Theory of impossible worlds: Toward a physics of information

Paolo Massimo Buscema, Pier Luigi Sacco, Francesca Della Torre, Giulia Massini, Marco Breda, and Guido Ferilli

Citation: Chaos **28**, 055914 (2018); doi: 10.1063/1.5024371
View online: https://doi.org/10.1063/1.5024371
View Table of Contents: http://aip.scitation.org/toc/cha/28/5
Published by the American Institute of Physics

Theory of impossible worlds: Toward a physics of information

Paolo Massimo Buscema,[1,2,a)] Pier Luigi Sacco,[3,4] Francesca Della Torre,[1] Giulia Massini,[1] Marco Breda,[1] and Guido Ferilli[1,3]

[1]*Semeion Research Centre of Science of Communication, Rome 00128, Italy*
[2]*Department Mathematical and Statistical Sciences, University of Colorado at Denver, Denver, Colorado 80204, USA*
[3]*Faculty of Arts, Tourism and Markets, IULM University, Milan 20143, Italy*
[4]*FBK-IRVAPP, Via Santa Croce 77, Trento 38122, Italy*

(Received 31 January 2018; accepted 23 April 2018; published online 29 May 2018)

In this paper, we introduce an innovative approach to the fusion between datasets in terms of attributes and observations, even when they are not related at all. With our technique, starting from datasets representing independent worlds, it is possible to analyze a single global dataset, and transferring each dataset onto the others is always possible. This procedure allows a deeper perspective in the study of a problem, by offering the chance of looking into it from other, independent points of view. Even unrelated datasets create a metaphoric representation of the problem, useful in terms of speed of convergence and predictive results, preserving the fundamental relationships in the data. In order to extract such knowledge, we propose a new learning rule named double backpropagation, by which an auto-encoder concurrently codifies all the different worlds. We test our methodology on different datasets and different issues, to underline the power and flexibility of the Theory of Impossible Worlds. *Published by AIP Publishing.*

https://doi.org/10.1063/1.5024371

This article proposes an approach to the simultaneous analysis of datasets of any kind, related or not, named Theory of Impossible Worlds. The fusion involves two main aspects: first, various different datasets can be merged into a more complex one. Second, each dataset can be rewritten using the hypothetical values of the variables of another one. One of the main reasons why this technique is very useful is the change of viewpoint on the data that, according to the experiments, results in an important improvement in terms of information gain. In the case of classification problems, for example, the results obtained on the fused dataset are better than the ones got by using the original ones. We have hypothesized that the extra point of view works as a sort of metaphor, that makes use of seemingly unrelated, but structurally affine content to improve the understanding of the problem. Moreover, since the fusion process allows us to see each record as belonging to other worlds, we can theoretically build up new relationships. The fusion techniques are essentially of two types: the double backpropagation and the analytic strategy. In the case of the double backpropagation, the datasets are learnt simultaneously from the same auto-encoder. The weights are corrected and updated through a new learning rule called double backpropagation, and presented here for the first time. In the case of the analytic strategy, starting from two distinct datasets to be fused, we have two different auto-encoders A_1 and A_2, equipped with the same number H of hidden units, independently learning the two datasets. Subsequently, the first dataset is rewritten by the variables of the second one using the previously trained A_1 units, and vice versa. In this way, each dataset is projected into a transition world described by H variables, and then rewritten into the variables of the chosen target world.

I. INTRODUCTION

In relational database theory[17] entities can be fused by joining operations.[30] Without filtering (no "where" condition), the join can be a *Cartesian product*, with each resulting combined observation carrying all the attributes of both entities, unless otherwise specified. If, however, there is reason to impose match criteria between some attributes during the join ("where" condition), only filtered observations are generated, which respect the match criteria. The other pairs of observations, for which there has been no match, may or may not be returned, depending on the type of join. In the case of *inner join*, no observation is returned that does not match the defined criteria. In the case of *left join*, all the first entity observations that do not match the second entity ones according to the criteria are also returned. In these cases, attributes of the first entity will be valued, whereas those of the second are set as null. Conversely, in the case of *right join*, all the second entity observations which do not match the first entity ones are also returned, nullifying the attributes of the first entity and repeating the ones of the second. Finally, with the *full outer join*, both left- and right-match types of observations are returned.

Combining data in relational databases is usually done through inner join on keys between tables that represent different aspects of the same phenomenon. The relational keys establish in these cases precise connections between

[a)] Author to whom correspondence should be addressed: m.buscema@semeion.it observations from various entities. Any related observations provide an enrichment of the information content of the others. When relations become weak, inner joins return a few combined observations and, to have results, match criteria must be loosened. In this case, every observation from an entity generally refers to more than one observation from others. The informational content is thus enriched in a weaker way. When the criteria are completely absent, therefore, the Cartesian product represents a fusion of all-with-all, without any substantial reciprocal enrichment of the information content of the observations. When known relationships do not allow to get anything but a mere Cartesian product between two entities, to enrich the information content of both, a different way to cause an interaction between them is called for.

In this regard, one of the most challenging goals in Machine Learning research[24] is data fusion between datasets in terms of attributes (variables) and observations (records), no matter whether the entities refer to the same phenomenon, or are only partially related, or even are not related at all. It is important to emphasize that the in latter case, the coupling of sources related to different phenomena extends the concept of fusion to the connection of different "worlds," and this is the focus of the present paper. Fusion is the possibility of considering two or more separate datasets as one. This type of process can be seen in two different ways, both of which are discussed below. On the one hand, it can be thought of as a new dataset characterized by abstract variables into which both source worlds can be merged. On the other hand, one can think of a transfer of one of the worlds onto the other (or the others), translating it into the corresponding variables of the destination world. In the classical theory of possible worlds and its updates,[6,7,22,23,27,31,32,38] an entity can be transferred from a source world (dataset A) to a destination world (dataset B) if at least one of its attributes is shared in both worlds, but not in case of a void intersection of attributes. In this paper, a way to make such "impossible" transfer actually possible is presented. We have named this approach *Theory of Impossible Worlds* (TIW for short) [The Theory of Impossible Worlds has been developed by M. Buscema and Semeion researchers at Semeion Research Center (Rome, Italy), from 2017 to the present.] to the present.]. It can be applied to both datasets from a same domain, and to entirely heterogeneous datasets. As in human reasoning, where we often use mental models drawn from the analysis of a certain phenomenon to metaphorically infer properties of a completely different phenomenon, we would like to be able to perform the same kind of inferences in the machine learning domain. TIW does exactly this, helping us to empower machine learning environments with the capacity to "fuse" different databases and variables belonging to different phenomena into a same cognitive space, so that it is possible to examine each phenomenon from the structural viewpoint of the other. If this leads to a significant improvement of our capacity to carry out pattern recognition in the specific domains of the source phenomena, then this way of proceeding makes sense and is conceptually useful. We will show that this is actually the case for very diverse examples. The fusion procedure requires a change of viewpoint upon the data that, to a varying extent for different experiments, yields significant gains in terms of information extraction and correct classification. This approach proves to be particularly useful in those social sciences domains where phenomena of interest are not fully amenable to empirical analysis for lack of data on joint occurrences, but where partial data for sub-phenomena are often available.

The idea that information may be considered the real basic constituent of physics research has a long tradition, and has been in recent times strongly advocated by the school of thought originated in the Santa Fe Institute.[4] This idea has provided fertile ground for cross-disciplinary research aimed at finding structural commonalities between phenomena belonging to very different spheres. It has proved especially effective in the application of physics-inspired methods to the social sciences. H. Eugene Stanley coined the term econophysics in 1996 to describe the conceptual and methodological contamination between physics and economics, and in the space of a few years this intuition has led to the emergence of a newly established research field.[29]

The TIW approach that we present in this paper sits in this tradition of thought. In particular, we extend the information-theoretic framework not only to the analysis of information embedded in any dataset generated from any natural or social process, but also to the informational relationships between any such couple of datasets, and in particular, to the detection of "hidden" information structures that may emerge from the analysis of such relationships. Our basic assumption is that relationships between elements of a database (or of a composite database made of many databases) explain the characteristics of the single elements to a better extent than how the characteristics of the single elements explain their relationships: the relational dimension is therefore the prime conceptual dimension of inquiry. Taking this reasoning to the extreme level, we could say, that any element exists only if connected to others through a complex network of weighted relationships. Relationships "explain" elements in the same way as information "explains" energy.

Establishing meaningful relationships between different phenomena has always been a feature of human reasoning and understanding, and here we try and explore the potential of such feature to a new level, as a way to generate deeper insights about the nature and structure of the individual phenomena that we relate to each other.

II. THEORY

A. Definition of the model: The training phase

Definition 1. Given a $M \times N$ dataset, the N attributes are the coordinates that characterize each observation as a *hyper point* of that dataset.

Remark 1. We are allowed to consider each observation as a hyper point since, after all the numerical transformations of any qualitative variables, each dataset can be thought of as a subset $D \subset \mathbb{R}^N$ having cardinality $|D| = M$.

Thus, we can consider attributes as *points of view* from where it is possible to observe the data. The structure of a generic dataset $M \times N$ is shown in Table I.

TABLE I. Generic $M \times N$ dataset.

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix}$$

Definition 2. A procedure that rewrites the attributes of each dataset using a special set of matrix operations to observe the same data from a different point of view is called *data transformation*.

If the data transformation offers a point of view by using a different number of coordinates with respect to the space of origin, then it is called *data projection*.

Example 1. The Principal Component Analysis (PCA)[3] is an example of data transformation. In this case, we get a linear transformation of attributes: a new ordered set of orthogonal variables is generated, with decreasing variances, so that in the new system of coordinates the reading of the data results optimally simplified.

Recently, Artificial Neural Networks (ANNs) AutoEncoders (AE) have qualified as increasingly relevant in deep learning strategies.[5,21,26,34,37,39,40] Auto-encoders are artificial neural networks used in the field of unsupervised learning, generally with the aim to learn the main features of the source data, and to make it possible to encode (and thus decode) them with an economy of information. One of the simplest possible ways to build an auto-encoder is to work with Multi Layer Perceptron (MLP) equipped with at least one hidden layer. and with a number of inputs and outputs equal to the number N of variables in the dataset. The number of hidden units, i.e., the nodes sitting in the hidden layer, may be arbitrary reduced or increased according to needs. The flow diagram illustrating how an auto-encoder can be used to rewrite the dataset is shown in Fig. 1.

An ANN AE can execute a non-linear data projection of the attributes of the original dataset using its hidden units as the new set of coordinates. The number of the hidden units defines the dimensionality of the projection space. It is known[10] that each arrow connecting the different layers of a neural network corresponds to a weight. Initially, the weights are assigned random values, then each of the records of the dataset is plugged into the input layer, and the output is calculated accordingly. Inputs and outputs are then compared, and weights are updated to ensure a closer correspondence between inputs and outputs. The objective of the auto-encoder is to learn to replicate as output what it sees as input, encoding all the fundamental traits of the dataset into the hidden units. At the end of the learning phase, the sum of the squared differences between each input vector of the dataset and its corresponding output vector [Eq. (1)] can be used to define the accuracy of the new hidden coordinates: the closer this amount to zero, the greater the accuracy of the hidden coordinates.

$$Error = \sum_{i=1}^{N} (Input_i - Output_i)^2. \tag{1}$$

As usual, N corresponds to the amount of variables of the data.

