**ORIGINAL RESEARCH**

# Impact of the characteristics of data sets on incremental learning

**Patrick Marques Ciarelli, Elias Oliveira, Evandro O. T. Salles**

1. Department of Industrial Technology, Universidade Federal do Espírito Santo, Brazil. 2. Department of Information Science, Universidade Federal do Espírito Santo, Brazil. 3. Department of Electrical Engineering, Universidade Federal do Espírito Santo, Brazil.

**Correspondence:** Patrick Marques Ciarelli. Address: Department of Industrial Technology, Universidade Federal do Espírito Santo, Brazil. Email: pciarelli@lcad.inf.ufes.br

**Received:** June 30, 2013 **Accepted:** September 8, 2013 **Online Published:** September 29, 2013
**DOI:** 10.5430/air.v2n4p63 **URL:** http://dx.doi.org/10.5430/air.v2n4p63

## Abstract

Little attention has been paid to identifying the characteristics of a data set that provide favorable conditions for the task of incremental learning. In this work, several metrics were used to characterize data sets and identify the characteristics that may influence the trade-off between stability and plasticity. Three metrics are proposed for the evaluation of stability, plasticity and the trade-off between them in incremental techniques. The experiments were carried out using four incremental neural networks, and the results showed that the shape of the class boundary and spatial distribution of the samples have a great influence on this trade-off.

### Key words

Incremental learning, Stability, Plasticity, Complexity measures

## 1 Introduction

Several incremental neural networks have been proposed in the literature [1-7] to treat processes for which attaining a sufficient number of representative examples may not be feasible, either because the environment changes over time or because the rate at which examples become available is too slow [8].

In such situations, it is desirable to have a flexible model that can be upgraded without its performance on old data being affected. The training process should be such that it can learn new knowledge without forgetting the knowledge acquired previously. Such a model must be capable of maintaining the trade-off between two properties [9]:

- Stability: the model must be able to retain knowledge without a catastrophic forgetting failure, but it is not guaranteed that the model will be able to accommodate new knowledge;
- Plasticity: the model must be able to continuously learn new knowledge, without any guarantee that previously acquired knowledge will be preserved.

We argue that incremental learning techniques are amongst the most appropriate approaches to accommodating the conflicting requirements of stability and plasticity because such techniques make it possible to continue to learn new information while maintaining previously acquired knowledge. The difficulty is in maintaining acquired knowledge and learning new information because these objectives may be in conflict, and how each technique addresses this conflict indicates whether it is more inclined to provide greater stability or greater plasticity.

Little attention has been paid to analyzing the characteristics of data sets that provide favorable conditions for the task of incremental learning. Identifying such characteristics would help to determine the tractability of these data sets by systems based on incremental learning and to identify the best approach to addressing them.

In this work, several complexity measures employed to characterize data sets [10, 11] were used to analyze the characteristics of data sets in terms of the trade-off between stability and plasticity. We propose new metrics for evaluation of the stability, plasticity and the trade-off between stability and plasticity associated with incremental learning techniques.

The objectives of this study were mainly to investigate the impact of several data sets on the stability and plasticity of some incremental one-step learning techniques and to identify the characteristics of these data sets that influence the trade-off between stability and plasticity. The techniques evaluated in this work were incremental supervised neural networks designed for classification tasks, and their architectures adapt to each individual training sample. The techniques used for evaluation do not require reprocessing of any previous training sample to continue to learn new information. Although neural networks are the focus of this work, other classification techniques may be evaluated using the same procedure described in this study. It is also important to note that it is not an objective of this work to compare the performance of different classifiers but rather to use them to identify the effects of the characteristics of the data sets.

In our experiments, we note that data sets with simpler and well-defined class boundaries and attributes with high discriminative power tend to yield a better trade-off between stability and plasticity. Conversely, the incremental learning task becomes more complicated when class boundaries are complex and not well defined.

Lastly, an experiment was carried out to predict the trade-off between stability and plasticity of each incremental neural network evaluated using the characteristics of data sets. The results of this experiment show that it is possible to have a notion of the behavior of these classifiers without having to run them over a data set.

This paper is organized as follows. In Section 2, we describe the metrics employed to characterize the data sets and the metrics proposed for the evaluation of stability, plasticity and the trade-off between them. In Section 3, we describe our experiments and their results. We lastly present our conclusions in Section 4.

## 2 Measures

This section is divided into two parts. In the first subsection, we describe the complexity measures used to quantify the characteristics of the data sets that we used. In the second subsection, we describe the metrics proposed to evaluate the stability and plasticity of the classifiers for each of these data sets.

The measures of stability and plasticity help in the analysis of the behavior of a classifier when tested under conditions that are quite extreme with respect to the order of presentation of training samples, i.e., when it is presented only one class at a time. The use of complexity measures helps in the identification of some relevant characteristics in data sets, as nonlinearity, the discriminative power of attributes and the complexity of the class boundary. With the characteristics of data sets and the results obtained using a classifier for these same data sets, it is possible to draw a cause-and-effect profile and determine which characteristics of the data sets have greater impacts on the results obtained by a classifier.

### 2.1 Complexity measures

We have selected ten measures to describe the important aspects of each data set, such as statistical information, a feature’s discriminating power, overlap regions and class shapes. These measures, called complexity measures, are described below. We used in the experiments the Orriols-Puig et al algorithm [10] for computing the complexity measures (http://dcol.sourceforge.net/).

**The maximum Fisher's discriminant ratio** ($F1$) computes the maximum discriminative power of each attribute. It compares the difference between the means of the classes and the sum of the variances of the classes. A possible generalization for $C$ classes, which also considers all feature dimensions, can be stated as follows [10]:

$$F_1 = \max_{a=1}^{d} fd_a \tag{1}$$

where

$$fd_a = \frac{\sum_{c_i=1}^{C} \sum_{c_j=c_i+1}^{C} p_{c_i} p_{c_j} \left(\mu_{a,c_i} - \mu_{a,c_j}\right)^2}{\sum_{c_i=1}^{C} p_{c_i} \sigma_{a,c_i}^2}$$

where $C$ is the number of classes, $d$ is the number of input attributes, $fd_a$ is the discriminant ratio of the $a$th attribute and $p_{ci}$ is the proportion of examples of class $c_i$. The value of $\mu_{a,ci}$ is the median value of attribute a for class $c_i$, and $\sigma^2_{a,ci}$ is the variance of attribute a for class $c_i$. A slightly different version of this measure is presented in [12].

A high value for Fisher's discriminant ratio indicates that at least one of the attributes allows the separation of instances of different classes with partitions that are parallel to an axis of the feature space [10].

