LVQ algorithm can be outlined as follows:

begin
  Initialize the weights of the labeled neurons in:
  $W = \{(\mathbf{w}_1, c_1), \ldots, (\mathbf{w}_P, c_P)\}, \quad \mathbf{w}_i \in \mathbb{R}^M$
  for $j = 1$ to E **do**
    for $i = 1$ to N **do**
      find $\mathbf{w}_m$, the nearest neuron to $\mathbf{x}_i$
      if $(c_i == y_i)$ **then**
        $\mathbf{w}_m = \mathbf{w}_m + \eta \cdot (\mathbf{x}_i - \mathbf{w}_m)$
      else
        $\mathbf{w}_m = \mathbf{w}_m - \eta \cdot (\mathbf{x}_i - \mathbf{w}_m)$
end

The number of epochs E, the number of neurons P and the learning rate η are parameters of the algorithm.

6. Experimental results

As mentioned in the Introduction, we performed a large set of experiments to validate the rationale behind the design of the protocol we used to collect the data in the DARWIN dataset, as well as to assess the capability of the proposed feature set to allow us to effectively discriminate between AD patients and healthy people.

6.1. Experimental protocol

To reduce as much as possible the bias introduced by the randomness in selecting the samples of the training and test set, we performed twenty runs. In each run, the dataset was randomly shuffled and split into a training and a test set. Furthermore, to allow each classifier to work in its best configuration, before each training, we performed a 5-fold cross-validated grid search to select the best set of hyperparameters for the classifier. In practice, we defined a set of values for each parameter to be tuned, and then exhaustively tested all parameter

Table 4
Hyperparameters ranges explored during the grid search for each classifier. Omitted parameters are set to the default value as defined in the SciKit-Learn library.

Classifier	Parameter	min value	Max value	Step
RF	MaxDepth	3	10	1
	n_Estimators	100	300	50
	bootstrap	True	False	
	min_samples_split	2		
	min_samples_leave	1		
LR	C	0.001	5	0.005
KNN	b_neighbors	5	15	1
LDA	solver	svd		
GNB	priors	2		
	var_smoothing	1e−9		
SVM	kernel	RBF, Linear		
	C	0.5	1.5	0.1
	γ	0.5		
DT	criterion	gini, entropy		
	max_depth	2	10	1
	min_samples_split	2	5	1
	min_samples_leaf	2	20	2
	max_leaf_node	2	20	2
MLP	activation	relu, logistic, tanh		
	hidden_layer_size	8	20	1
	learning_rate_init	0.05	0.4	0.05
	max_iteration	1000		
	α	0.0001		
LVQ	prototype_for_classes	1	50	5
	β	2	50	5
	max_iter	2500		

combinations. Table 4 shows the ranges of the hyper-parameters tested. To provide an estimate of the training cost, Table 5 shows the average computing time for training one instance of the classifier, for evaluating the performance on the validation test during the training, and for classifying the test set once the training was completed. These times have been recorded during the baseline evaluation session.

A combination of the outputs of the best classifiers was also evaluated, as will be explained in the following (Sections 6.4 and 6.5). In this case, we trained and tested the combined classifiers on a new dataset partition, and repeated the 5-fold cross-validated grid search.

6.2. Baseline evaluation session

In the first experiment, we evaluated the performance of the classifiers by using as feature set the union of the features extracted from each of the 25 tasks. Table 6 shows the mean accuracy, the specificity and the sensitivity achieved by each classifier. For the sake of space, the table shows only the standard deviation of the mean accuracy, but similar values and distribution among the classifiers were observed for both specificity and sensitivity.

The mean accuracy values and their standard deviation shown in the table prove that all the classifiers performed well (mean accuracy >70%) and that the performance does not depend on the actual samples in the training and test set nor on the randomness in the training of the classifiers. Furthermore, a statistical analysis based on the Friedman test (Friedman, 1937) followed by a Nemenyi post-hoc test ($\alpha = 0.05$) (Nemenyi, 1963) has shown that the differences in the mean accuracy between the best classifiers, i.e. RF, LR, GNB and MLP, were not statistically significant. Overall, the results of this experiment suggest that, as a whole, the tasks we included in our test and the features we extracted from their execution provide relevant information for discriminating between AD patients and healthy people, independently of the adopted classifier, even though there are classification models that exploit to a greater extent than others the information carried by the features. They also show that the best classifiers achieved high sensitivity (>84%), confirming that the test can be very reliably detects AD patients.

6.3. Classification by task

In this experiment, the classifiers were trained and evaluated on 25 different feature sets, one for each task, leading to 25 task-specific classifiers for each classification model. Table 7 shows, for each classification model, the mean accuracy achieved on each task, whereas Table 8 shows the mean specificity and sensitivity. The latter table do not shows the standard deviations as their values and distribution among the classifiers do not differ from those shown in Table 7.

From the tables we can observe that, independently of the classifier, the accuracy achieved on any single task by any classifier was lower than that achieved by the same classifier in the previous experiment (see Table 6), with the exception of KNN (tasks #7 and #17), LDA (task #21) and DT (task #23), which exhibited slightly better performance than in the previous experiment. On the contrary, the best performing models achieving of the previous experiment gave worse performance on single tasks. These results confirm our assumption that different tasks elicit different aspects of handwriting movements, providing a more comprehensive characterization than any single task.