Definition 3. Given n datasets $DB_1, DB_2, \ldots, DB_n$ having $N_1, N_2, \ldots, N_n$ attributes and $M_1, M_2, \ldots, M_n$ records, we call *Cartesian dataset* the dataset whose records consist of the Cartesian product $DB_1 \times DB_2 \times \cdots \times DB_n$ of the source datasets, i.e., the n starting datasets.

Remark 2. The number of inputs of the Cartesian dataset is obviously given by the sum of the number of inputs of the datasets we intend to fuse. In fact, if $DB_1 \subset \mathbb{R}^{N_1}$, $DB_2 \subset \mathbb{R}^{N_2}, \ldots, DB_n \subset \mathbb{R}^{N_n}$ then $DB_1 \times DB_2 \times \cdots \times DB_n \subset \mathbb{R}^{N_1+N_2+\cdots+N_n}$. The number $N = \sum_{i=1}^{n} N_i$ is called *input of the Cartesian dataset*. The number of patterns, given by the product $M = \prod_{i=1}^{n} M_i$, is called *cardinality of the Cartesian dataset*. In the following, we will also use the notation $\overline{M}$ to denote the summation of all the n cardinalities $M_1 + M_2 + \cdots + M_n$.

Our *impossible worlds theory* is based upon the idea that ANNs AE can be trained simultaneously on different datasets, despite that both their variables and records have a void intersection, with the primary goal of transporting the elements of the source world onto the others. In this paper, we present two different ways to deal with this problem: the *double backpropagation* and the *analytic strategy*.

1. Double backpropagation

The main idea behind the this approach is the data projection (see Definition 2) of the Cartesian dataset by means of an autoencoder, considering as attributes the Cartesian inputs. One of the most commonly used methods to correct the weights of a MLP auto-encoder is back propagation.[35] In this paper, an innovative training algorithm, named *double backpropagation*, is proposed as an alternative. The innovation of this method is not in the architecture, which in fact remains that of the MLP with the only difference that, for this problem, the input consists of the Cartesian database,

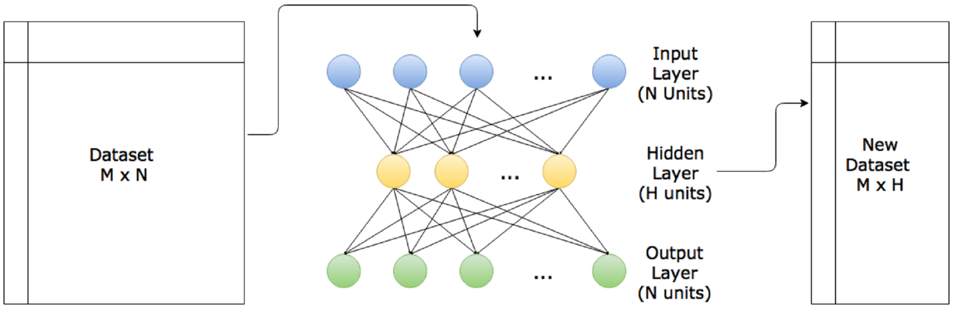


FIG. 1. Example of AE architecture.

that is the juxtaposition of n datasets, instead of a single one, as shown in Fig. 2. To make the representation easier to read, we chose to indicate by a single arrow all the links that connect the input layer to the hidden, and the hidden layer to the output, for each of the datasets. This means that each arrow is intended as the union of all the arrows that start from each of the nodes of DB_a and arrive at each of the H nodes of the hidden layer, in the case of the Input-hidden links, and as the union of the arrows that start from the hidden layer and hit each of the nodes corresponding to DB_b in the case of the hidden-output connections, for all a, $b \in \{1,\ldots, n\}$.

In the case of a three-layer auto-encoder, as the one in Fig. 1, three different types of units shall be considered: $u_j^{[in]}$, input units, $u_j^{[h]}$, hidden units, and $u_j^{[o]}$, output units, where $j \in \{1,\ldots, N\}$ with N equal to the number of variables. Equations (2) and (3) show the usual algorithm to compute the output values. The ℓ superscript indexes the hidden and output layers; in the case of a three-layer MLP, for instance, ℓ may be equal to 1 or 2. The input layer $u_{j_{(t)}}^{[in]} = u_j^{[in]} = x_j$ is sometimes associated with the value $\ell = 0$. The t subscript indexes the iteration of the learning algorithm. The quantities $w_{ji_{(t)}}^{[\ell]}$ correspond to the weights connecting unit i of layer $\ell - 1$ to the unit j of layer ℓ during the iteration t. The function $f(x)$ is named activation, and it is often chosen as the sigmoidal function $f(x) = \frac{1}{1+e^{-x}}$. Equations (4) and (5) calculate the error value at the last layer. The entire output is compared with the target value (the input itself in the case of AE). Then the quantity (4) is calculated as the difference, for each node, between the target and the output times the derivative of $f(x)$ at that point. $\Delta w_{ji}{}_{(t)}^{[2]}$ [Eq. (5)] is the value used to update the weight. Equations (6) and (7) allow the error to be calculated for all the layers preceding the last one—in our case, only one. The r parameter is a learning coefficient chosen according to the difficulty of the problem. By means of Eqs. (8) and (9), weights are updated. The value $M_{[\ell-1]}$ denotes the number of units in the layer $\ell - 1$, with the usual convention such that, when $\ell = 1$, then $\ell - 1 = 0$ is the input layer, so that $u_{i(t)}^{[0]} = u_i^{[0]} = u_i^{[in]}$

$$Net_{j_{(t)}}^{[\ell]} = \sum_{i=1}^{M_{[\ell-1]}} u_{i_{(t)}}^{[\ell-1]} \cdot w_{ji_{(t)}}^{[\ell]} + \theta_{j_{(t)}}^{[\ell]}, \tag{2}$$

$$u_{j_{(t)}}^{[\ell]} = f\left(Net_{j_{(t)}}^{[\ell]}\right). \tag{3}$$

In the case of a 3-layer MLP

$$\Delta out_{j_{(t)}} = \left(target_j - u_{j_{(t)}}^{[o]}\right) \cdot f'\left(Net_{j_{(t)}}^{[2]}\right), \tag{4}$$

$$\Delta w_{ji}{}_{(t)}^{[2]} = r \cdot \Delta out_{j_{(t)}} \cdot u_{i_{(t)}}^{[h]}, \tag{5}$$

$$\Delta\, hidden_{i(t)} = f'\left(Net_{i_{(t)}}^{[h]}\right) \cdot \sum_{j=1}^{M_{[o]}} \Delta out_{j(t)} \cdot w_{ji}{}_{(t)}^{[2]}, \tag{6}$$

$$\Delta w_{ik}{}_{(t)}^{[1]} = r \cdot \Delta\, hidden_{i_{(t)}} \cdot u_{k}{}_{(t)}^{[in]}, \tag{7}$$

$$w_{ji_{(t+1)}}^{[1]} = w_{ji_{(t)}}^{[1]} + \Delta w_{ji}{}_{(t)}^{[1]}, \tag{8}$$

$$w_{ik_{(t+1)}}^{[2]} = w_{ik_{(t)}}^{[2]} + \Delta w_{ik}{}_{(t)}^{[2]}. \tag{9}$$

The reference to back propagation is motivated by Eq. (6), where the error on the last layer is used to calculate the error on the hidden layer. Unlike the classic back propagation, the double back propagation does not consider the output as a whole, but as composed of as many parts as there are datasets. Then, it calculates the error, one dataset DB_p at a time, by propagating the $\Delta out_{j(t)}$ value not only to the hidden layer but also to all the other outputs not belonging to DB_p, as shown in Fig. 3. In this way, it is possible to rewrite Eqs. (6) and (7) as Eqs. (10) and (11), respectively. The same procedure must be repeated until the errors on all the n datasets have been calculated

$$\Delta\, hidden_{i,DB_p\,(t)} = f'\left(Net_{i_{(t)}}^{[h]}\right) \cdot \sum_{j=1}^{M_{[DB_p]}} \Delta out_{j(t)} \cdot w_{ji}{}_{(t)}^{[2]}, \tag{10}$$

$$\Delta w_{ik,DB_p\,(t)}^{[\ell]} = \begin{cases} r \cdot \Delta\, hidden_{i,DB_p\,(t)} \cdot u_{k}{}_{(t)}^{[in]} & if\ \ell = 1 \\ r \cdot \Delta\, hidden_{i,DB_p\,(t)} \cdot u_{k}{}_{(t)}^{[o]} & if\ \ell = 2, \end{cases} \tag{11}$$

where, in this case, k is an index running over all the input units and all the output nodes but the ones of DB_p.

Figure 3 shows the AE weights updating during the learning phase. The output error of the $i-th$ source dataset corrects the hidden-output weights $W^{[2]}(i)$ and, according to the classic chain rule, is backpropagated to the all input-hidden weights $W^{[1]}(\cdot)$. The great novelty of this network is that the error is double-backpropagated also to the output

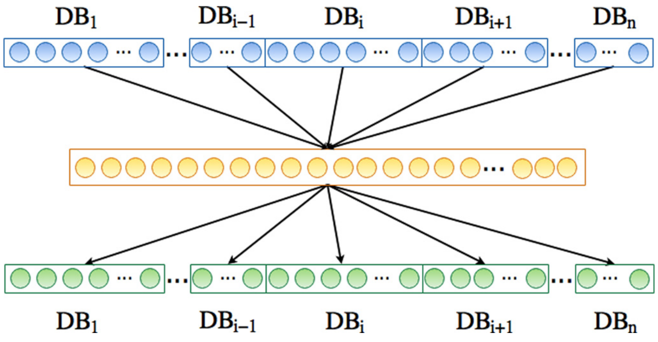


FIG. 2. Example of AE architecture with a Cartesian dataset as input.

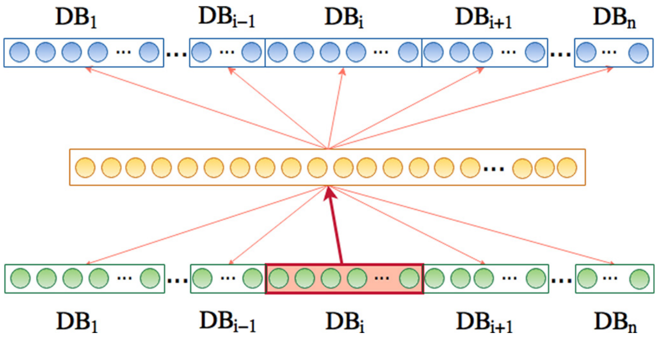


FIG. 3. Error propagation in the case of double back propagation.

weights of the other source datasets $W^{[2]}(k)$, where $k \neq i$. This mechanism causes the weights that connect each data source to the hidden layer to map the different sources in a conjugate way on the same hidden layer. We call this error propagation scheme *double backpropagation*.