**The volume of overlap region** ($F2$) computes the length of the overlap range for each feature within two classes (when there are only two classes), normalized by the length of the total range in which all values of both classes are distributed. The volume of the overlap region is the product of the normalized lengths of the overlapping ranges for all features [12]. As a generalization for $C$ classes, $C > 2$, this measure is the sum of the absolute values for all pairs of classes [10-12] as described by the following equation:

$$F_2 = \sum_{c_i,c_j} \left| \prod_{a=1}^{d} \frac{\min\_\max_a - \max\_\min_a}{\max\_\max_a - \min\_\min_a} \right| \tag{2}$$

where

$$\min\_\max_a = \min\{\max(f_a, c_i), \max(f_a, c_j)\},$$

$$\max\_\min_a = \max\{\min(f_a, c_i), \min(f_a, c_j)\},$$

$$\max\_\max_a = \max\{\max(f_a, c_i), \max(f_a, c_j)\},$$

$$\min\_\min_a = \min\{\min(f_a, c_i), \min(f_a, c_j)\},$$

where $d$ is the number of input attributes, $f_a$ is the $a$th feature, $c_i$ is the $i$th class, $(c_i,c_j)$ goes through all pairs of classes and $\max(f_a, c_i)$ and $\min(f_a, c_i)$ are the maximum and minimum values, respectively, of feature $f_a$ for class $c_i$.

A low value of this measure means that the attributes can discriminate between examples of different classes [10].

**The maximum (individual) feature efficiency** (*F*3) is the maximum fraction of instances that can be separated by a particular feature [12]. For each pair of classes, the ratio of the number of instances separated by a particular feature to the number of instances of the pair of classes is computed for each feature. Then, the maximum discriminative ratio is taken as measure *F*3 [10].

**The collective feature efficiency** (*F*4) is determined in the same way as *F*3, except that this measure considers the discriminative power of all attributes. The collective discriminative power is computed in the following manner [10]. First, the most discriminative attribute, which is the attribute that can separate the maximum number of instances of one class, is computed. Next, all instances that can be discriminated are removed from the data set, and the next most discriminative attribute (from the remaining examples) is selected. This procedure is repeated until all of the examples have been discriminated or all of the attributes in the feature space have been analyzed. Lastly, the measure returns the proportion of instances that have been discriminated.

**The fraction of points on the class boundary** (*N*1) provides an estimate of the length of the class boundary. This method builds a Minimum Spanning Tree (MST) that connects all instances in the data set to their nearest neighbors and then counts the number of instances connected to different classes by the MST. These instances are considered to be close to the class boundary. This number is divided by the total number of instances in the data set, and the result is the value of *N*1 [12, 13].

High values of this measure indicate that the majority of the instances are close to the class boundary and that a classifier may have more difficulty defining the class boundary accurately [10].

**The ratio of average intra/inter class nearest neighbor (NN) distance** (*N*2) is a measure that computes the Euclidean distance from each instance to its nearest neighbor within or outside the class. The average of all the distances to the nearest intra-class neighbors and the average of all the distances to the nearest inter-class neighbors are then calculated. The ratio of these two values is the value of the measure *N*2.

Smaller values of this measure suggest more discriminant data, whereas higher values indicate that the examples of the same class are more dispersed [10].

**The leave-one-out error rate of the NN classifier** (*N*3) indicates how close the examples of different classes are and returns the leave-one-out error rate of the NN classifier. Low values of this measure indicate that there is a large gap in the class boundary [10, 13].

**The nonlinearity of the NN classifier** (*N*4) provides a measure of the nonlinearity of a data set. For a given data set, a test set is created by linear interpolation with random coefficients between pairs of randomly selected instances of the same class. The measure then returns the test error of the NN classifier trained with the original data set. This measure is sensitive to the smoothness of the classifier boundary and the overlap on the convex hull of the classes [10].

**The fraction of maximum covering spheres** (*T*1) counts the number of hyperspheres needed to cover each class, where each hypersphere is centered on one instance and grows to its maximum size before it reaches an instance of another class. Hyperspheres that are completely inside other hyperspheres are removed. The metric *T*1 is the number of hyperspheres divided by the number of instances in the data set. This metric provides a description of the shapes of the classes [10, 11].

**The average number of points per dimension** (*T*2) describes the density of the spatial distribution of instances, computed as the average number of instances per dimension (the number of attributes). This measure is a rough indicator of the sparseness of the data set [10, 11].

## 2.2 Measures of stability and plasticity

Three measures are proposed to evaluate the degrees of stability, plasticity and the trade-off between them. Before introducing these measures, however, we will explain how to use the procedures to address these measures.

When using the three measures, it is necessary to divide each data set into $C$ subsets, where $C$ is the number of classes of the data set, $N_i$, $i = 1, \ldots, C$, is the number of instances in each subset, and each subset contains only instances of a unique class. This division of the data sets avoids similar patterns arising in different subsets, which would make it difficult to perceive when a classifier has learned new information or has simply used prior knowledge.

After the division of the data set as previously described, the classifier is initially trained with the first subset $S_1$ and tested with the same subset. The number of instances correctly classified in subset $S_1$ is then counted, and the value is denoted by $A_{1,1}$, where the first index indicates the number of subsets used for training and the second index is the subset used for the test. Next, the subsets $S_i$, $i = 2, \ldots, C\text{-}1$, are also used for training; the classifier is tested with the subset $S_C$, and the number of instances correctly classified is referred as $B_{C\text{-}1,C}$, where the meaning of the indexes is the same as above. Later, the subset $S_C$ is also used for training, the value of $A_{C,1}$ is computed for $S_1$ and $B_{C,C}$ is obtained for $S_C$.

This procedure is repeated $C$ times so that all of the subsets are used once as the first and once as the last subset of training. That is, this procedure for training and testing is repeated until the values of $A_{1,i}$, $A_{C,i}$, $B_{C\text{-}1,i}$ and $B_{C,i}$, $i = 1, \ldots, C$, have all been calculated.

Figure 1 illustrates the calculation of values $A$ and $B$ for a data set with two classes ($C = 2$). First, the data set is divided into 2 subsets, where the subset $S_1$ is the first subset used for training. After the training with $S_1$, this same subset is used for testing, and the number of instances correctly classified is counted. This number is the value of $A_{1,1}$. Then, the subset $S_2$ is used for testing, and the number of instances correctly classified is counted. This number is the value of $B_{1,2}$. Next, the subset $S_2$ is also used for training (that is, the classifier is trained with $S_1$ and $S_2$, respectively) and both subsets $S_1$ and $S_2$ are utilized for testing. The number of instances correctly classified of $S_1$ and $S_2$ are identified by the values $A_{2,1}$ and $B_{2,2}$, respectively. After these calculations, the order of the training subsets is reversed. Now, $S_2$ is the first subset used for training and $S_1$ is the second subset. With this change, the values of $A_{1,2}$ and $B_{1,1}$ are obtained when the classifier is trained only with $S_2$, and $A_{2,2}$ and $B_{2,1}$ are calculated when the two subsets are used for training. The values of $A$ are obtained for the first training subset, and the values of $B$ are obtained for the last training subset. The number of sets is equal to number of classes.