Even more interestingly, the results of this experiment demonstrate that no classifier performed best on all the tasks. Then to find the best classifier for each task, we performed the Friedman test ($\alpha = 0.05$) to test the null hypothesis that there were no statistically significant differences between the accuracy distributions of the classifiers. The null hypothesis was rejected in all 25 Friedman tests, confirming that for each task there was at least a pair of classifiers whose performance were significantly different. Then, the classifiers were sorted according to the average ranks calculated by the Friedman test. We then performed a Nemenyi post-hoc test ($\alpha = 0.05$) to compare the pairs of classifiers. The post-hoc analysis revealed that for each task, the classifier with the highest rank had a performance that was not significantly different from the performance of some other classifiers. These results confirmed that for each task there was a group of best performing models. The best model for a task was then selected by ranking the models belonging to the group of the best performing models according to their mean sensitivity. The 25 best classification, listed from the first to the last task, were the following: RF, LR, LR, RF, RF, RF, LR, LDA, RF, RF, GNB, LR, RF, RF, RF, DT, RF, MLP, RF, RF, RF, RF, DT, LR, RF.

6.4. Combining all

To obtain the final classification for each participant, and in contrast with the first experiment in which we merged the feature sets, in this experiment we combined the outputs of the 25 task-specific classifiers using the majority vote decision rule (Kittler et al., 1998). We combined them in two ways: in the first, we combined the outputs of the 25 task-specific classifiers generated by a given classification model; in the second, we combined the outputs of the 25 best classifiers we obtained from the previous experiment. In the following, we will refer to the first nine systems by the acronym of the classification model (see Table 3) used to generate the single-task classifiers, whereas we will refer to the last implementation as BFT. The results achieved by these systems are shown in Table 9.

From the table we can observe that all multiclassifiers achieved good performance, providing further support to our assumption that the tasks we have designed elicit different aspects of handwriting, and that their combination captures the distinctive aspects of the two groups (AD patients and healthy people) better than any single task. Furthermore, looking at the results in Tables 6 and 9 we can see that for all classification models (except GNB) each multiclassifier performed better than the corresponding baseline classifier in terms of overall accuracy. In particular, the *DT* multiclassifier achieved the highest accuracy, specificity and sensitivity, and the largest improvement with respect to the baseline classifier. Finally, we can observe that the *BFT* multiclassifier ranked the second best in terms of overall accuracy and specificity.

Table 5
Average training and testing times, expressed in seconds.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	LVQ
Time for training (No grid search)	0.3601	0.0219	0.0035	0.0648	0.0048	0.0122	0.0192	0.5725	1.6603
Time to evaluate the training set	0.0288	0.0043	0.0103	0.0040	0.0044	0.0047	0.0037	0.0067	0.0043
Time to evaluate the TestSet	0.0222	0.0018	0.0033	0.0012	0.0015	0.0014	0.0010	0.0023	0.0013

Table 6
Mean accuracy (and standard deviation), specificity, and sensitivity (expressed in percentage) of the baseline classifiers.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	LVQ
Accuracy	88.29 (±4.90)	81.86 (±7.20)	71.43 (±8.34)	72.14 (±8.44)	85.00 (±5.47)	79.00 (±7.55)	78.57 (±7.21)	83.14 (±7.97)	77.43 (±7.41)
Specificity	86.18	79.41	89.41	72.65	79.12	80.59	74.41	81.76	87.35
Sensitivity	90.28	84.17	54.44	71.67	90.56	77.50	82.50	84.44	68.06

Table 7
Mean accuracy (expressed in percentage) achieved by the classifiers on each task.