To correctly synchronize the learning process, the updating is carried out in two phases: at first, in correspondence of the presentation of a pattern, all the delta values are calculated and stored; then, in a second phase, at the end of the cycle, the weights are updated. It should be noticed that each weight is corrected once, but with the contribution of n corrections per cycle.

At the end of the training phase, i.e., when the AE reached convergence by making errors lower than the maximum threshold set by the user in replicating the input in output, each combination of patterns of the source datasets, i.e., each record of the Cartesian dataset, is represented by a unique hidden vector as in the case of traditional auto-encoders.

Definition 4. The unique hidden vector representing the pth record (pattern) of the Cartesian dataset is called pth *fused component*.

Consequently, all the weights matrices of the AE are the parameters of the fusion of the source datasets in a unique hyper-surface. This interesting aspect will be further explored in the section on applications.

Once the learning process has finished, what we have is an auto-encoder trained on the Cartesian dataset. In order to obtain an actual fusion, it is necessary to introduce the recall phase. Such phase allows the creation of both types of previously introduced fused datasets: a dataset made of virtual variables; and the transfer of any world onto any other.

Definition 5. Given n heterogeneous datasets DB_1, DB_2,…, DB_n having N_1, N_2,…, N_n attributes and the same numbers of records M, we define *Leave one out dataset* with respect to the i^{th} source dataset DB_i the dataset obtained by placing all the source datasets, except DB_i, side by side

$$\overline{\overline{DB_i}} = [DB_1, DB_2, \ldots, DB_{i-1}, DB_{i+1} \ldots, DB_n]. \tag{12}$$

Definition 6. Given n heterogeneous datasets DB_1, DB_2,…, DB_n having N_1, N_2,…, N_n attributes and M_1, M_2,…, M_n records, the dataset $\overline{\overline{DB_i}}(DB_i)$ that represents DB_i considering just the $\overline{\overline{N}} = N_1 + N_2 + \cdots + N_{i-1} + N_{i+1} + \cdots + N_n$ variables of the other $n-1$ datasets, is called ith *Slight Dataset*.

Remark 3. $\overline{\overline{DB_i}}(DB_i)$ represents how the ith Leave one out dataset "sees" DB_i, which corresponds to how each record in dataset i would be expressed using variables from the others.

To build each Slight dataset, we need to input the attributes of any pattern of each source dataset, x_i, $i \in \{1, \ldots, \overline{M}\}$, one at a time, setting all the other inputs to zero, and then consider all the outputs except those corresponding to the dataset itself (see Fig. 4). In other words, for each one of the n datasets, at first, it is necessary to build a new dataset $\mathcal{C}_n$ where each record $x_i^{[\mathcal{C}_n]}$ is expressed according to Eqs. (13) and (14). Then, the previously trained network is interrogated by submitting the new records one at a time

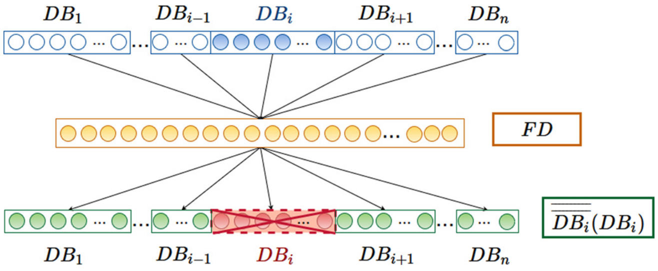


FIG. 4. Recall strategy that allows the creation of the FD and the Slight datasets.

$$x_i^{[\mathcal{C}_n]} = \left[x_i^{DB_1}, x_i^{DB_2}, \ldots, x_i^{DB_n}\right], \tag{13}$$

where

$$x_i^{DB_k} = \begin{cases} \{0\}^{N_k} = \overbrace{\{0, \ldots, 0\}}^{N_k} & \textit{if } x_i \notin DB_k \\ x_i & \textit{if } x_i \in DB_k, \end{cases} \tag{14}$$

with $k \in \{1,\ldots, n\}$.

Remark 4. It should be noted that, given $x_i \in DB_k$, $|x_i| = N_k$ while $|x_i^{[\mathcal{C}_n]}| = N$ and $x_i|_{DB_j} = 0$ when $j \neq k$. Moreover, the order with which the vectors x_i or $\{0\}^{N_k}$ are arranged must reflect the order according to which the Cartesian dataset was created.

Each time a new record $x_i^{[\mathcal{C}_n]}$ is submitted to the network, the activation values of the hidden units, h_i, and the output vector, y_i are saved. If we consider the set of all the $\overline{M}$ hidden vectors saved during the procedure, we obtain the fusion of all the starting n datasets in one, described by the H abstract variables corresponding to the cardinality of the hidden nodes. If, instead, we consider the new dataset consisting of the records described by Eq. (15), we obtain the translation of the $k-th$ dataset into the variables of all the $n-1$ others, i.e., the Slight dataset $\overline{\overline{DB_k}}(DB_k)$

$$x_i^{[\overline{\overline{DB_k}}}(DB_k)] = \bigcup_{\substack{i=1 \\ i \neq k}}^{n} y_i. \tag{15}$$

Remark 5. $x_i^{[\overline{\overline{DB_k}}(DB_k)]}$ represents the $i-th$ pattern of the $k-th$ Slight dataset. In this case $|x_i^{[\overline{\overline{DB_k}}(DB_k)]}| = N - N_k$

Definition 7. The whole output of all datasets given by $S + TIW = DB_i + \overline{\overline{DB_i}}(DB_i)$ is named *Combined Dataset*.

Definition 8. The dataset obtained by collecting all the fused components (Definition 4) relevant to the $\mathcal{C}_n$ dataset is said Fused Dataset–DBP (FD_{DBP}).

We created a new point of view through which we can observe each individual in the light of all the specific attributes of the other datasets.

At the end of the learning phase, for each source dataset, we get four different matrices of data:

- The source dataset $S = DB_i$;
- The slight dataset $TIW = \overline{\overline{DB}}_i(DB_i)$ (i.e., the same records seen from the "point of view" of the other dataset);
- The combined dataset $S + TIW = DB_i + \overline{\overline{DB}}_i(DB_i)$.
- The Fused Dataset FD_{DBP}.

2. Analytic strategy

Although double backpropagation presents an interesting structure, in some cases it has shown a few limitations. When the datasets to be fused are very large, the learning phase could be slow. Furthermore, the double backpropagation strategy allows only the use of multilayer perceptron ANNs. A different approach other than double backpropagation is possible, in order to speed up the learning phase, and to allow the use of different ANNs. This different approach, the analytic strategy, is summarized in Fig. 5.

In this case, we consider two completely different datasets: A, characterized by N_A variables and M_A records, and B, with N_B variables and M_B records, for which we assume no intersection between both variables and records. Each of the datasets is analyzed by means of two independent auto associative neural networks, ANN_1 and ANN_2, which have a hidden layer made of the same number H of nodes, although, obviously, they will take different values as the weights have been trained on completely different datasets. The only precaution necessary to ensure that the fusion will be correct is to start from the same initial configuration of random weights. The auto associative nets work as encoders and decoders of datasets. After the training phase, each record of dataset A is the input of ANN_1, and the relevant hidden layer is saved. The relevant equation is shown in (16), where g_A is the transfer function from the input to the hidden layer (encoding), and A indicates that the weights $w_1^{[A]}$ are being used

$$A_H = ANN_1(A|H_A) = g_A\left(x^{[A]}, w_1^{[A]}\right). \tag{16}$$

At the end of this procedure, we get a new dataset A_H having H variables and M_A records, corresponding to the encoding of the records of A. We use the weights of ANN_2 in order to decode A_H and switch from H variables into N_B. The result of such transformation corresponds to the dataset $\overline{\overline{A}}(A) = B(A)$, composed by the starting M_A records of dataset A expressed by the M variables of dataset B

$$B(A) = ANN_2(A_H|H_B) = f_B\left(g_A\left(x^{[A]}, w_1^{[A]}\right), w_2^{[B]}\right), \tag{17}$$

where f_B is the transfer function from the hidden to the output layer (decoding) of ANN_2. In the same way, each record of the dataset B is encoded into H variables according to the relevant auto associative net. We get a $M_B \times N_A$ dataset corresponding to the point of view of A over B.

Equations (18) and (19) illustrate the traditional way by means of which neural networks produce their outputs. Equations (20) and (21), in contrast, show the innovative way through which one can mix up the weight matrix and get the output of dataset A from the point of view of dataset B

$$y^{[A]} = f_\alpha\left(g_\alpha\left(x^{[A]}, w_1^{[A]}\right), w_2^{[A]}\right), \tag{18}$$

$$y^{[B]} = f_\beta\left(g_\beta\left(x^{[B]}, w_1^{[B]}\right), w_2^{[B]}\right), \tag{19}$$

$$y^{[B(A)]} = f_{\alpha(\beta)}\left(g_\beta\left(x^{[B]}, w_1^{[B]}\right), w_2^{[A]}\right), \tag{20}$$

$$y^{[A(B)]} = f_{\beta(\alpha)}\left(g_\alpha\left(x^{[A]}, w_1^{[A]}\right), w_2^{[B]}\right). \tag{21}$$

This procedure actually allows us to obtain both types of fusion we had in mind. Considering the concatenation of H_A and H_B, we obtain the fusion of the starting datasets expressed in terms of virtual variables not belonging to either of the two worlds. If we consider A(B) and B(A), we obtain the so-called transfer of one world onto another.

Thus, also in this case, we will have four different data matrices:

- The source dataset $S = DB_i$.
- The slight dataset $TIW = \overline{\overline{DB}}_i(DB_i)$ [i.e., the same records seen from the "point of view" of the other dataset

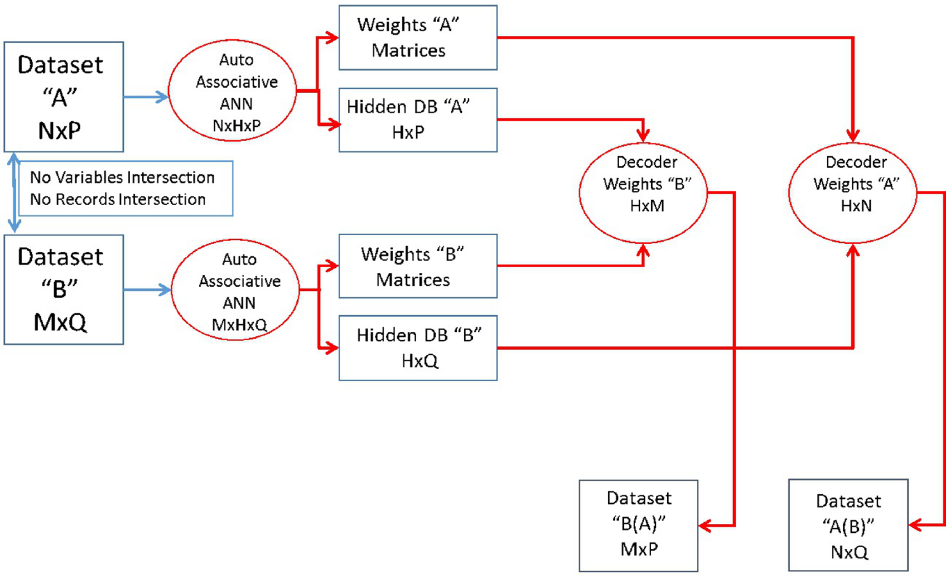


FIG. 5. Impossible world flow chart using the Analytic Strategy ($n = 2$).

corresponding to corresponds to the juxtaposition of all the $i-th$ records of the datasets built as in Eqs. (20) and (21)].