DATA SET
S1
S2

**Figure 1.** Illustration of the division of a data set with two classes into two subsets ($S_1$ and $S_2$) for the calculation of $A$ and $B$. Each subset contains instances of a unique class. The two subsets are used for training, with $S_1$ being the first subset used. The values of $A$ are the instances correctly classified of $S_1$ before and after $S_2$ is used for training. The values of $B$ are the instances correctly classified of $S_2$ before and after $S_2$ is used for training.

The value of $A_{1,i}$ is the number of instances that the classifier learns of subset $S_i$, and $A_{C,i}$ is the number of instances recognized of subset $S_i$ after the data of the $C$ classes have been used for training. On the other hand, $B_{C\text{-}1,i}$ is the number of instances of subset $S_i$ that the classifier recognizes before it is trained with subset $S_i$, and $B_{C,i}$ is the number of instances recognized after training with $S_i$. The measures presented below are computed for the following two cases: the average amount of information retained for each class after learning the other classes and the average amount of information learned for each class after the other classes have been learned.

**Retention** (*R*) measures the degree of stability of a classifier, i.e., the ability of a classifier to retain old knowledge when a new piece of information is presented.

The Retention value *R* of a classifier with respect to a data set is calculated using Eq. (3):

$$R = \frac{1}{C}\sum_{i=1}^{C} R_i \times 100\% \tag{3}$$

where

$$R_i = \begin{cases} \dfrac{A_{C,i}}{A_{1,i}}, & \text{if} \quad A_{1,i} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

The value of the Retention metric is computed as the mean of the ratios between the number of recognized instances of each class before and after presenting all of the classes to the classifier. The value of *R* ranges from 0 to 100%. The larger the value of *R* is, the more able the classifier is to retain old knowledge.

**Innovation** (*I*) measures the degree of plasticity of a classifier, i.e., the ability of a classifier to learn new knowledge.

The Innovation value *I* is calculated using Eq. (4):

$$I = \frac{1}{C}\sum_{i=1}^{C} I_i \times 100\% \tag{4}$$

where

$$I_i = \begin{cases} \dfrac{B_{C,i} - B_{C-1,i}}{N_i - B_{C-1,i}}, & \text{if} \quad N_i - B_{C-1,i} > 0, \\ 1, & \text{otherwise.} \end{cases}$$

The value of the Innovation metric is calculated as the mean of the ratios between the number of recognized instances of each class before and after presenting this class for the classifier. The value of *I* ranges from 0 to 100%. The larger the value of *I* is, the more able the classifier is to learn new knowledge.

**The Harmonic Mean between retention and innovation** (*H*) evaluates the trade-off between the stability and plasticity of a classifier for a data set. This value is calculated using Eq. (5):

$$H = \frac{2RI}{R+I} \times 100\% \tag{5}$$

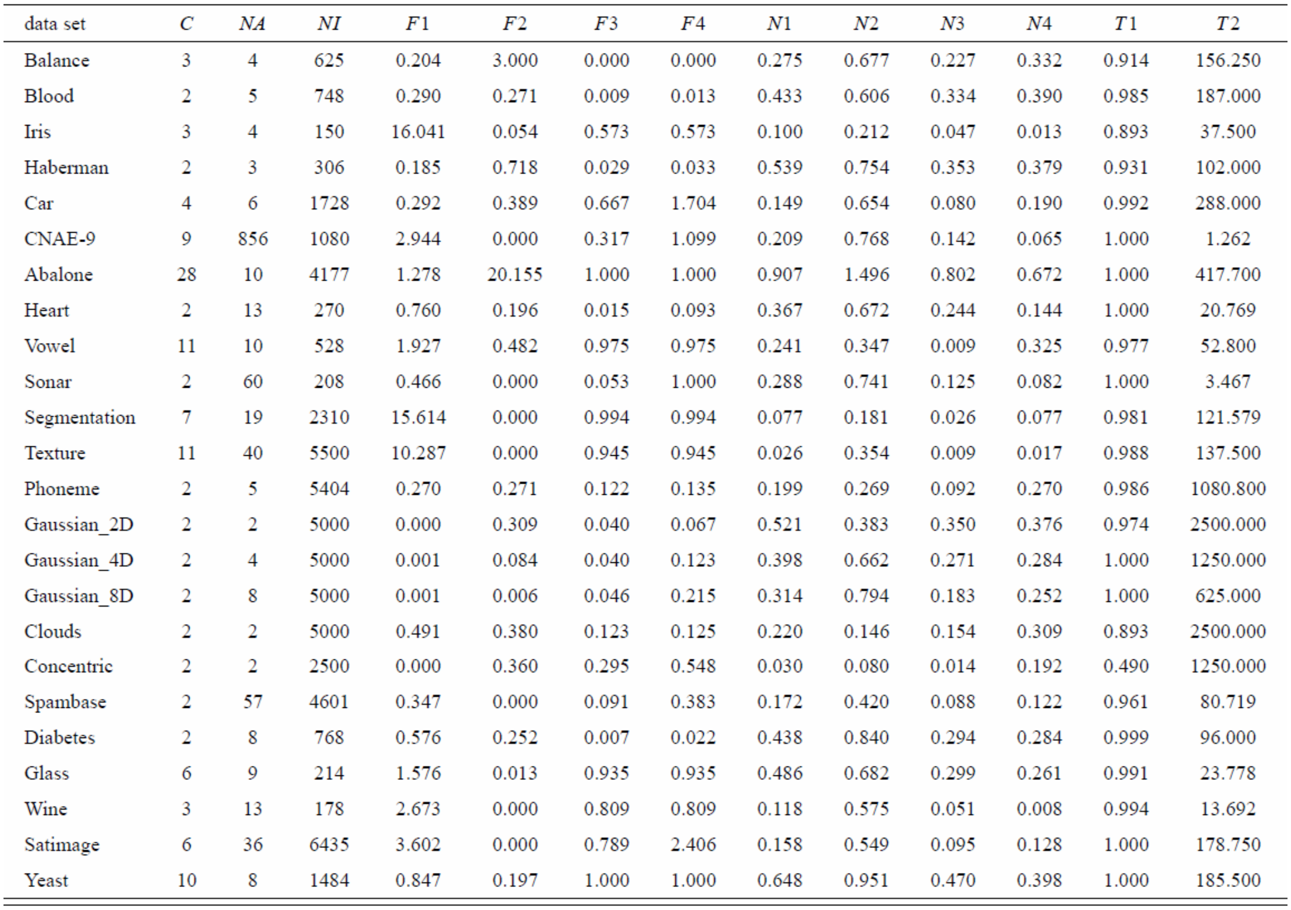
The larger the value of *H* is, the better able the classifier is to perform a trade-off between stability and plasticity. The value of *H* ranges from 0 to 100%.