Task #	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	LVQ
1	65.86 (±8.22)	62.86 (±7.32)	49.47 (±9.81)	60.75 (±5.37)	63.16 (±10.69)	62.26 (±7.29)	57.44 (±8.83)	61.95 (±9.33)	62.11 (±7.61)
2	67.14 (±7.78)	62.57 (±5.93)	55.29 (±7.43)	64.71 (±5.58)	58.71 (±6.84)	60.86 (±6.75)	62.86 (±9.98)	60.14 (±6.18)	62.43 (±7.72)
3	66.57 (±8.90)	67.86 (±8.93)	51.86 (±9.19)	67.00 (±6.72)	66.29 (±8.37)	68.00 (±7.95)	62.71 (±7.84)	65.57 (±7.73)	65.29 (±7.20)
4	71.29 (±6.78)	63.57 (±8.02)	64.71 (±5.58)	61.57 (±7.21)	58.57 (±3.99)	61.43 (±7.95)	66.57 (±7.42)	71.86 (±6.77)	68.14 (±5.66)
5	72.14 (±6.35)	73.71 (±6.85)	65.86 (±5.90)	67.57 (±5.81)	71.14 (±5.24)	71.14 (±7.35)	73.71 (±7.28)	69.29 (±6.07)	71.00 (±5.10)
6	72.43 (±7.66)	74.00 (±6.81)	65.57 (±10.22)	68.57 (±6.15)	73.29 (±5.66)	74.71 (±5.73)	74.86 (±7.22)	73.14 (±6.52)	73.43 (±6.94)
7	78.00 (±7.54)	72.00 (±6.40)	72.00 (±5.76)	70.57 (±7.19)	69.57 (±6.37)	71.29 (±5.75)	71.00 (±8.41)	72.29 (±7.00)	73.29 (±7.08)
8	64.86 (±6.36)	68.71 (±8.00)	57.00 (±8.57)	69.00 (±7.43)	57.00 (±5.97)	69.86 (±8.21)	61.14 (±6.52)	71.71 (±8.18)	67.57 (±7.20)
9	77.43 (±7.69)	71.43 (±5.79)	74.86 (±5.83)	72.00 (±5.29)	55.29 (±4.07)	72.43 (±6.89)	69.29 (±6.99)	68.57 (±8.84)	72.43 (±8.66)
10	69.29 (±6.80)	68.57 (±7.81)	58.14 (±8.95)	65.71 (±6.81)	66.86 (±6.90)	67.86 (±7.91)	60.14 (±6.97)	64.14 (±9.57)	65.00 (±5.40)
11	64.86 (±6.23)	64.71 (±8.31)	53.71 (±6.12)	62.14 (±7.58)	62.00 (±5.65)	62.71 (±8.05)	62.43 (±9.65)	62.57 (±7.97)	61.43 (±7.95)
12	67.14 (±7.78)	62.57 (±5.93)	55.29 (±7.43)	64.71 (±5.58)	58.71 (±6.84)	60.86 (±6.75)	62.86 (±9.98)	60.14 (±6.18)	62.43 (±7.72)
13	66.57 (±8.90)	67.86 (±8.93)	51.86 (±9.19)	67.00 (±6.72)	66.29 (±8.37)	68.00 (±7.95)	62.71 (±7.84)	65.57 (±7.73)	65.29 (±7.20)
14	71.29 (±6.78)	63.57 (±8.02)	64.71 (±5.58)	61.57 (±7.21)	58.57 (±3.99)	61.43 (±7.95)	66.57 (±7.42)	71.86 (±6.77)	68.14 (±5.66)
15	72.14 (±6.35)	73.71 (±6.85)	65.86 (±5.90)	67.57 (±5.81)	71.14 (±5.24)	71.14 (±7.35)	73.71 (±7.28)	69.29 (±6.07)	71.00 (±5.10)
16	72.43 (±7.66)	74.00 (±6.81)	65.57 (±10.22)	68.57 (±6.15)	73.29 (±5.66)	74.71 (±5.73)	74.86 (±7.22)	73.14 (±6.52)	73.43 (±6.94)
17	78.00 (±7.54)	72.00 (±6.40)	72.00 (±5.76)	70.57 (±7.19)	69.57 (±6.37)	71.29 (±5.75)	71.00 (±8.41)	72.29 (±7.00)	73.29 (±7.08)
18	64.86 (±6.36)	68.71 (±8.00)	57.00 (±8.57)	69.00 (±7.43)	57.00 (±5.97)	69.86 (±8.21)	61.14 (±6.52)	71.71 (±8.18)	67.57 (±7.20)
19	77.43 (±7.69)	71.43 (±5.79)	74.86 (±5.83)	72.00 (±5.29)	55.29 (±4.07)	72.43 (±6.89)	69.29 (±6.99)	68.57 (±8.84)	72.43 (±8.66)
20	71.43 (±8.03)	66.43 (±6.41)	60.29 (±7.47)	67.86 (±6.41)	64.57 (±4.58)	67.29 (±7.09)	65.86 (±7.73)	65.57 (±6.39)	68.14 (±6.44)
21	72.29 (±6.56)	72.86 (±7.21)	64.43 (±7.21)	72.29 (±6.29)	67.29 (±6.52)	71.14 (±7.35)	66.00 (±7.64)	72.29 (±8.65)	69.71 (±7.38)
22	75.00 (±7.80)	67.29 (±9.15)	68.43 (±6.04)	70.00 (±6.84)	67.14 (±6.84)	68.57 (±9.63)	69.14 (±6.53)	67.29 (±6.90)	69.86 (±7.27)
23	80.00 (±5.48)	74.29 (±7.64)	68.71 (±7.21)	70.43 (±6.77)	69.14 (±6.91)	74.43 (±6.39)	82.00 (±6.75)	70.86 (±6.66)	73.00 (±6.11)
24	72.14 (±4.98)	71.43 (±5.64)	61.29 (±6.25)	67.57 (±6.51)	66.86 (±5.89)	69.43 (±6.23)	67.14 (±5.83)	69.14 (±5.91)	72.57 (±6.11)
25	73.71 (±7.79)	71.71 (±7.35)	68.43 (±6.78)	71.29 (±6.84)	67.57 (±7.26)	70.43 (±6.77)	68.86 (±7.58)	68.71 (±8.87)	68.29 (±5.93)

Table 8
Specificity (*Sp*) and sensitivity (*Se*) (expressed in percentage) achieved by the classifiers on each task.