- The combined dataset $S+TIW = DB_i + \overline{\overline{DB_i}}(DB_i)$.
- The fused dataset FD_{AS}.

Comparing the obtained datasets with the definitions of Slight Dataset, Combined Dataset and Fused Dataset previously introduced (Definitions 6, 7, and 8), a difference should be noticed. Although the first two definitions are still valid, the definition of Fused Dataset in this case does not work. In fact, the FD_{AS} dataset was built by considering the records of the source datasets rewritten by the hidden units of the different auto-encoders. It therefore seems reasonable to slightly modify definition 8 to make it more general.

Definition 9. Given n heterogeneous datasets DB_1, $DB_2,\ldots, DB_n$ having $N_1, N_2,\ldots, N_n$ attributes and $M_1, M_2,\ldots, M_n$ records each, the dataset composed by all the source records rewritten by the hidden units of one or more auto-encoders used to learn the n source datasets according to any of the two strategies is named *Fused Dataset (FD)*.

B. The transfer fitness

An evaluation of the fitness of the combination of heterogeneous datasets (impossible worlds transfer) according to the procedure described above is possible if each one of the source datasets to be combined has an independent target or, in other words, if we can put each one of these datasets through a supervised learning process. If such independent targets condition is satisfied, then we need to apply a K Cross Validation[25] test for each of the source datasets (a supervised Multilayer Perceptron may be used). We repeat the same K Cross Validation test for each dataset using the new attributes generated by the fusion procedure (the values of the variables that each dataset will take according to the variables types of the other datasets, i.e., S and $S+TIW$ datasets), in order to compare the results. Figures 6 and 7 show a summary of the validation procedure.

The analysis of fitness may be done for two different purposes:

1. In order to test which records maintain, reduce or increase their informational content, passing from one dataset to another one;
2. In order to test whether adding to the source input of a dataset the new attributes that it inherits from the other datasets, its global information content increases in a significant way.

Definition 10. Given n heterogeneous datasets DB_1, $DB_2,\ldots, DB_n$ having $N_1, N_2,\ldots, N_n$ attributes, $M_1, M_2,\ldots, M_n$ records and $t_1, t_2,\ldots, t_n$ independent targets, we define as *TIW Fitness* the hyperbolic tangent of the difference between the real and the transferred accuracy

$$F(DB_i) = \tan h(TIW_i - Real_i), \tag{22}$$

$$= \frac{e^{(TIW_i - Real_i)} - e^{-(TIW_i - Real_i)}}{e^{(TIW_i - Real_i)} + e^{-(TIW_i - Real_i)}}, \tag{23}$$

where TIW_i corresponds to the accuracy attained by performing the classification of the i^{th} dataset after the fusion procedure, while $Real_i$ is the original accuracy.

The choice of this specific function is linked to its properties. When $TIW_i > Real_i$, $F(DB_i)$ grows rapidly towards 1, whereas if $TIW_i < Real_i$, $F(DB_i)$ takes opposite values close to -1. In case $TIW_i \approx Real_i$ then $F(DB_i) \approx 0$, so the transformation did not bring any particular advantage in terms of correct classification.

C. A step forward

Once the fusion phase is terminated, the trained AE is able to generate a huge number of simulations of dynamic scenarios, according to the What If Theory.[16] When a subset of inputs of the combined dataset is constrained to specific

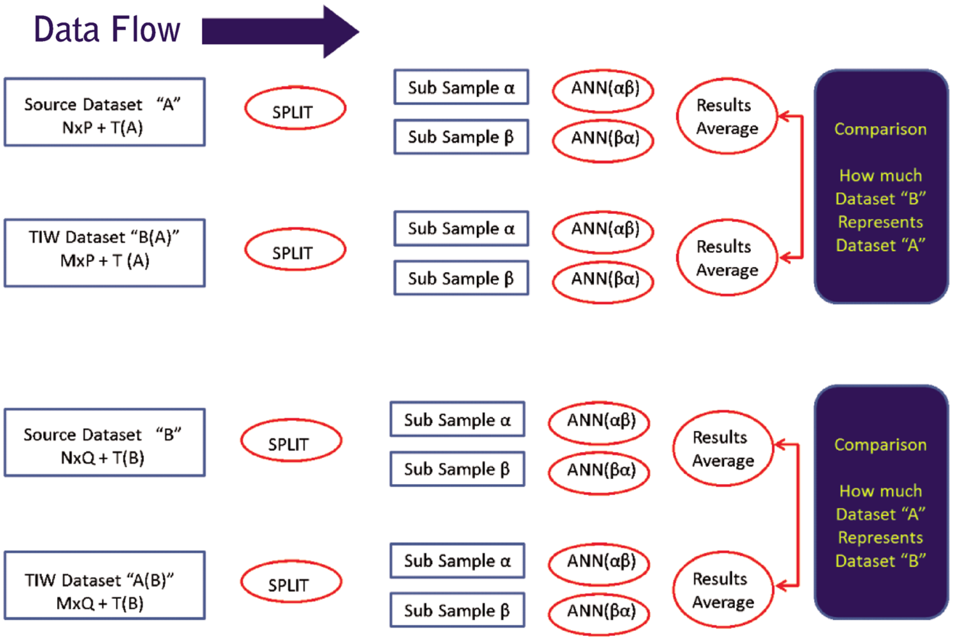


FIG. 6. TIW validation procedure where the datasets to be combined are two. The two branches show how to compare the classification performances obtained by the source dataset and by the transferred one.

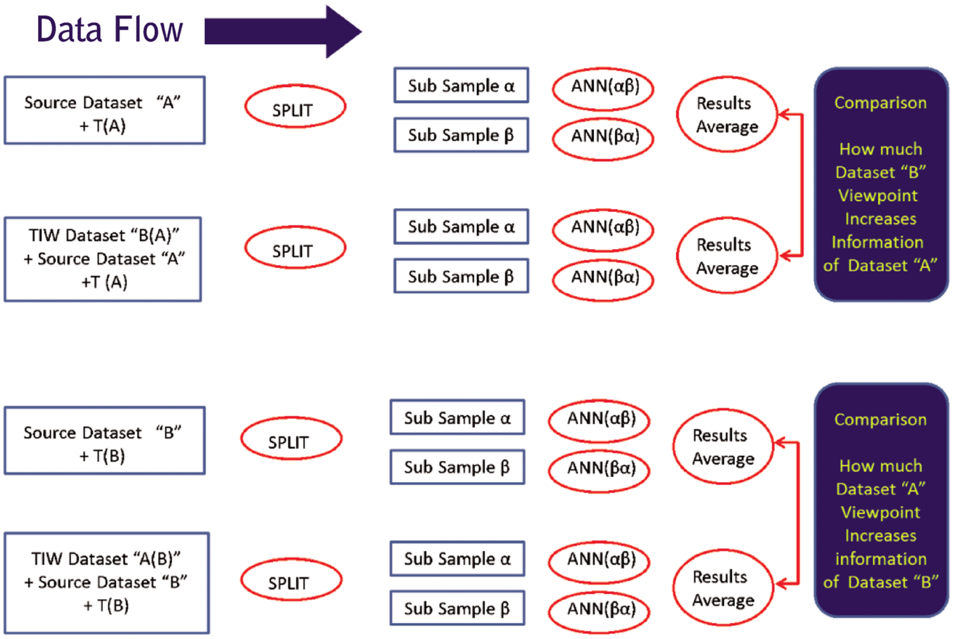

Data Flow
Source Dataset "A" + T(A)
SPLIT
Sub Sample α
Sub Sample β
ANN(αβ)
ANN(βα)
Results Average
TIW Dataset "B(A)" + Source Dataset "A" +T (A)
SPLIT
Sub Sample α
Sub Sample β
ANN(αβ)
ANN(βα)
Results Average
Comparison
How much Dataset "B" Viewpoint Increases Information of Dataset "A"
Source Dataset "B" + T(B)
SPLIT
Sub Sample α
Sub Sample β
ANN(αβ)
ANN(βα)
Results Average
TIW Dataset "A(B)" + Source Dataset "B" + T(B)
SPLIT
Sub Sample α
Sub Sample β
ANN(αβ)
ANN(βα)
Results Average
Comparison
How much Dataset "A" Viewpoint Increases information of Dataset "B"

FIG. 7. TIW validation procedure where the datasets to be combined are two. In this case, the two branches show how to compare the classification performances obtained by the source dataset and by the combined one.

values, the AE is dynamically able to adjust the values of all the other variables in a finite number of cycles. The convergence point reached by the trained AE represents the implied scenario for the initial choice of constrained values.

Definition 11. The convergence hyper-point reached by the trained AE is called *Metaphoric Point*.

Remark 6. We have called it "Metaphoric" since the system is creating a scenario, i.e., is answering a question, by providing a hyper-point belonging to the "Cartesian world." It means that the system is mixing together different (and perhaps unconnected) worlds to make the answer clearer, exactly as a human does by means of metaphoric examples to clarify a concept.

According to this strategy, a trained AE, with the help of specific algorithms, such as Spin Net,[8,15] can work as a Content Addressable Memory, mixing up variables and records that originally belonged to different, unrelated datasets, into a new, common causal process. The typology of problems that a trained AE, used as dynamic associative memory, can cope with has already been studied for simple cases.[15] A same AE for the purpose of TIW may be implemented using different ANNs: the traditional Multilayer Perceptron with backpropagation learning rule is the most common choice. Also a New Recirculation ANN[9] is suitable, and maybe more suitable than backpropagation,[20] for this kind of task. The AutoCM ANN[15] has shown also to perform pretty well in auto associative learning, and it is very efficient as well as a Content Addressable Memory using its specific adaptive algorithm, the above-mentioned Spin Net, for the recall phase.

III. APPLICATIONS

In this section, different applications of the theory of impossible worlds will be presented. First, we consider supervised datasets and classification issues. Four different tests are reported. Two of them deal with pairs of datasets belonging to the same domain—Digits and Credit scoring—whereas the third and the fourth ones consider instead data from couples of completely different, well-known problems: Parity-Negation and Parity-Spirals. Subsequently, we apply TIW in an unsupervised context, in order to show that fused datasets preserve the knowledge stored in the data. We choose to present several different applications to illustrate the flexibility of application and the peculiar features of the method, so that readers can appreciate aspects that especially concern their particular research interests.

A. Examples of supervised applications

Supervised applications are those where the main task of the artificial intelligence algorithm is to learn to recognize the correct gold standard, called target, for each record in the dataset.