A result when *R* = *I* suggests that the incremental method is not sensitive to the order of presentation of the classes, although this does not imply that it is insensitive to the order of the data presentation.

# 3 Experiments

The experiments were carried out on the data sets shown in Table 1. Their characteristics are the number of classes (*C*), the number of attributes (*NA*), the number of instances (*NI*), and the complexity measures described in Section 2. These data sets were obtained from the UCI Machine Learning Database Repository (http://archive.ics.uci.edu/ml/) and from the ELENA European Project (http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/).

**Table 1.** Data sets: characteristics and complexity measures.



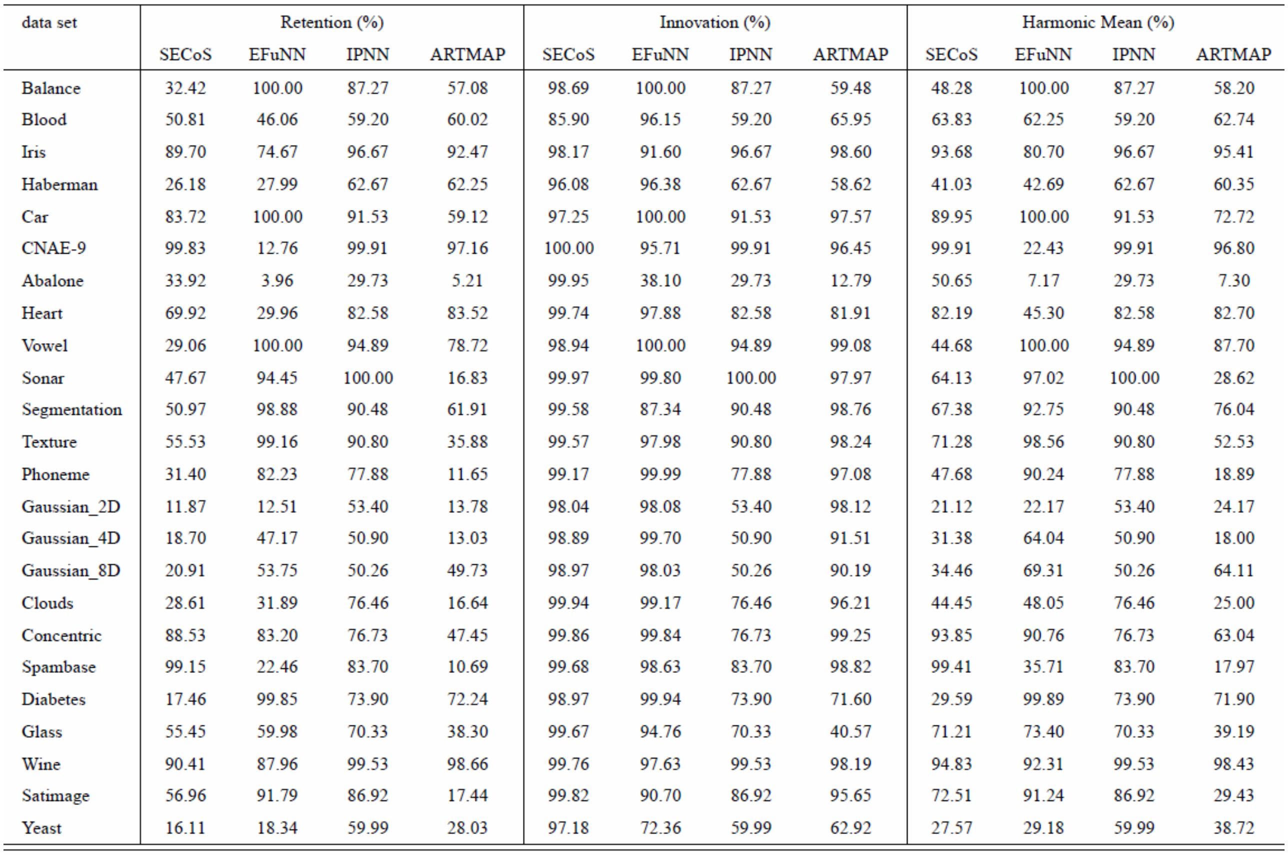
| data set | *C* | *NA* | *NI* | *F1* | *F2* | *F3* | *F4* | *N1* | *N2* | *N3* | *N4* | *T1* | *T2* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Balance | 3 | 4 | 625 | 0.204 | 3.000 | 0.000 | 0.000 | 0.275 | 0.677 | 0.227 | 0.332 | 0.914 | 156.250 |
| Blood | 2 | 5 | 748 | 0.290 | 0.271 | 0.009 | 0.013 | 0.433 | 0.606 | 0.334 | 0.390 | 0.985 | 187.000 |
| Iris | 3 | 4 | 150 | 16.041 | 0.054 | 0.573 | 0.573 | 0.100 | 0.212 | 0.047 | 0.013 | 0.893 | 37.500 |
| Haberman | 2 | 3 | 306 | 0.185 | 0.718 | 0.029 | 0.033 | 0.539 | 0.754 | 0.353 | 0.379 | 0.931 | 102.000 |
| Car | 4 | 6 | 1728 | 0.292 | 0.389 | 0.667 | 1.704 | 0.149 | 0.654 | 0.080 | 0.190 | 0.992 | 288.000 |
| CNAE-9 | 9 | 856 | 1080 | 2.944 | 0.000 | 0.317 | 1.099 | 0.209 | 0.768 | 0.142 | 0.065 | 1.000 | 1.262 |
| Abalone | 28 | 10 | 4177 | 1.278 | 20.155 | 1.000 | 1.000 | 0.907 | 1.496 | 0.802 | 0.672 | 1.000 | 417.700 |
| Heart | 2 | 13 | 270 | 0.760 | 0.196 | 0.015 | 0.093 | 0.367 | 0.672 | 0.244 | 0.144 | 1.000 | 20.769 |
| Vowel | 11 | 10 | 528 | 1.927 | 0.482 | 0.975 | 0.975 | 0.241 | 0.347 | 0.009 | 0.325 | 0.977 | 52.800 |
| Sonar | 2 | 60 | 208 | 0.466 | 0.000 | 0.053 | 1.000 | 0.288 | 0.741 | 0.125 | 0.082 | 1.000 | 3.467 |
| Segmentation | 7 | 19 | 2310 | 15.614 | 0.000 | 0.994 | 0.994 | 0.077 | 0.181 | 0.026 | 0.077 | 0.981 | 121.579 |
| Texture | 11 | 40 | 5500 | 10.287 | 0.000 | 0.945 | 0.945 | 0.026 | 0.354 | 0.009 | 0.017 | 0.988 | 137.500 |
| Phoneme | 2 | 5 | 5404 | 0.270 | 0.271 | 0.122 | 0.135 | 0.199 | 0.269 | 0.092 | 0.270 | 0.986 | 1080.800 |
| Gaussian_2D | 2 | 2 | 5000 | 0.000 | 0.309 | 0.040 | 0.067 | 0.521 | 0.383 | 0.350 | 0.376 | 0.974 | 2500.000 |
| Gaussian_4D | 2 | 4 | 5000 | 0.001 | 0.084 | 0.040 | 0.123 | 0.398 | 0.662 | 0.271 | 0.284 | 1.000 | 1250.000 |
| Gaussian_8D | 2 | 8 | 5000 | 0.001 | 0.006 | 0.046 | 0.215 | 0.314 | 0.794 | 0.183 | 0.252 | 1.000 | 625.000 |
| Clouds | 2 | 2 | 5000 | 0.491 | 0.380 | 0.123 | 0.125 | 0.220 | 0.146 | 0.154 | 0.309 | 0.893 | 2500.000 |
| Concentric | 2 | 2 | 2500 | 0.000 | 0.360 | 0.295 | 0.548 | 0.030 | 0.080 | 0.014 | 0.192 | 0.490 | 1250.000 |
| Spambase | 2 | 57 | 4601 | 0.347 | 0.000 | 0.091 | 0.383 | 0.172 | 0.420 | 0.088 | 0.122 | 0.961 | 80.719 |
| Diabetes | 2 | 8 | 768 | 0.576 | 0.252 | 0.007 | 0.022 | 0.438 | 0.840 | 0.294 | 0.284 | 0.999 | 96.000 |
| Glass | 6 | 9 | 214 | 1.576 | 0.013 | 0.935 | 0.935 | 0.486 | 0.682 | 0.299 | 0.261 | 0.991 | 23.778 |
| Wine | 3 | 13 | 178 | 2.673 | 0.000 | 0.809 | 0.809 | 0.118 | 0.575 | 0.051 | 0.008 | 0.994 | 13.692 |
| Satimage | 6 | 36 | 6435 | 3.602 | 0.000 | 0.789 | 2.406 | 0.158 | 0.549 | 0.095 | 0.128 | 1.000 | 178.750 |
| Yeast | 10 | 8 | 1484 | 0.847 | 0.197 | 1.000 | 1.000 | 0.648 | 0.951 | 0.470 | 0.398 | 1.000 | 185.500 |