Task #	RF		LR		KNN		LDA		GNB		SVM		DT		MLP		LVQ	
	Sp	*Se*	*Sp*	*Se*	*Sp*	*Se*	*Sp*	*Se*	*Sp*	*Se*	*Sp*	*Se*	*Sp*	*Se*	*Sp*	*Se*	*Sp*	*Se*
1	67.65	63.89	63.24	61.94	59.12	40.56	63.82	57.50	76.47	50.83	67.35	57.22	60.88	55.28	63.82	60.00	67.94	56.94
2	72.94	65.28	71.76	72.50	70.59	61.39	74.71	67.22	90.59	51.11	76.18	70.28	65.29	69.44	68.53	70.00	67.94	66.94
3	75.00	61.67	72.06	67.22	59.41	61.67	75.29	65.56	84.71	54.17	72.94	64.44	77.94	61.94	67.35	64.72	69.41	64.44
4	72.35	66.67	77.65	59.17	72.06	61.11	77.35	53.33	88.82	51.94	84.71	55.56	67.65	59.72	65.00	61.67	73.82	60.83
5	68.53	72.22	78.24	61.11	71.18	60.00	80.29	56.11	90.59	56.39	83.53	50.00	62.94	68.33	67.65	65.00	72.94	59.72
6	76.47	66.94	81.47	65.00	75.59	54.72	77.65	58.06	91.47	52.50	84.12	57.22	78.24	63.89	77.35	61.11	80.59	58.61
7	74.71	70.83	74.41	75.28	66.76	59.17	74.41	76.39	71.47	71.94	77.35	73.89	71.76	66.94	73.53	72.22	73.82	68.06
8	75.59	67.50	80.88	68.89	67.65	51.94	77.06	69.72	90.29	54.72	83.53	68.33	73.24	60.56	75.88	65.00	85.00	63.61
9	78.82	70.83	79.41	65.00	63.82	63.06	78.53	65.56	90.88	55.56	86.47	61.39	73.24	62.22	72.35	67.50	80.00	59.17
10	69.12	69.44	76.18	61.39	72.94	44.17	75.29	56.67	91.18	43.89	80.00	56.39	67.06	53.61	61.76	66.39	71.18	59.17
11	62.65	66.94	68.82	60.83	66.47	41.67	70.88	53.89	50.00	73.33	71.18	54.72	54.41	70.00	63.53	61.67	62.65	60.28
12	78.24	56.67	63.82	61.39	58.24	52.50	68.82	60.83	60.00	57.50	66.18	55.83	67.35	58.61	62.35	58.06	68.53	56.67
13	71.47	61.94	79.12	57.22	59.12	45.00	73.53	60.83	84.71	48.89	80.88	55.83	66.76	58.89	70.88	60.56	74.12	56.94
14	72.65	70.00	67.94	59.44	72.65	57.22	65.29	58.06	44.41	71.94	63.53	59.44	70.00	63.33	74.71	69.17	73.82	62.78
15	75.00	69.44	79.71	68.06	80.59	51.94	72.94	62.50	90.59	52.78	76.76	65.83	78.53	69.17	69.71	68.89	74.12	68.06
16	77.35	67.78	80.29	68.06	67.65	63.61	81.18	56.67	87.06	60.28	82.35	67.50	77.06	72.78	73.53	72.78	76.47	70.56
17	79.41	76.67	75.59	68.61	73.82	70.28	70.29	70.83	80.29	59.44	74.71	68.06	69.12	72.78	71.47	73.06	77.65	69.17
18	64.71	65.00	71.18	66.39	58.24	55.83	72.35	65.83	23.82	88.33	72.35	67.50	62.06	60.28	72.06	71.39	69.12	66.11
19	80.88	74.17	70.59	72.22	79.71	70.28	73.82	70.28	12.06	96.11	76.76	68.33	71.47	67.22	75.29	62.22	77.94	67.22
20	74.12	68.89	73.53	59.72	75.88	45.56	74.12	61.94	92.06	38.61	78.24	56.94	63.53	68.06	65.88	65.28	74.41	62.22
21	67.94	76.39	72.94	72.78	59.41	69.17	75.00	69.72	86.47	49.17	73.82	68.61	63.24	68.61	70.88	73.61	68.82	70.56
22	74.71	75.28	69.41	65.28	69.71	67.22	74.71	65.56	91.76	43.89	73.53	63.89	65.59	72.50	64.41	70.00	77.06	63.06
23	77.06	82.78	75.29	73.33	68.24	69.17	75.29	65.83	77.35	61.39	77.35	71.67	76.18	87.50	72.94	68.89	72.65	73.33
24	76.76	67.78	74.41	68.61	66.18	56.67	70.88	64.44	85.88	48.89	77.94	61.39	70.29	64.17	73.82	64.72	77.35	68.06
25	73.24	74.17	75.29	68.33	73.24	63.89	77.06	65.83	88.53	47.78	76.47	64.72	69.41	68.33	74.71	63.06	71.47	65.28

Table 9
Accuracy, specificity, and sensitivity (expressed in percentage) achieved by combining the task specific classifiers.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	LVQ	BFT
Accuracy	88.57	85.71	85.71	77.14	82.86	88.57	94.29	88.57	82.86	91.43
Specificity	82.35	88.24	94.12	88.24	94.12	94.12	94.12	88.24	88.24	88.24
Sensitivity	94.44	83.33	77.78	94.34	77.22	83.33	94.44	88.89	77.78	94.44

Table 10
Accuracy, and number of selected tasks of the multiclassifiers combining the top basic classifiers.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	LVQ	BFT
Accuracy	88.57	94.29	91.43	94.29	85.71	94.28	94.29	88.57	91.43	91.43
Specificity	82.35	88.24	88.24	100	94.12	88.24	94.12	82.35	88.24	88.24
Sensitivity	94.44	88.89	94.44	88.89	77.78	100	94.44	94.44	94.44	94.44
# of classifiers	5	5	11	6	9	5	25	15	9	8