1. Hand-written digits

The first application of TIW we present considers two different datasets of hand written digits. The first one is made of 5620 digits in a 8 × 8 grid (64 inputs coded as integers between 0 and 16).[28] The second is made of 1593 digits in a 16 × 16 grid (256 Boolean inputs).[36] The two datasets are completely different (input length, patterns and coding format), but the target is the same (one of ten possible outputs). Some patterns selected from the datasets are shown in Fig. 8. The target for each record corresponds to the class to which it belongs: zero, one, …, nine.

We projected the 64 inputs onto 256 ($D_{64} \rightarrow D_{256}$), and the 256 onto 64 ($D_{256} \rightarrow D_{64}$). After removing the targets from D_{256} and D_{64}, we considered their Cartesian Dataset. We used it to train a back propagation AE, according to the fusion strategy (Double Back Propagation) presented above (see Sec. II A 1, Figs. 2 and 3).

At the end of the learning process, for each source dataset, we are interested in the analysis of the following three different matrices of data:

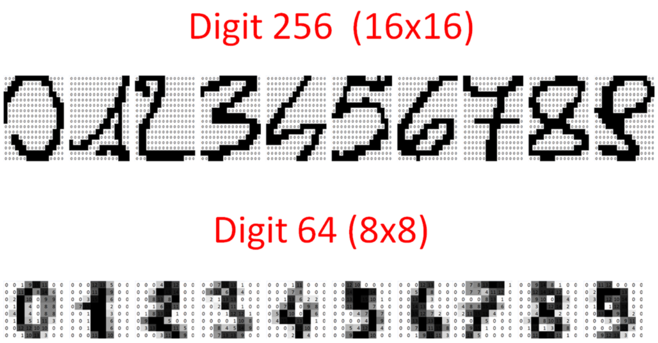


FIG. 8. Random samples of hand written digits taken from the two datasets.

- The original dataset $S = D_{64}$ (respectively D_{256});
- The slight dataset $TIW = \overline{D_{64}}(D_{64}) = D_{256}(D_{64})$ (respectively $\overline{D_{256}}(D_{256}) = D_{64}(D_{256})$) (i.e., the same records seen from the "point of view" of the other datasets);
- The combined dataset $S + TIW = D_{64} + D_{256}(D_{64})$ [respectively $D_{256} + D_{64}(D_{256})$].

We can now carry out the same classification task in multiple rounds: once upon the original data, once again upon the transferred data, and finally upon the fused components matrix for each of the datasets, and then compare the results. The standard steps for a classification are: divide the sample into training and testing sets; train the network by repeatedly exposing it to the records in the training set and to the correct targets; ask the net to predict the target of records belonging to the testing set. To make the experiments comparable, we have split the datasets into two halves (training and testing set) using the same criterion for each group. In particular, S, TIW and S + TIW share the same subset A for training, and the subset B for blind testing (and subsequently the other way round). Table II shows the results for D_{64}, whereas Table III shows results for D_{256}.

Remark 7. Both tables report in the first column the name of the analyzed dataset, the ten targets to estimate, the global accuracy, the number of the errors, and the error standard deviation. The other columns report the average accuracy of the double blind test validation (training on subset A, testing on subset B, and the other way round) for each dataset. Column 2 provides the results from using the source dataset; Column 3 the results from using only the new TIW input; and finally Column 4 the results from using the source input and the new TIW input together (i.e., the input of the Cartesian dataset).

TABLE II. $D_{64} \rightarrow D_{256}$ results.

$D_{64} \rightarrow D_{256}$	S	TIW	S + TIW
0	98.87	98.30	98.30
1	98.93	99.44	99.46
2	99.45	97.71	99.45
3	97.30	98.93	98.39
4	98.36	98.93	98.93
5	96.70	96.70	98.90
6	98.35	97.25	97.80
7	98.88	98.33	98.88
8	90.21	94.23	94.81
9	92.25	92.24	92.80
Global Acc.	**96.93**	**97.21**	**97.77**
Errors	27.50	25	20
St. Dev.	6.4	2.8	7.1

TABLE III. $D_{256} \rightarrow D_{64}$ results.

$D_{256} \rightarrow D_{64}$	S	TIW	S + TIW
0	96.93	82.56	96.93
1	91.52	58.35	93.06
2	91.84	58.45	91.84
3	88.47	64.43	89.69
4	88.27	56.53	89.38
5	91.94	50.08	91.37
6	97.51	64.00	95.54
7	87.96	47.41	88.45
8	87.94	54.67	90.80
9	89.22	48.80	89.86
Glob. Acc.	**91.16**	**58.53**	**91.69**
Errors	70.5	329	66
St. Dev.	14.85	9.90	22.63

The results show that the transformation of the dataset 64 × 10 into the 256 variables of the other dataset generates an equivalent or more informative dataset for pattern recognition, whereas the transformation of the 256 × 10 into 64 variables generates a dataset that is less informative for supervision tasks. This means that the logic of the 256 × 10 dataset applied to the 64 × 10 dataset is effective, but not vice versa. Figure 9 shows how some patterns of the 64 × 10 dataset are projected onto the 256 dimensions of the other dataset.

These results also show that when we augment the original input with the new one generated by TIW [augmented input: $DB_i + \overline{\overline{DB_i}}(DB_i)$], the pattern recognition capacity for both datasets increases significantly. That is to say: when we observe the same phenomenon from different points of view simultaneously, our understanding improves. Table IV shows the relevant fitness values.

The fitness table confirms our remark. The $D_{64} \rightarrow D_{256}$ transformation yields an improvement in terms of results, whereas the $D_{256} \rightarrow D_{64}$ one definitely does worse with respect to the source dataset. What is interesting to notice is that in both cases there is an improvement in prediction when the S + TIW dataset is used. This example, therefore, shows how in the transition from one world to another, classification capacity can improve or worsen. What is particularly surprising is that even if the transfer results in a worse performance, the combined data*set* always improves.

2. Credit scoring datasets

In this experimentation, we consider two datasets reporting good vs. bad payers of two different banks: an Australian bank,[1] and a German one.[2] The Australian dataset is made of 14 inputs (not declared) and 690 patterns (383 good payers and 307 bad payers). The German dataset is made of 20 inputs and 1000 patterns (700 good payers and 300 bad payers). Differently from the previous case, we choose Auto CM NN[15] as the AE, and adopt the analytic strategy while

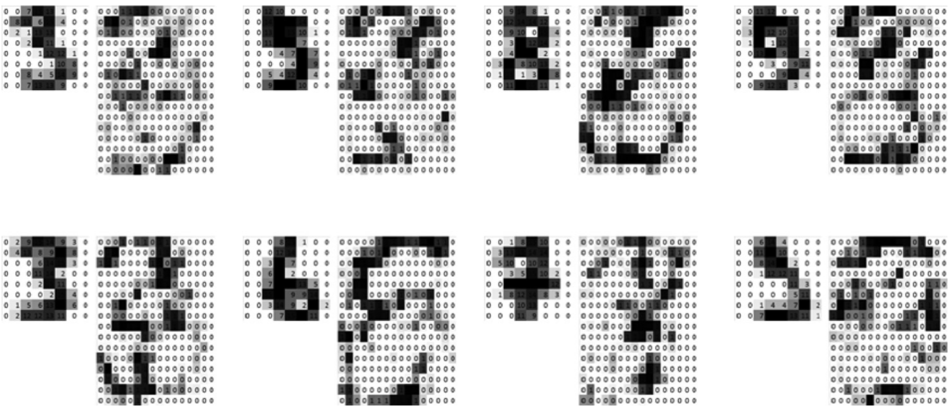


FIG. 9. How the 16×16 digits (256×10 dataset, the big digits) "see" the 8×8 digits (64×10 dataset, the smaller digits). First line from left: 3, 5, 8, 9. Second line from left: 3, 6, 7, 9.

TABLE IV. Fitness table for D_{64} and D_{256}.

Source Dataset	TIW	S + TIW
D_{64}	0.273	0.686
D_{256}	−1	0.485

TABLE V. $Cr_{AU} \rightarrow Cr_{GE}$ results.

	$Cr_{AU} \rightarrow Cr_{GE}$	S	TIW	S + TIW
ANN_1	Bad payers	90.91	90.91	90.26
	Good payers	86.91	86.39	90.58
	Mean acc.	**88.91**	**88.65**	**90.42**
	Weighted acc.	**88.70**	**88.41**	**90.43**
	Errors	39	40	33
ANN_2	Bad payers	88.89	94.12	90.85
	Good payers	82.81	78.12	83.85
	Mean acc.	**85.85**	**86.12**	**87.35**
	Weighted acc.	**85.51**	**85.22**	**86.96**
	Errors	50	51	45
ANN_{avg}	Bad payers	89.90	92.52	90.56
	Good payers	84.86	82.26	87.22
	Mean acc.	**87.38**	**87.39**	**88.89**
	Weighted acc.	**86.96**	**86.96**	**88.70**
	Errors	45	45	39

TABLE VI. $Cr_{GE} \rightarrow Cr_{AU}$ results.

	$Cr_{GE} \rightarrow Cr_{AU}$	S	TIW	S + TIW
ANN_1	Bad payers	76.29	74.29	76.57
	Good payers	65.33	65.33	68.67
	Mean acc.	**70.81**	**69.81**	**72.62**
	Weighted acc.	**73.00**	**71.60**	**74.20**
	Errors	135	142	129
ANN_2	Bad payers	77.14	76.57	79.71
	Good payers	70.67	69.33	72.67
	Mean acc.	**73.90**	**72.95**	**76.19**
	Weighted acc.	**75.20**	**74.40**	**77.60**
	Errors	124	128	112
ANN_{avg}	Bad payers	76.72	75.43	78.14
	Good payers	68.00	67.33	70.67
	Mean acc.	**72.36**	**71.38**	**74.41**
	Weighted acc.	**74.10**	**73.00**	**75.90**
	Errors	129.5	135	120

following the usual TIW protocol already described [i.e., (i) Remove the targets from the two datasets; (ii) Fusion learning by an Auto Encoder (Auto CM); (iii) Independent cross validation of the two source datasets using supervised ANNs; (iv) Independent cross validation of the transformed datasets with the new inputs (TIW), using supervised ANNs; (v) Independent cross validation of the two datasets with an augmented input vector—source inputs + TIW inputs—using supervised ANNs].

Tables V and VI show analogous results with respect to the previous experiment. Using only the "point of view" of each dataset on the other (TIW), we obtain, more or less, the same results we get by using the source datasets (S). If we combine the two input vectors (S + TIW), global accuracy increases. Table VII shows the fitness values of both transformations.

Furthermore, TIW has a strong potential for applications in socio-economic research. As seen in the example, it allows to synergize the informational content of non-conformable databases covering analogous phenomena. It can also enable the application of powerful "metaphorical" thinking to databases covering very different phenomena, including some largely overlooked by policy analyses, but that might become more relevant once their potential connections to more acknowledged ones are explored from multiple "viewpoints." A relevant example is the emerging field of research on culture-driven crossovers, namely, the still under-developed investigation of the structural relationships between active cultural participation (a variable that has so far received very little attention in policy analyses) and a variety of diverse, highly relevant policy domains such as health; innovation; environmental sustainability; social cohesion, etc.[33] On another note, TIW may be very useful in cases where abundant data exist on two separately relevant

TABLE VII. Fitness table for Cr_{AU} and Cr_{GE}.