The incremental neural networks used for evaluation were the Evolving Fuzzy Neural Network (EFuNN) [14, 15], the Simple Evolving Connectionist System (SECoS) [16], the Incremental Probabilistic Neural Network (IPNN) [17] and the Fuzzy ARTMAP [18]. These neural networks were chosen because they have similar features: they have a constructive architecture that grows as needed, they process one training sample at a time, they can continually learn over their whole existence, and they do not reprocess any previous training sample.

The EFuNN uses fuzzy logic for classification. Fuzzification is performed for each sample, and the result is compared to the pattern stored within its architecture. More fuzzy logic membership functions result in a greater architecture size. This neural network has been used for various applications, and it is generally considered to be a neural network that can be trained quickly and adapts well to new data [19].

The SECoS is an incremental version of the Multi-Layer Perceptron (MLP) (SECoS is also called eMLP (evolving MLP) [15]) that has a hidden layer. Each neuron added in the hidden layer represents the training sample that caused its inclusion. SECoS can also be viewed as a version of EFuNN in which fuzzy logic is not used for classification.

**Table 2.** Results achieved for the Retention, Innovation and Harmonic Mean metrics.



| data set | Retention (%) | | | | Innovation (%) | | | | Harmonic Mean (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SECoS | EFuNN | IPNN | ARTMAP | SECoS | EFuNN | IPNN | ARTMAP | SECoS | EFuNN | IPNN | ARTMAP |
| Balance | 32.42 | 100.00 | 87.27 | 57.08 | 98.69 | 100.00 | 87.27 | 59.48 | 48.28 | 100.00 | 87.27 | 58.20 |
| Blood | 50.81 | 46.06 | 59.20 | 60.02 | 85.90 | 96.15 | 59.20 | 65.95 | 63.83 | 62.25 | 59.20 | 62.74 |
| Iris | 89.70 | 74.67 | 96.67 | 92.47 | 98.17 | 91.60 | 96.67 | 98.60 | 93.68 | 80.70 | 96.67 | 95.41 |
| Haberman | 26.18 | 27.99 | 62.67 | 62.25 | 96.08 | 96.38 | 62.67 | 58.62 | 41.03 | 42.69 | 62.67 | 60.35 |
| Car | 83.72 | 100.00 | 91.53 | 59.12 | 97.25 | 100.00 | 91.53 | 97.57 | 89.95 | 100.00 | 91.53 | 72.72 |
| CNAE-9 | 99.83 | 12.76 | 99.91 | 97.16 | 100.00 | 95.71 | 99.91 | 96.45 | 99.91 | 22.43 | 99.91 | 96.80 |
| Abalone | 33.92 | 3.96 | 29.73 | 5.21 | 99.95 | 38.10 | 29.73 | 12.79 | 50.65 | 7.17 | 29.73 | 7.30 |
| Heart | 69.92 | 29.96 | 82.58 | 83.52 | 99.74 | 97.88 | 82.58 | 81.91 | 82.19 | 45.30 | 82.58 | 82.70 |
| Vowel | 29.06 | 100.00 | 94.89 | 78.72 | 98.94 | 100.00 | 94.89 | 99.08 | 44.68 | 100.00 | 94.89 | 87.70 |
| Sonar | 47.67 | 94.45 | 100.00 | 16.83 | 99.97 | 99.80 | 100.00 | 97.97 | 64.13 | 97.02 | 100.00 | 28.62 |
| Segmentation | 50.97 | 98.88 | 90.48 | 61.91 | 99.58 | 87.34 | 90.48 | 98.76 | 67.38 | 92.75 | 90.48 | 76.04 |
| Texture | 55.53 | 99.16 | 90.80 | 35.88 | 99.57 | 97.98 | 90.80 | 98.24 | 71.28 | 98.56 | 90.80 | 52.53 |
| Phoneme | 31.40 | 82.23 | 77.88 | 11.65 | 99.17 | 99.99 | 77.88 | 97.08 | 47.68 | 90.24 | 77.88 | 18.89 |
| Gaussian_2D | 11.87 | 12.51 | 53.40 | 13.78 | 98.04 | 98.08 | 53.40 | 98.12 | 21.12 | 22.17 | 53.40 | 24.17 |
| Gaussian_4D | 18.70 | 47.17 | 50.90 | 13.03 | 98.89 | 99.70 | 50.90 | 91.51 | 31.38 | 64.04 | 50.90 | 18.00 |
| Gaussian_8D | 20.91 | 53.75 | 50.26 | 49.73 | 98.97 | 98.03 | 50.26 | 90.19 | 34.46 | 69.31 | 50.26 | 64.11 |
| Clouds | 28.61 | 31.89 | 76.46 | 16.64 | 99.94 | 99.17 | 76.46 | 96.21 | 44.45 | 48.05 | 76.46 | 25.00 |
| Concentric | 88.53 | 83.20 | 76.73 | 47.45 | 99.86 | 99.84 | 76.73 | 99.25 | 93.85 | 90.76 | 76.73 | 63.04 |
| Spambase | 99.15 | 22.46 | 83.70 | 10.69 | 99.68 | 98.63 | 83.70 | 98.82 | 99.41 | 35.71 | 83.70 | 17.97 |
| Diabetes | 17.46 | 99.85 | 73.90 | 72.24 | 98.97 | 99.94 | 73.90 | 71.60 | 29.59 | 99.89 | 73.90 | 71.90 |
| Glass | 55.45 | 59.98 | 70.33 | 38.30 | 99.67 | 94.76 | 70.33 | 40.57 | 71.21 | 73.40 | 70.33 | 39.19 |
| Wine | 90.41 | 87.96 | 99.53 | 98.66 | 99.76 | 97.63 | 99.53 | 98.19 | 94.83 | 92.31 | 99.53 | 98.43 |
| Satimage | 56.96 | 91.79 | 86.92 | 17.44 | 99.82 | 90.70 | 86.92 | 95.65 | 72.51 | 91.24 | 86.92 | 29.43 |
| Yeast | 16.11 | 18.34 | 59.99 | 28.03 | 97.18 | 72.36 | 59.99 | 62.92 | 27.57 | 29.18 | 59.99 | 38.72 |