Table 11
Mean accuracy (and standard deviation), specificity, and sensitivity (expressed in percentage) achieved using a single classifier merging the features from the best tasks.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	LVQ
Accuracy	85.29 (±6.03)	83.86 (±4.57)	77.29 (±7.15)	61.43 (±8.32)	85.14 (±5.53)	81.86 (±4.57)	78.57 (±7.21)	82.71 (±6.52)	82.29 (±4.41)
Specificity	82.35	84.41	89.12	38.00	86.76	83.24	74.41	82.35	85.88
Sensitivity	88.06	83.33	68.00	66.88	83.61	80.56	82.50	83.06	78.89

6.5. Combining the best

The results of the second experiment (Section 6.3) proved that the classification models implemented achieved different performance on different tasks. To evaluate to what extent each task contributed to characterizing the handwriting of people affected by AD, we sorted the tasks according to the average ranks provided by the Friedman test for each classification model. Starting with the top ranked one, the remaining tasks were selected according to their ranking, and the corresponding classifiers were added to the set of classifiers to be combined. The plots in Fig. 7 shows the accuracy as function of the number of tasks for RF and BFT. Similar trends have been observed for the remaining classifiers, but they are not shown for the sake of space. From the plots we can observe that the best performance was achieved by combining the responses of a small number of the single tasks.

Table 10 shows, for each multiclassifier system, the number of selected tasks leading to the highest accuracy, and the corresponding specificity and sensitivity achieved. These results proved that all systems achieved the best performance by combining the outputs of at most fifteen models, with the more parsimonious ones (RF, LR and SVM) combining only five models. The only exception was the DT multiclassifier that combined all models. It is worth noting that none of the multiclassifiers achieving the highest accuracy was the best in terms of specificity and sensitivity. It is however remarkable that the SVM multiclassifier achieved 94.28% accuracy, 88.24% specificity and 100% sensitivity by using only five tasks.

Finally, as further combination strategy of the information from the best performing tasks, we used an approach similar to that described in Section 6.2. In practice, we merged the features extracted from the subset of tasks leading to the best performance into a single feature vector and used it to train and test each model. Table 11 shows the mean accuracy, specificity, and sensitivity achieved by each model. Comparing the results in Tables 6 and 11 we can see that combining only the best feature sets was beneficial for some classifiers and detrimental for others, but in all cases the difference in terms of mean accuracy was smaller than 6%, with the only exception being the LDA classifier which gave a reduction of the mean accuracy of 10.71% in comparison to the baseline classifier. Furthermore, the comparison between Tables 10 and 11 proved that merging the feature sets did not lead to better performance with respect to those obtained by combining the outputs of the task-specific classifiers.

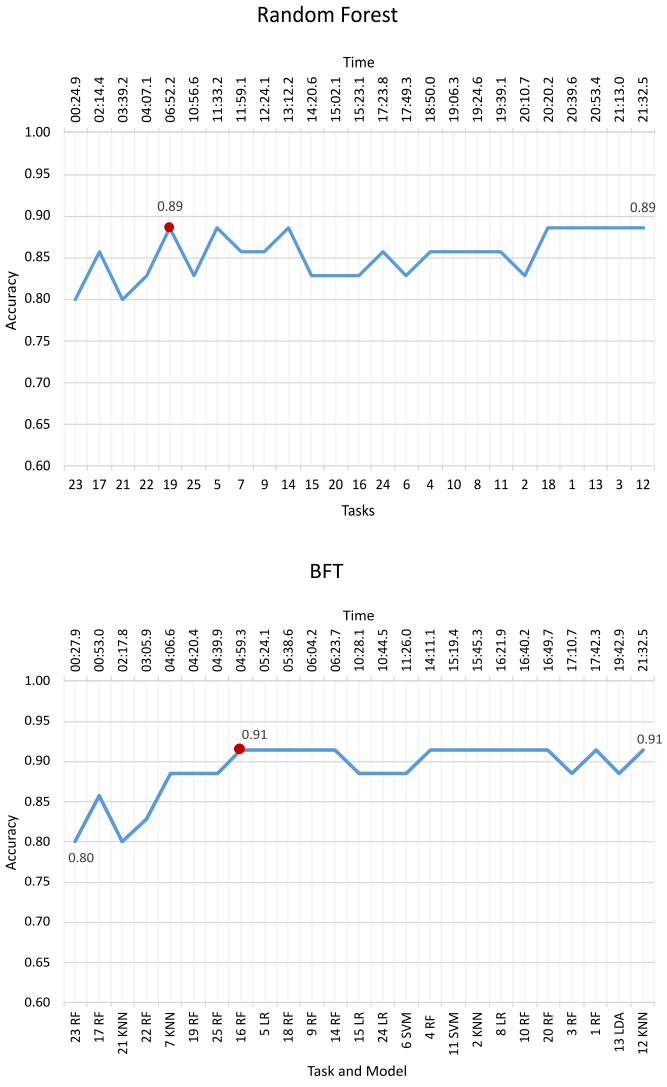


Fig. 7. Accuracy achieved by incrementally combining classification results. The ordered tasks are shown on the abscissa, whereas the average total time required to execute the tasks is shown on top of the plot.