Source Dataset		TIW	S + TIW
Cr_{AU}	ANN_1	−0.254	0.907
	ANN_2	0.264	0.905
	ANN_{avg}	0.005	0.906
Cr_{GE}	ANN_1	−0.762	0.948
	ANN_2	−0.740	0.980
	ANN_{avg}	−0.753	0.967

phenomena whose mutual relationship is still ill-understood (e.g., gender inequality and education, or organizational behavior; consumer choice and psychological well-being, etc.).

In this case, although the TIW fitness is not impressive, the S + TIW fitness is pretty high. This example, besides confirming what was observed in the previous application, shows a further characteristic of the Theory of Impossible Worlds. In this case, the 14 variables of the Cr_{AU} dataset were not declared. By transferring from one world to another, however, it is possible to see the TIW-reconstructed value of the variables that Australian payers would face in the world Cr_{GE}, thus obtaining additional information.

3. Parity4 and Negation4

In this experimentation, two independent datasets have been used.

The *Parity* dataset is a small dataset made of 16 records, 4 boolean variables and the target $t \in \{0, 1\}$. For each record p_i, $t_i = 1$ if the occurrences of values equal to 1 in p_i is an odd number, and 0 otherwise.

The *Negation* dataset has 16 records, 4 boolean variables and the target $\mathbf{t} \in \{0, 1\}^3$. For each record $n_j = (n_{j1}, n_{j2}, n_{j3}, n_{j4})$, if $n_{j1} = 0$ then $\mathbf{t} = (n_{j2}, n_{j3}, n_{j4})$, whereas if $n_{j1} = 1$ then $\mathbf{t} = (1 - n_{j2}, 1 - n_{j3}, 1 - n_{j4})$.

Both problems are famous for being challenging in terms of convergence of learning processes. TIW can help also for this kind of problem. The fusion of datasets has been carried out with three different kinds of auto-encoders: Auto CM NN,[15] Multi Layer Perceptron (MLP),[35] and New Recirculation NN (NRC).[9] Subsequently, a MLP with three hidden units has been used to learn the following four datasets: the source dataset S, the CM fused dataset (TIW_{CM}), the MLP fused dataset (TIW_{BP}), and the NRC fused dataset (TIW_{NRC}). Finally, the results have been comparatively analyzed.

In the first case (S dataset used as input to the MLP), the Root Mean Square Error (RMSE) value remained pretty constant and equal to 35%, as shown in Fig. 10(a), and no learning occurred. The training was stopped after more than 6000 epochs.

In the case of TIW_{NRC}, an interesting error reduction has been observed. After a few epochs, the RMSE value dropped from 35% to 15%. Furthermore, after about 2200 epochs, the algorithm quickly converged to zero error [Fig. 10(b)].

The cases of TIW_{BP} and TIW_{CM} are even more interesting, since learning occurred after about 1000 and 500 epochs, respectively, as reported in Figs. 10(c) and 10(d).

It should be noticed that in Fig. 10(a), the delta values of hidden units were close to zero, while the ones for the output were high. This implies that MLP was wrong on the output and no learning took place. In Figs. 10(b)–10(d), after a starting phase analogous to that of Fig. 10(a), the delta values for the output went down to zero and the hidden units started to learn. This example shows how TIW can be useful in improving convergence speed. It also stresses the fact that no noise is being added, but only real, hidden information.

4. Spirals and Parity8

The *Spirals* dataset is a 384 × 2 dataset representing the coordinates $(x, y) \in \mathbb{R}^2$ of points belonging to two concentric spirals (see Fig. 11). The target is a Boolean value characterizing the spirals: if $t = 1$ then the point belongs to spiral $\mathcal{A}$, if $t = 0$ the point belongs to spiral $\mathcal{B}$. *Parity8* is equivalent to the dataset described above in III A 3, but involving 8 bits instead of 4.

The same analysis of Sec. III A 3 has been carried out and, also in this case, TIW proved to be important for convergence as reported in Fig. 12. The source dataset RMSE starts to decrease after 700 epochs, although slowly, whereas already after 100 epochs the S + TIW's RMSE is significantly reduced.

By observing how inputs from Parity see Spirals data after the fusion, an original behaviour has been noticed. The fused dataset has been linearly scaled into the range [0; 1] to make the values comparable, and the records belonging to spiral $\mathcal{A}$ have been separated from those in $\mathcal{B}$. Then, the linear correlation among rewritten parity input and the appropriate spiral coordinates has been computed, and the relevant values are reported in Table VIII.

Table VIII shows low correlation among parity inputs and the y values, and a pretty highly negative correlation with the x coordinates. Then, parity input and the spirals have been crossed over, so that the correlation between the 8 variables of parity of points belonging to $\mathcal{A}$ and the input of $\mathcal{B}$—and vice versa—can be computed. Results are shown in Table IX. In this case, the correlation sign changes and, especially for In_3, In_5 and In_8 the values are rather large in module. This "reversed" correlation appears evident by plotting the input of Spirals and comparing them to the input of Parity as shown in Figs. 13(a), 13(b), and 13(c), where the values of In_3, In_5, and In_8 are compared to the input of $\mathcal{B}$. This relationship seems to suggest that each Spiral can learn from Parity something about the other one, and that this fused knowledge plays a relevant role during convergence. It seems like the Spirals' input convey learning about which is the right spiral for each record, while the Parity ones about which one is not. The fusion of these two different but complementary types of knowledge speeds up learning.

B. An example of an unsupervised application

Unsupervised applications are problems for which no targets are introduced, and the purpose of the research is to understand the deep structure of the data under study.

1. Gang and food

In this section, two unsupervised datasets are analyzed to show how the impossible world transfer preserves the relationships already present in the data. *Gang* is a small database made of 27 records corresponding to members of the "Jets vs. Sharks" gangs taken from the West Side Story musical, and five variables that we use to characterize each of them (name of gang member, gang name, age range, level of education, marital status and occupation). The dataset is summarized in Table X.

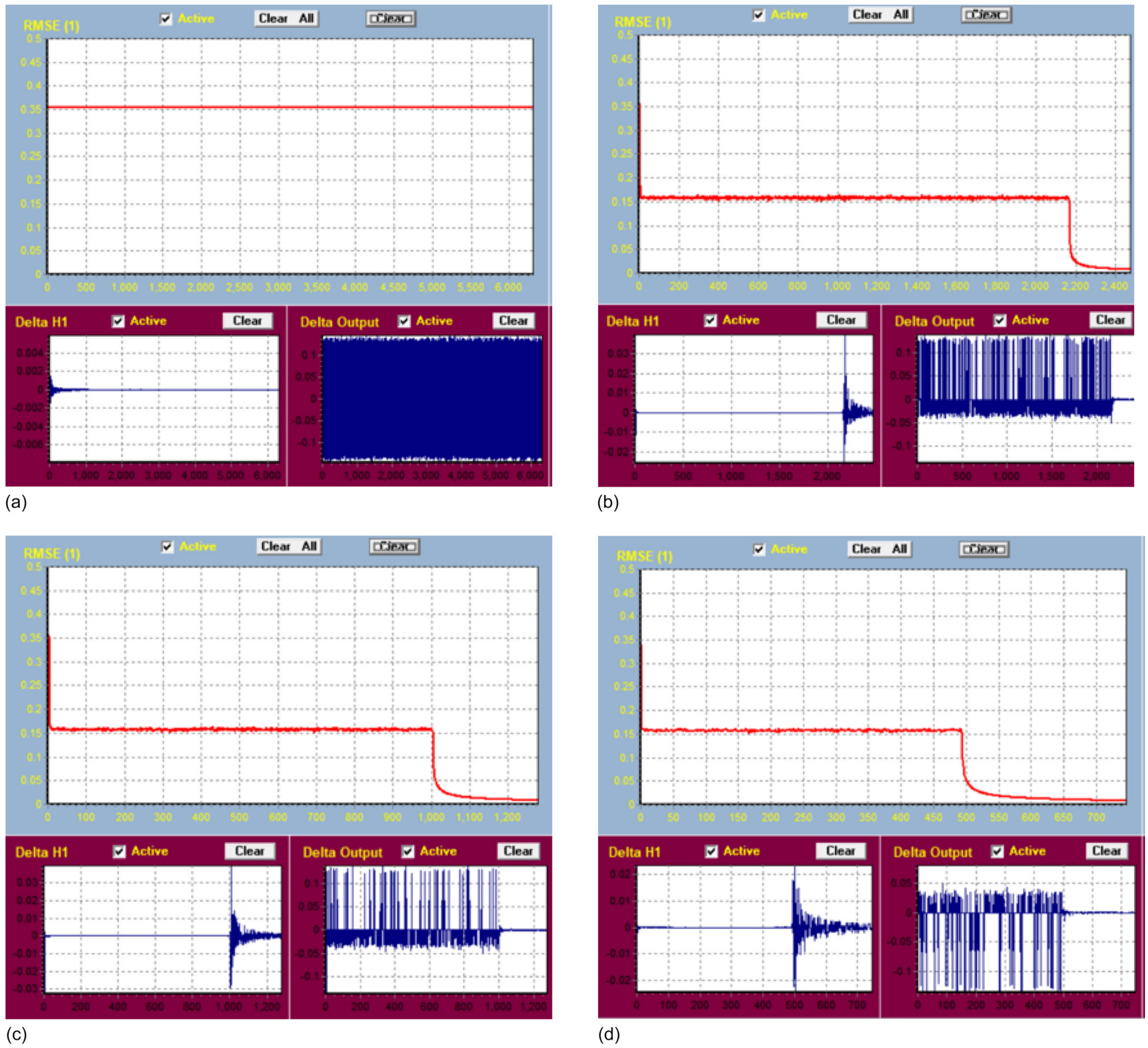


FIG. 10. (a) Source Dataset: RMSE error curve and the delta values of hidden and output units. (b) TIW_{NRC} Dataset: RMSE error curve and the delta values of hidden and output units. (c) TIW_{BP} Dataset: RMSE error curve and the delta values of hidden and output units. (d) TIW_{CM} Dataset: RMSE error curve and the delta values of hidden and output units. SOURCE: Semeion Software n.12 ver.29.1 Semeion.

Food is a 16 × 9 dataset that reports the amount of 9 different kinds of foods eaten in 16 European countries (see Table XI).[13,14]

After having fused the data by means of the double back propagation method, a new dataset $TIW_{G\text{–}F}$ made of 43 variables and $14 \times 9 = 126$ records has been created. $TIW_{G\text{–}F}$ has been processed with Auto CM NN[15] to obtain the relevant similarity graph,[15] as shown in Fig. 14. The similarity graph is the Minimum Spanning Tree (MST),[18] whose weights are derived from the Auto CM NN learning weights.[15]

The graph seems to show that the geographical relationships are preserved. On the right, one finds the main countries of the Mediterranean diet: Italy, Greece and Spain. In the central part, Portugal, Belgium, Germany, France, Austria and the Netherlands are strongly linked to each other. Great Britain and Ireland, as well as Sweden and Norway, are very close. Furthermore, in the lower part of the graph, there are Denmark, Finland and Iceland.