The IPNN is an incremental version of the Probabilistic Neural Network [20]. In the training phase of this network, training samples are only stored in its architecture, and the transfer function of its neurons is Gaussian. The advantages of the IPNN algorithm include its easy implementation, quick training and the fact that the order in which samples are presented for training does not affect its learning. However, it has the disadvantage of having many neurons in its architecture, thus requiring a large amount of memory and classification time.

The Fuzzy ARTMAP is based on aspects of how the brain processes information. The ARTMAP has two layers of prototype units in a supervised learning structure. The first unit takes the input instances and the second unit takes the labels associated with the instances. During the learning, some adjustments are made in the units to make the correct classification. A similar structure is used in the Fuzzy ARTMAP, but the units use fuzzy logic.

The experiments were performed using the following methodology. Initially, each attribute f of each data set in the interval [a, b] was mapped to [0, 1] using the following equation:

$$f = \frac{(f-a)(d-c)}{b-a} - c \tag{6}$$

where $a$ and $b$ are the smallest and largest values of $f$ , respectively, $c$ = 0 and $d$ = 1. The information in Table 1 was obtained after this procedure.

Then, each algorithm was calibrated so that half of each data set was used for training and the other half was used for validation. After calibration, each data set was divided into $C$ subsets, where $C$ is the number of classes in the data set, and the methodology presented in Section 2 was used to calculate the Retention, Innovation and Harmonic Mean metrics.

These procedures were performed 20 times with random partitions. The average results for each data set and classifier are shown in Table 2.

In Table 2, high values for the Harmonic Mean were obtained for some data sets, such as Balance, Iris, Car, CNAE-9, Segmentation, Texture and Wine. The high values of the Harmonic Mean indicate that the classifiers provide a good trade-off between stability and plasticity, and the corresponding data sets present characteristics that make them suitable for incremental learning tasks. On the other hand, other data sets yielded low values of the Harmonic Mean.

In analyzing the techniques, it was observed that the techniques SECoS, EFuNN and Fuzzy ARTMAP behaved similarly. These three algorithms have a great ability to learn new information, as indicated by the Innovation metric (many results above 90%). The Retention metric, however, has low values for some of the data sets. These results demonstrate that these techniques focus more on learning new information than on retaining old knowledge. This adaptation to new data is slow, so the rate of deterioration of old knowledge is reduced.

The results of the experiments show that unlike the other classifiers evaluated, IPNN is not influenced by the order of data presentation. Thus, its values of Retention and Innovation for each data set are equal. Although its values for Innovation are slightly smaller than those of the other methods, its values for Retention are higher. IPNN thus has a greater capacity to store old information and a weaker ability to learn new information. For tasks for which there is continuous change in the features, the Fuzzy ARTMAP, SECoS and EFuNN classifiers have a greater capacity than IPNN to adapt to these changes. On the other hand, IPNN has a more evenly balanced trade-off between retaining and learning information.

We performed two-tailed paired *t*-tests at the 1% significance level to compare the Retention and Innovation results for each algorithm. The results of the *t*-tests indicated that the Innovation (plasticity) of Fuzzy ARTMAP, SECoS and EFuNN were significantly greater than their respective Retention values (stability), confirming the results of the previous analysis. As expected, there were no significant differences between the Innovation and Retention values for the IPNN algorithm, because they were equal to each other.

An important issue is to identify before-hand which data set characteristics may affect the trade-off between stability and plasticity, thus predicting the behavior that a data set may produce in an incremental learning procedure. To determine these characteristics, the Kendall correlation coefficient [21] between each measure and result was computed, together with a test of the hypothesis of no correlation at a significance level of 1%, to verify whether the correlation between the parameters was significant. The complexity measures $N1$ and $N3$ were found to be significantly correlated with the results of IPNN, SECoS and EFuNN. The smaller the values of these measures were, the larger the value of the Harmonic Mean was. The measure $N4$, which measures the degree of nonlinearity, was also significantly correlated with the results of IPNN and SECoS. High values of this measure suggest that the instances of different classes are very similar, that the class boundary is not well defined and that the separation between classes is nonlinear. That is, the spatial location of the instances has a great effect on the performance of these algorithms. For the Fuzzy ARTMAP and IPNN, the influence of the measure $T2$ was noted. The smaller the value of $T2$ is, the larger the value of the Harmonic Mean is. Higher values of $T2$ indicate a greater concentration of instances in a region, and if these instances have a complex boundary of separation, many instances may be misclassified. Another relevant measure for Fuzzy ARTMAP is $N4$. Classes with nonlinear boundaries are harder to classify correctly, although $N4$ was not significantly correlated.