7. Conclusions

The aim of the work reported in this paper was to show that the analysis of handwriting is an innovative way of characterizing the effects of AD. While there is a consistent body of literature exploiting handwriting analysis for building automatic systems supporting physicians in the early diagnosis of PD, the large majority of works on automatic diagnosis of AD exploits neuroimaging as the main source of data, making the procedure for data acquisition expensive, invasive and time-consuming. On the contrary, we believe that handwriting may offer a cheap, non invasive and easy-to-administer test to acquire data that can be exploited to extract features useful to train machine learning tools to help physicians diagnose earlier AD and for monitor the effects of the treatments, pursuing patient-centered care for improving the patients' quality of life and reducing the social cost of their impairments.

To achieve our purpose, and recognizing that there is a lack of public datasets to support the research community, we collected data from 89 AD patients, at various stages of the disease, and 85 healthy people, to build the DARWIN dataset. The dataset was collected by adopting a protocol we have developed that comprises 25 different tasks, that were suggested as the most appropriate to elicit different aspects of the diseases. Each sample of the dataset, i.e. the movements performed during the execution of a task, is described by a set of 18 features, including those adopted by other researchers for PD and others we have derived from studies in neuroscience. The number of participants involved and the number of tasks performed, make the DARWIN dataset the largest currently publicly available. It was conceived and collected with the aim of allowing other researchers to develop and test machine learning-based systems to support the diagnosis of AD from handwriting, and provide a common ground for a fair comparison of the developed methods with the state-of-the-art solutions we have presented here.

To ascertain to what extent the dataset contains useful information to develop decision support system based on machine learning methodologies to help physicians diagnosing as early as possible the insurgence of AD, we evaluated the performance of nine different classification models, selected among the top performing and most widely used in the field. We have also implemented twenty multiclassifier systems, varying the classification model adopted to build the basic classifiers and using different strategies to build the pool of classifiers to be combined.

The results in Sections 6.2 and 6.3 prove that using a feature vector made of the 18 features extracted from the 25 tasks, totaling 450 features, led to better performance than those achieved by using as feature vector the 18 features extracted from any given task, with a few exceptions. Furthermore, statistical analysis has confirmed that: (i) there is a pool of classifiers whose performance are significantly better than those of the other ones; (ii) for each task, there is a classifier that performs better than the others. All together, these results support our hypothesis that including different tasks in the test allow us to gather information that characterize the handwriting of AD patients better than any single task. Furthermore, these results prove that the classifiers we have developed, with the exception of KNN and LDA, achieved a mean accuracy between 78.57% and 88.29%, confirming that the set of features we have adopted, as a whole, is capable of capturing the distinctive aspects of handwriting to effectively discriminate between AD patients and healthy people.

As it turned out that different classification models achieved different performance on different tasks, we designed and performed a second set of experiments, by building, for each classification model, as many models as the number of tasks, each working with the vector of 18 features extracted from a single task, and eventually combining their outputs by majority voting. We also built the BFT multiclassifier system by selecting for each task the top performing classifier and then combining their outputs as in the previous case. The results of these experiments have shown that the multiclassifiers built by combining the outputs of the 25 basic classifiers outperform the baseline classifier they are based upon in terms of mean accuracy, except GNB, whose mean accuracy drops from 85.00% to 82.76%. Six of them, namely the KNN, LDA, SVM, DT, MLP and LVQ multiclassifiers improved the performance also in terms of specificity and sensitivity. The results of this experiment also show that the BFT multiclassifier achieved similar performance than those exhibited by the top performing multiclassifier built by using the same classification model on each task. All together, and from a different perspective, these results provide a further support to our hypothesis that the different tasks included in our protocol allow us to gather information that characterizes the handwriting of people affected by AD better than a single task, independently of the classifier. Furthermore, our results suggest that the best way to exploit the information brought by the feature sets extracted from different tasks is that of combining the results of the classification on each feature set rather than combine the feature sets into a single one and build a single classifier.

Finally, we investigated the relevance of each task on the final output. For this purpose, for each classification model we ranked the tasks according to the accuracy achieved by the model on the corresponding feature set, and then incrementally added the outputs of the model to the outputs to be combined by majority voting, so as to find the smallest set of tasks whose corresponding models need to be combined to achieve the best performance. These results show that the multiclassifiers exploiting a subset of the tasks achieved the same or better performance with respect to the multiclassifiers using all tasks, with many of them achieving more than 90% in terms of mean accuracy, specificity and sensitivity. At last, but not least, for each basic classification model we merged the feature extracted from the top performing tasks and used them to train the model. Overall, the results of these experiments suggest that it is possible to design simpler test to be administered, consequently reducing the amount of time required to carry out the test up to more than 50%.

From an application point of view, we believe that the combination of the performance achieved by the top performing multiclassifier systems in terms of accuracy and sensitivity, together with the short time needed to execute the test, may eventually favor the adoption of the test by physicians, particularly family doctors and neurologists, for an early diagnosis of AD. From this point of view, we consider that the multiclassifier combining the five top performing SVM-based classifier exhibited the best trade-off between performance, time for executing the test and the total time for training the system. On the other hand, as it has been observed that the interpretability of the criteria used by AI-based systems to reach a decision is a factor of paramount importance to determine the acceptance by doctors, the DT-based multiclassifier may represent a better alternative, but it involves a larger number of tasks, leading to longer times for executing the test.

It is worth noting that we did not report any comparison with the performance presented in Werner et al. (2006), Pirlo et al. (2015a), Garre-Olmo et al. (2017), Kahindo et al. (2018) and Ishikawa et al. (2019) as they were obtained on different datasets, so that a direct comparison is meaningless. On the contrary, the experiments reported in Cilia et al. (2019b,c,a) used the same dataset, although with different features/classifiers and fewer tasks. The results reported in those papers show that our method outperforms any combination of features/classifiers used in those studies.