Another interesting aspect is shown in Tables XII and XIII representing each dataset through the point of view of the other. So we could say, in a metaphorical sense, that if Italy were a gangster, he would be a 30-year-old high school educated married pusher, and if Clyde were a country, milk would be a very popular drink. So, as noticed before, TIW allows to reason about a world by using attributes from another one, exactly like a metaphor does.

IV. A STEP BY STEP EXAMPLE

Whereas the procedure introduced in this paper is rather sophisticated and complex, this section is devoted to the step-by-step explanation of a typical example in which the

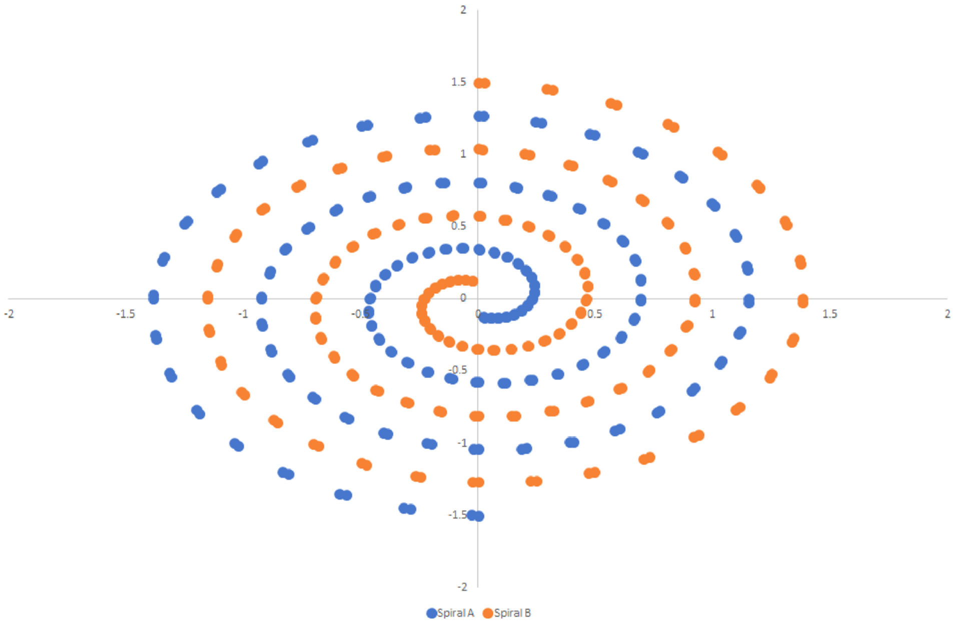


FIG. 11. Spirals dataset graphic representation.

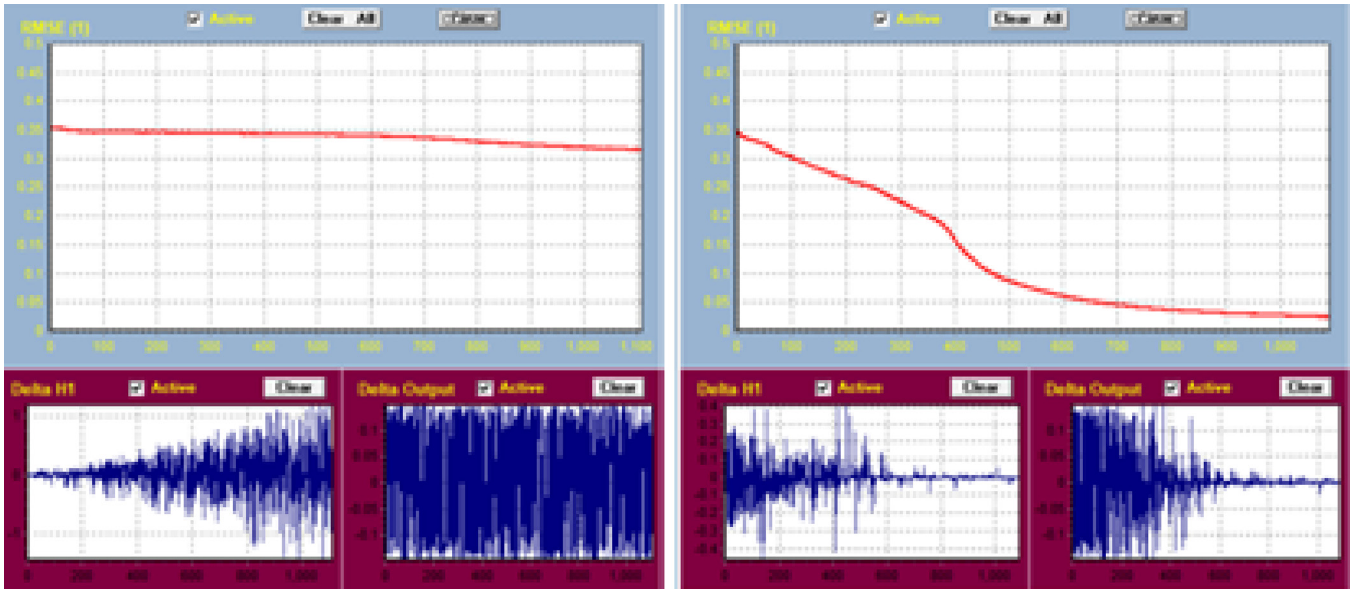


FIG. 12. RMSE error curves and delta values of hidden and output units of Source (on the left) and TIW datasets (on the right). SOURCE: Semeion Software n.12 ver.29.1 Semeion.

TABLE VIII. Linear correlation among the input of one of the spirals and the parity input of points belonging to it.

Parity input	$\mathcal{A}_x$	$\mathcal{A}_y$	$\mathcal{B}_x$	$\mathcal{B}_y$
In_1	−0.04	−0.24	0.06	0.27
In_2	−0.68	0.45	−0.64	0.49
In_3	−0.88	0.19	−0.92	0.22
In_4	−0.31	−0.02	0.42	0.01
In_5	−0.89	0.06	−0.94	0.08
In_6	−0.81	0.21	−0.84	0.28
In_7	−0.51	−0.01	−0.24	−0.18
In_8	−0.83	0.24	−0.87	0.28

whole methodology is used. In this case we used the analytic strategy (Sec. II A 2).

A. Step 1: Datasets

In order to apply the Theory of Impossible Worlds, it is necessary to have at least two different datasets to join. As has been repeatedly shown, there is no constraint on datasets, that may have no intersection for both variables and records. In addition, the two datasets can belong to completely separate universes. In this case, we chose:

Food: The same 9 × 16 dataset seen in III B, with the addition of three targets that correspond to the geographical position occupied by the various European countries (Table XIV). The variables are: Cereals, Rice, Potatoes, Sugar, Vegetables, Meat, Milk, Butter, Eggs. The targets are: Center Europe, Scandinavian (North), Mediterranean (South).

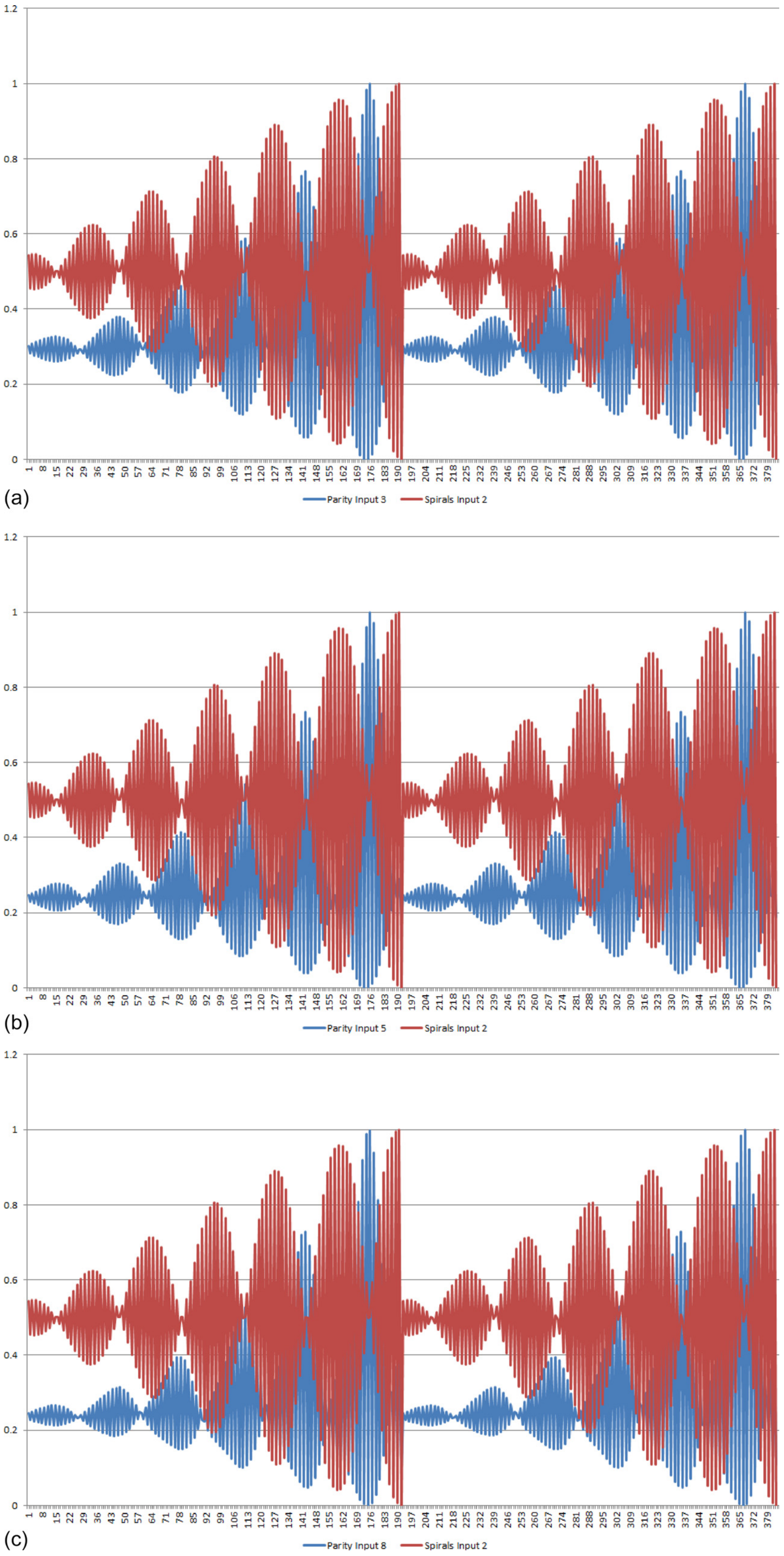


FIG. 13. Comparison between the less correlated input of Parity and the y coordinate of spiral $\mathcal{B}$. (a) Plot of In_3 vs spiral $\mathcal{B}_y$. (b) Plot of In_5 vs spiral $\mathcal{B}_y$. (c) Plot of In_8 vs spiral $\mathcal{B}_y$.