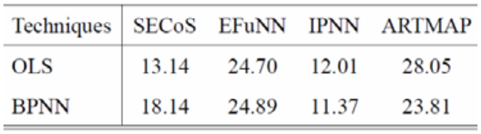
The complexity measures $F1$ and $F2$ were found to be correlated (but not significantly) with the results. The larger the value of $F1$ is and the smaller the value of $F2$ is, the larger the value of the Harmonic Mean is. That is, a high discriminative power of the attributes ($F1$) and a low distribution overlap of different classes ($F2$) are important in addressing the problem of the trade-off between stability and plasticity. However, we observed high correlations between the values of $F1$ (a negative correlation) and $F2$ (a positive correlation) with the values of $N4$. This is evidence that the same conditions affect all three of these measures. However, the measure $N4$ better reflects this effect on the Harmonic Mean, as shown by the results of the statistical tests.

The results of this analysis explain the high values for Harmonic Mean obtained for at least three techniques for the data sets Iris, Segmentation and Texture, which had high values of $F1$ and low values of $N4$. For these three data sets, the $F1$ measures were greater than 10. Other data sets with low values of $N4$, such as CNAE-9 and Wine, also had high values for the Harmonic Mean. However, data sets with high values for $N1$, $N3$ and $N4$, such as Abalone, Gaussian_2D and Yeast, had low Harmonic Mean values. Eventually, one or more techniques behave differently than expected, and one of the reasons is related to the characteristics of the algorithms. For example, EFuNN is the only one of the algorithms considered that did not achieve a high value for the CNAE-9 data set. Nonetheless, CNAE-9 has a huge amount of zeros in its attributes (more than 90%), and as noted by Watts [19], this has an adverse effect on the performance of this technique.

It would be interesting to use the characteristics of data sets to predict the relationship between stability and plasticity. To assess this possibility, we carried out a final experiment to predict the Harmonic Mean values of the data sets using the leave-one-out procedure and determined the value of the mean absolute error (MAE) for each algorithm.

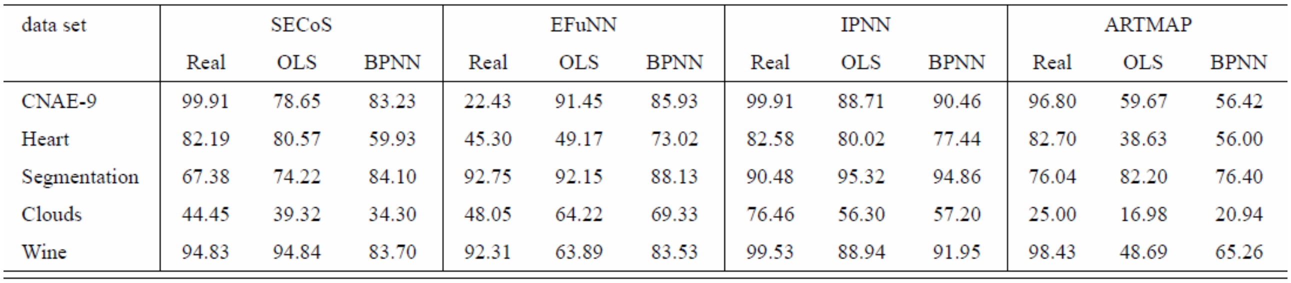
In this experiment, each data set was considered to be an instance, its attributes were considered to be the complexity measures, and each instance was labeled with the Harmonic Mean. All measures were mapped to the interval [-1, 1] using Equation 4, where $c$ = -1 and $d$ = 1. Each instance was then normalized by the infinity norm. The Harmonic Mean values were also mapped to the interval [-1, 1]. Next, two methods were used to predict the Harmonic Mean for each technique: a method based on Ordinary Least Squares (OLS) regression, which considers a linear relationship among the attributes, and an approach based on the use of a Back-Propagation Neural Network (BPNN). The results reported are the mean values for 20 BPNN runs. The predicted values were scaled to the interval [0, 100].

**Table 3.** Mean Absolute Error in the Prediction of Harmonic Mean.



| Techniques | SECoS | EFuNN | IPNN | ARTMAP |
|---|---|---|---|---|
| OLS | 13.14 | 24.70 | 12.01 | 28.05 |
| BPNN | 18.14 | 24.89 | 11.37 | 23.81 |

**Table 4.** Prediction of Harmonic Mean in percentage.



| data set | SECoS | | | EFuNN | | | IPNN | | | ARTMAP | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Real | OLS | BPNN | Real | OLS | BPNN | Real | OLS | BPNN | Real | OLS | BPNN |
| CNAE-9 | 99.91 | 78.65 | 83.23 | 22.43 | 91.45 | 85.93 | 99.91 | 88.71 | 90.46 | 96.80 | 59.67 | 56.42 |
| Heart | 82.19 | 80.57 | 59.93 | 45.30 | 49.17 | 73.02 | 82.58 | 80.02 | 77.44 | 82.70 | 38.63 | 56.00 |
| Segmentation | 67.38 | 74.22 | 84.10 | 92.75 | 92.15 | 88.13 | 90.48 | 95.32 | 94.86 | 76.04 | 82.20 | 76.40 |
| Clouds | 44.45 | 39.32 | 34.30 | 48.05 | 64.22 | 69.33 | 76.46 | 56.30 | 57.20 | 25.00 | 16.98 | 20.94 |
| Wine | 94.83 | 94.84 | 83.70 | 92.31 | 63.89 | 83.53 | 99.53 | 88.94 | 91.95 | 98.43 | 48.69 | 65.26 |

The MAE results for each technique are presented in Table 3, and the results for some data sets are presented in Table 4. Although OLS is a simpler method than BPNN, the two yielded similar mean results. The greatest differences were for the neural networks SECoS and ARTMAP. The results obtained for SECoS and IPNN were more accurate than those for EFuNN and ARTMAP. The Kendall correlation coefficient values indicate that the SECoS and IPNN results were more strongly correlated with the complexity measures than the other techniques, meaning that it is easier to predict their behavior.

From the results obtained for these techniques, particularly SECoS and IPNN, it is possible to estimate the trade-off between stability and plasticity, making it possible to avoid running these incremental approaches over the data sets. Furthermore, this analysis approach aids in the identification of the most promising incremental approach for tackling a specific problem because the behavior of a technique can be estimated in advance for the data set at hand.