Although the performance achieved in our experiments compares favorably with the state of the art, there is still room for improvement. For instance, as an alternative to the task selection procedure we have presented here, we will use feature selection techniques applied to the feature set of each task, to further improve the performance of our approach. We will also develop stacking-based approaches to combine the responses provided by the models trained on each single task, for achieving a more accurate final prediction of the cognitive state of the people being analyzed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Albu, A., Precup, R., Teban, T., 2019. Results and challenges of artificial neural networks used for decision making and control in medical applications. Facta Univ. Ser.: Mech. Eng. 17 (3).

Armstrong, M.J., Litvan, I., Lang, A.E., Bak, T.H., Bhatia, K.P., Borroni, B., Boxer, A.L., Dickson, D.W., Grossman, M., Hallett, M., et al., 2013. Criteria for the diagnosis of corticobasal degeneration. Neurology 80 (5), 496–503.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

Borlea, I.-D., Precup, R.-E., Borlea, A.-B., Iercan, D., 2021. A unified form of fuzzy C-means and K-means algorithms and its partitional implementation. Knowl.-Based Syst. 214, 106731.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and Regression Trees. CRC Press.

Cavaliere, F., Della Cioppa, A., Marcelli, A., Parziale, A., Senatore, R., 2020. Parkinson's disease diagnosis: Towards grammar-based explainable artificial intelligence. In: 2020 IEEE Symposium on Computers and Communications (ISCC). IEEE, pp. 1–6.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) 2, 27:1–27:27.

Cilia, N.D., De Stefano, C., Fontanella, F., Di Freca, A.S., 2018. An experimental protocol to support cognitive impairment diagnosis by using handwriting analysis. Procedia Comput. Sci. 141, 466–471.

Cilia, N.D., De Stefano, C., Fontanella, F., di Freca, A.S., 2019a. How word choice affects cognitive impairment detection by handwriting analysis: A preliminary study. In: Italian Workshop on Artificial Life and Evolutionary Computation. Springer, pp. 113–123.

Cilia, N.D., De Stefano, C., Fontanella, F., Molinara, M., Di Freca, A.S., 2019b. Handwriting analysis to support alzheimer's disease diagnosis: a preliminary study. In: International Conference on Computer Analysis of Images and Patterns. Springer, pp. 143–151.

Cilia, N.D., De Stefano, C., Fontanella, F., Molinara, M., Di Freca, A.S., 2019c. Using handwriting features to characterize cognitive impairment. In: International Conference on Image Analysis and Processing. Springer, pp. 683–693.

De Gregorio, G., Desiato, D., Marcelli, A., Polese, G., 2021. A multi classifier approach for supporting alzheimer's diagnosis based on handwriting analysis. In: Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R. (Eds.), Pattern Recognition. ICPR International Workshops and Challenges. Springer International Publishing, Cham, pp. 559–574.

Drotár, P., Mekyska, J., Rektorová, I., Masarová, L., Smékal, Z., Faundez-Zanuy, M., 2016. Evaluation of handwriting kinematics and pressure for differential diagnosis of parkinson's disease. Artif. Intell. Med. 67, 39–46.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification, second ed. Wiley, New York.

Fletcher, R., 1987. Practical Methods of Optimization, second ed. John Wiley & Sons, New York, NY, USA.

Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Amer. Statist. Assoc. 32 (200), 675–701.

Garre-Olmo, J., Faúndez-Zanuy, M., López-de Ipiña, K., Calvó-Perxas, L., Turró-Garriga, O., 2017. Kinematic and pressure features of handwriting and drawing: preliminary results between patients with mild cognitive impairment, alzheimer disease and healthy controls. Curr. Alzheimer Res. 14 (9), 960–968.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, pp. 200–220, http://www.deeplearningbook.org.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer.

Impedovo, D., Pirlo, G., Vessio, G., 2018. Dynamic handwriting analysis for supporting earlier parkinson's disease diagnosis. Information 9 (10), 247.

Isenkul, M., Sakar, B., Kursun, O., 2014. Improved spiral test using digitized graphics tablet for monitoring Parkinson's disease. In: Proc. of the Int'L Conf. on E-Health and Telemedicine, pp. 171–175.

Ishikawa, T., Nemoto, M., Nemoto, K., Takeuchi, T., Numata, Y., Watanabe, R., Tsukada, E., Ota, M., Higashi, S., Arai, T., et al., 2019. Handwriting features of multiple drawing tests for early detection of alzheimer's disease: A preliminary result. In: MedInfo. pp. 168–172.

Jain, N., Virmani, D., Abraham, A., 2019. Proficient 3-class classification model for confident overlap value based fuzzified aquatic information extracted tsunami prediction. Intell. Decis. Technol. 13 (3), 295–303.

Kahindo, C., El-Yacoubi, M.A., Garcia-Salicetti, S., Rigaud, A.-S., Cristancho-Lacroix, V., 2018. Characterizing early-stage alzheimer through spatiotemporal dynamics of handwriting. IEEE Signal Process. Lett. 25 (8), 1136–1140.

Kalia, L., Lang, A., 2015. Parkinson's diseases. Lancet 386, 896–912.

Kittler, J., Hatef, M., Duin, R.P., Matas, J., 1998. On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20 (3), 226–239.

Kohonen, T., 1995. Learning vector quantization. In: Self-Organizing Maps. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 175–189.

Myszczynska, M.A., Ojamies, P.N., Lacoste, A.M.B., Neil, D., Saffari, A., Mead, R., Hautbergue, G.M., Holbrook, J.D., Ferraiuolo, L., 2020a. Applications of machine learning to diagnosis and treatment of neurodegenerative diseases. Nat. Rev. Neurol. 16, 440–456.

Myszczynska, M.A., Ojamies, P.N., Lacoste, A.M., Neil, D., Saffari, A., Mead, R., Hautbergue, G.M., Holbrook, J.D., Ferraiuolo, L., 2020b. Applications of machine learning to diagnosis and treatment of neurodegenerative diseases. Nat. Rev. Neurol. 1–17.

Nemenyi, P.B., 1963. Distribution-Free Multiple Comparisons. Princeton University.

P. Drotár, P., Mekyska, J., Rektorová, I., Masarová, L., Smékal, Z., Faundez-Zanuy, M., 2013. A new modality for quantitative evaluation of parkinson's disease: In-air movement. In: 13th International Conference on Bioinformatics and Bioengineering.

Parziale, A., Della Cioppa, A., Senatore, R., Marcelli, A., 2019. A decision tree for automatic diagnosis of parkinson's disease from offline drawing samples: experiments and findings. In: International Conference on Image Analysis and Processing. Springer, pp. 196–206.

Parziale, A., Senatore, R., Della Cioppa, A., Marcelli, A., 2021. Cartesian genetic programming for diagnosis of parkinson disease through handwriting analysis: Performance vs. interpretability issues. Artif. Intell. Med. 111, 101984.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Pereira, C.R., Pereira, D.R., Da Silva, F.A., Hook, C., Weber, S.A.T., Pereira, L.A.M., Papa, J.P., 2015. A step towards the automated diagnosis of parkinson's disease: Analyzing handwriting movements. In: 2015 IEEE 28th International Symposium on Computer-Based Medical Systems. pp. 171–176.

Pereira, C.R., Pereira, D.R., Silva, F.A., Masieiro, J.P., Weber, S.A.T., Hook, C., Papa, J.P., 2016a. A new computer vision-based approach to aid the diagnosis of parkinson's disease. Comput. Methods Programs Biomed. 136, 79–88.

Pereira, C.R., Pereira, D.R., Weber, S.A., Hook, C., de Albuquerque, V.H.C., ao P. Papa, J., 2019. A survey on computer-assisted parkinson's disease diagnosis. Artif. Intell. Med. 95, 48–63.

Pereira, C.R., Weber, S.A.T., Hook, C., Rosa, G.H., Papa, J.P., 2016b. Deep learning-aided parkinson's disease diagnosis from handwritten dynamics. In: Proceedings of the SIBGRAPI 2016 - Conference on Graphics, Patterns and Images, pp. 340–346.

Pirlo, G., Diaz, M., Ferrer, M.A., Impedovo, D., Occhionero, F., Zurlo, U., 2015a. Early diagnosis of neurodegenerative diseases by handwritten signature analysis. In: International Conference on Image Analysis and Processing. Springer, pp. 290–297.

Pirlo, G., Diaz-Cabrera, M., Ferrer, M., Impedovo, D., Occhionero, F., Zurlo, U., 2015b. Early diagnosis of neurodegenerative diseases by handwritten signature analysis. In: International Conference on Image Analysis and Processing. pp. 290–297.

Pozna, C., Precup, R., 2014. Applications of signatures to expert systems modelling. Acta Polytech. Hung. 11 (2), 2014.

Precup, R.-E., Teban, T.-A., Albu, A., Borlea, A.-B., Zamfirache, I.A., Petriu, E.M., 2020. Evolving fuzzy models for prosthetic hand myoelectric-based control. IEEE Trans. Instrum. Meas. 69 (7), 4625–4636.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323 (6088), 533–536.

Senatore, R., Marcelli, A., 2019a. A paradigm for emulating the early learning stage of handwriting: Performance comparison between healthy controls and parkinson's disease patients in drawing loop shapes. Hum. Mov. Sci. 65, 89–101.

Senatore, R., Marcelli, A., 2019b. A paradigm for emulating the early learning stage of handwriting: Performance comparison between healthy controls and parkinson's disease patients in drawing loop shapes. Hum. Mov. Sci. 65, 89–101.

Tanveer, M., Richhariya, B., Khan, R.U., Rashid, A.H., Khanna, P., Prasad, M., Lin, C.T., 2020. Machine learning techniques for the diagnosis of alzheimer's disease: A review. ACM Trans. Multimedia Comput. Commun. Appl. 16 (1s).

Vessio, G., 2019. Dynamic handwriting analysis for neurodegenerative disease assessment: A literary review. Appl. Sci. 9 (21), 4666.

Werner, P., Rosenblum, S., Bar-On, G., Heinik, J., Korczyn, A., 2006. Handwriting process variables discriminating mild alzheimer's disease and mild cognitive impairment. J. Gerontol. Ser. B: Psychol. Sci. Soc. Sci. 61 (4), P228–P236.

Yu, H., Huang, F., Lin, C., 2011. Dual coordinate descent methods for logistic regression and maximum entropy models. Mach. Learn. 85 (1–2), 41–75.