# 4 Conclusions

In this study, three metrics are proposed for measuring stability, plasticity, and the trade-off between the two in incremental supervised learning. A series of experiments was carried out to assess the degrees of stability and plasticity of some incremental supervised neural networks designed for classification tasks. In general, the incremental neural networks presented high degrees of plasticity for all data sets. Some data sets had high values for stability and the trade-off between stability and plasticity, and other data sets had low values. IPNN was an exception in that it was not affected by the order of the data presentation; it exhibited a higher capacity to retain old knowledge, although it had a lower capacity to learn new information.

The analysis of the results indicated that the spatial distribution of the samples in a data set is an important characteristic influencing the trade-off between stability and plasticity. Data sets with attributes that have high discriminative power and well-defined class boundaries tend to provide a better trade-off between stability and plasticity. On the other hand, when the instances of different classes are more similar and class boundaries are not well defined, the task of incremental learning is more complex.

Lastly, an experiment was carried out to predict the trade-off between stability and plasticity of each incremental neural network evaluated in this study. In this experiment, reasonable predictions were obtained and showed that it was possible to have an idea of the behavior of classifiers without having to run them on a data set. This type of analysis may be useful in identifying the most promising incremental approach for tackling a specific problem with respect to the trade-off between plasticity and stability. Another advantage of this analysis approach is that it may be used to build ensembles of techniques that can be weighted with respect to their ability to achieve a trade-off between plasticity and stability. This approach can be used to obtain an estimate of the ensemble weights without having to perform long and complex processes for calibration of the weights, as presented in [22].

# References

[1] I. Igari, J. Tani, Incremental learning of sequence patterns with a modular network model. Neurocomputing. 2009; 72 (7-9): 1910–1919. http://dx.doi.org/10.1016/j.neucom.2008.05.002

[2] D. M. Lee, S. Choi, Application of on-line adaptable neural network for the rolling force set-up of a plate mill. Engineering Applications of Artificial Intelligence. 2004; 17 (5): 557–565. http://dx.doi.org/10.1016/j.engappai.2004.03.008

[3] C. MacLeod, G. Maxwell, S. Muthuraman, Incremental growth in modular neural networks. Engineering Applications of Artificial Intelligence. 2009; 22 (4-5): 660–666. http://dx.doi.org/10.1016/j.engappai.2008.11.002

[4] H. Ohta, Y. P. Gunji, Recurrent neural network architecture with pre-synaptic inhibition for incremental learning. Neural Networks. 2006; 19 (8): 1106–1119. http://dx.doi.org/10.1016/j.neunet.2006.06.005

[5] S. Shiotani, T. Fukuda, T. Shibata, A neural network architecture for incremental learning. Neurocomputing.1995; 9 (2): 111–130. http://dx.doi.org/10.1016/0925-2312(94)00061-V

[6] M. C. Su, J. Lee, K. L. Hsieh, A new ARTMAP-based neural network for incremental learning. Neurocomputing. 2006; 69 (16-18): 2284–2300. http://dx.doi.org/10.1016/j.neucom.2005.06.020

[7] M. Tscherepanow, M. Kortkamp, M. Kammer, A hierarchical ART network for the stable incremental learning of topological structures and associations from noisy data. Neural Networks. 2011; 24 (8): 906–916. http://dx.doi.org/10.1016/j.neunet.2011.05.009

[8] C. Giraud-Carrier, A note on the utility of incremental learning. AI Communications. 2000; 13: 215–223. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.4049

[9] R. Polikar, L. Udpa, S. S. Udpa, V. Honavar, Learn++: an incremental learning algorithm for supervised neural network. IEEE Transactions on Systems, Man, and Cybernetics. C: Applications and Reviews. 2001; 31 (4): 497–508. http://dx.doi.org/10.1109/5326.983933

[10] A. Orriols-Puig, N. Macià, T. K. Ho, Documentation for the data complexity library in C++. Technical Report. La Salle - Universitat Ramon Llull. 2010. Available from: http://dcol.sourceforge.net/

[11] J. M. Sotoca, J. S. Sánchez, R. A. Mollineda, A review of data complexity measures and their applicability to pattern classification problems. Actas del III Taller Nacional de Minería de Datos y Aprendizaje. 2005: 77 – 83. Available from: http://www.lsi.us.es/redmidas/CEDI/papers/407.pdf

[12] R. A. Mollineda, J. S. Sánchez, J.M. Sotoca, Data characterization for effective prototype selection. Proc. of the 2nd Iberian Conf. on Pattern Recognition and Image Analysis. 2005; 3523: 27 – 34. http://dx.doi.org/10.1007/11492542_4

[13] T. K. Ho, M. Basu, Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002; 24 (3): 289 – 300. http://dx.doi.org/10.1109/34.990132

[14] N. Kasabov, Evolving fuzzy neural networks - algorithms, applications and biological motivation. Methodologies for the conception, design and application of soft computing. 1998: 271–274.

[15] N. Kasabov, Connectionist Systems - The Knowledge Engineering Approach. Springer-Verlag London. 2nd edition, 2007.

[16] M. Watts, N. Kasabov, Simple evolving connectionist systems and experiments on isolated phoneme recognition. IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks. 2000: 232–239. http://dx.doi.org/10.1109/ECNN.2000.886239

[17] N. Bhattacharyya, A. Metla, R. Bandyopadhyay, B. Tudu, A. Jana, Incremental PNN classifier for a versatile electronic nose. Third International Conference on Sensing Technology. 2008: 242–247. http://dx.doi.org/10.1109/ICSENST.2008.4757106

[18] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, D. B. Rosen, Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Networks. 1992; 3 (5): 698–713. http://dx.doi.org/10.1109/72.159059

[19] M. J. Watts, A decade of Kasabov's evolving connectionist systems: a review. IEEE Transactions on Systems, Man, and Cybernetics. C: Applications and Reviews. 2009; 39 (3): 253–269. http://dx.doi.org/10.1109/TSMCC.2008.2012254

[20] D. Specht, Probabilistic neural networks. Neural Networks. 1990; 3 (1): 109 – 118. http://dx.doi.org/10.1016/0893-6080(90)90049-Q

[21] M. Kendall, A new measure of rank correlation. Biometrika. 1938; 30: 81–93. http://dx.doi.org/10.1093/biomet/30.1-2.81

[22] C. Domeniconi, M. Al-Razgan, Weighted Cluster Ensembles: Methods and Analysis. Department of Computer Science, George Mason University, Technical Report, 2007. Available from: http://cs.gmu.edu/~tr-admin/papers/ISE-TR-07-06.pdf