Zoo: A dataset composed of 99 species of animals with 16 attributes and belonging to 7 different types that will be considered as targets. The variables are: hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, legs, tail, domestic, catsize. The targets are: Mammal, Bird, Reptile, Fish, Amphibian, Bug, Invertebrate.[19] A little sample of the dataset is shown in Table XV.

B. Step 2: The training phase

We train independently the two or more datasets using two or more auto-encoders having the same number of

TABLE IX. Linear correlation among the input of one of the spirals and the parity input of points belonging to the other.

Parity input	$\mathcal{A}_x$	$\mathcal{A}_y$	$\mathcal{B}_x$	$\mathcal{B}_y$
In_1	−0.06	−0.27	0.04	0.24
In_2	0.65	−0.49	0.68	−0.45
In_3	0.92	−0.22	0.88	−0.19
In_4	−0.42	−0.01	0.31	0.02
In_5	0.94	−0.08	0.89	−0.06
In_6	0.84	−0.28	0.81	−0.21
In_7	0.24	0.18	0.51	0.01
In_8	0.87	−0.28	0.83	−0.24

hidden units (one hidden layer is enough). In this case, two MLP having 25 hidden units have been used. The first MLP, ANN_1, was trained on the Food dataset and had a $9 \times 25 \times 9$ structure, with 9 inputs, 25 hidden and 9 outputs. The second one, ANN_2, was trained on the Zoo dataset and therefore the relative architecture was $16 \times 25 \times 16$.

C. Step 3: The weights and the other features extraction

After the training phase, the weights matrices of the two or more auto-encoders and the values of the hidden units are saved, according to the recall strategies seen in Sec. II.

D. Step 4: The point of view of each dataset on the others

As shown by Eqs. (16) and (17), the values of the activations of the hidden layer of a dataset are used as input for the hidden layer of another. The output is then generated using the second weights matrix (decoding matrix) of the latter. In this way, the first dataset is rewritten through the variables of the other one. Then, the second dataset is decoded using the second matrix of the auto-encoder trained on the first. If there are more than two datasets to be merged, this procedure must be repeated for each pair (see Sec. II A 2).

TABLE X. Gang dataset.

Name	Gang	Age	Education	Status	Occupation
ART	Jets	40	Junior school	Single	Pusher
AL	Jets	30	Junior school	Married	Burglar
SAM	Jets	20	College	Single	Bookie
CLYDE	Jets	40	Junior school	Single	Bookie
MIKE	Jets	30	Junior school	Single	Bookie
JIM	Jets	20	Junior school	Divorced	Burglar
GREG	Jets	20	High school	Married	Pusher
JOHN	Jets	20	Junior school	Married	Burglar
DOUG	Jets	30	High school	Single	Bookie
LANCE	Jets	20	Junior school	Married	Burglar
GEORGE	Jets	20	Junior school	Divorced	Burglar
PETE	Jets	20	High school	Single	Bookie
FRED	Jets	20	High school	Single	Pusher
GENE	Jets	20	College	Single	Pusher
RALPH	Jets	30	Junior school	Single	Pusher
PHIL	Sharks	30	College	Married	Pusher
IKE	Sharks	30	Junior school	Single	Bookie
NICK	Sharks	30	High school	Single	Pusher
DON	Sharks	30	College	Married	Burglar
NED	Sharks	30	College	Married	Bookie
KARL	Sharks	40	High school	Married	Bookie
KEN	Sharks	20	High school	Single	Burglar
EARL	Sharks	40	High school	Married	Burglar
RICK	Sharks	30	High school	Divorced	Burglar
OL	Sharks	30	College	Married	Pusher
NEAL	Sharks	30	High school	Single	Bookie
DAVE	Sharks	30	High school	Divorced	Pusher

In our example, we first coded the Food dataset by going from 9 to 25 variables, then we used the second weights matrix of ANN_2 to decode the 25 variables into the 16 of the Zoo dataset. Similarly, we have first coded the Zoo dataset, going from 16 to 25 variables, then, using the decoding matrix of ANN_1 we have translated Zoo into the 9 variables of the Food dataset. Figure 15 shows how the European countries would look like if they were animals, whereas

TABLE XI. Food dataset.

	Cereals	Rice	Potatoes	Sugar	Vegetables	Meat	Milk	Butter	Eggs
Belgium	72.20	4.20	98.80	40.40	103.20	102.00	80.00	7.70	14.20
Denmark	70.50	2.20	57.00	39.50	50.00	105.80	145.20	4.10	14.30
Germany	71.30	2.30	74.10	37.10	83.10	97.20	90.70	6.90	14.80
Greece	109.80	5.40	90.00	30.00	229.50	77.10	63.10	0.90	11.30
Spain	71.40	5.80	107.80	26.80	191.70	102.10	98.40	0.60	15.30
France	73.00	4.30	78.20	34.10	95.00	110.50	98.90	8.90	15.00
Ireland	93.40	3.20	151.50	34.80	55.00	105.00	185.90	3.40	11.40
Italy	110.20	4.80	38.60	27.90	181.90	88.00	65.00	2.40	11.10
Netherland	54.60	5.00	86.70	39.70	99.00	89.40	136.20	5.40	10.70
Portugal	86.00	5.70	106.60	29.40	100.00	75.50	96.00	1.50	7.70
Gr. Britain	74.30	4.50	94.10	39.80	60.00	74.40	129.30	3.20	10.80
Austria	68.70	4.20	62.60	37.10	81.90	93.40	121.30	4.30	13.40
Finland	70.10	5.40	61.60	35.70	52.60	65.00	208.40	5.80	10.90
Island	79.70	1.90	50.20	54.90	50.00	71.70	205.60	4.60	11.30
Norway	76.90	3.50	73.20	37.30	48.30	54.90	176.50	2.10	11.30
Sweden	69.30	4.30	70.00	37.50	49.50	60.50	154.10	5.70	12.90

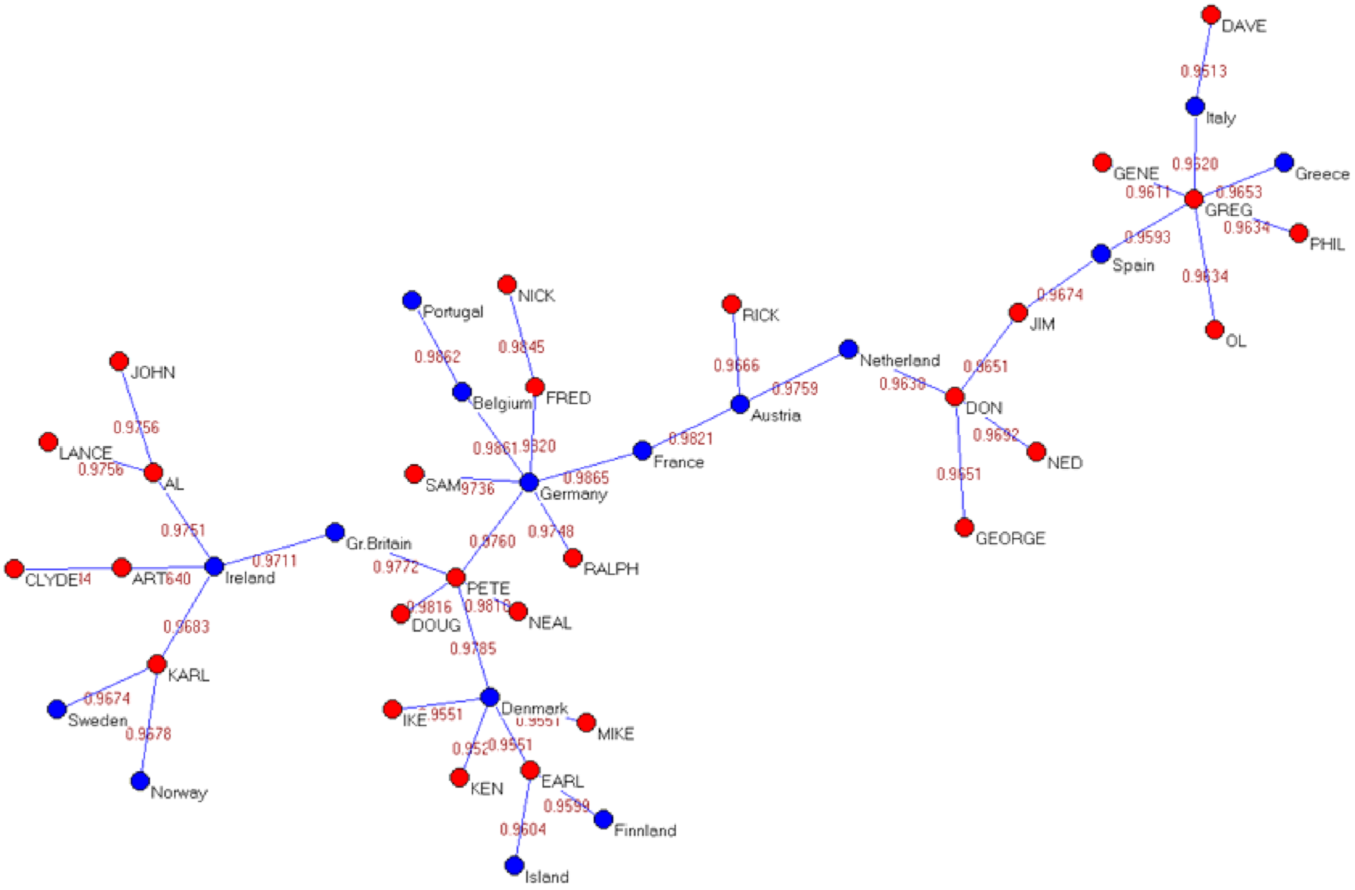


FIG. 14. Similarity graph based on Auto CM NN.

TABLE XII. Food from the point of view of gang.

	20's	30's	40's	JH	College	HS	Single	Married	Divorced	Pusher	Bookie	Burglar
Belgium	0.81	0.06	0.00	0.03	0.46	0.08	0.04	0.98	00.00	0.33	0.00	0.52
Denmark	0.67	0.01	0.03	0.24	0.00	0.41	0.98	0.01	0.01	0.3	0.1	0.08
Germany	0.94	0.01	0.00	0.02	0.27	0.20	0.62	0.47	0.00	0.05	0.20	0.14
Greece	0.17	0.87	0.00	0.00	0.00	0.99	0.00	0.82	0.70	0.59	0.05	0.08
Spain	0.91	0.28	0.00	0.36	0.00	0.76	0.00	0.55	0.90	0.05	0.00	1.00
France	0.92	0.03	0.00	0.01	0.26	0.35	0.05	0.98	0.00	0.09	0.00	0.81
Ireland	0.19	0.01	0.62	0.89	0.00	0.64	0.48	0.75	0.00	0.5	0.00	0.27
Italy	0.07	0.9	0.01	0.00	0.00	1.00	0.08	0.29	0.16	0.83	0.12	0.01
Netherland	0.03	0.85	0.01	0.62	0.16	0.01	0.01	0.97	0.02	0.20	0.00	0.94
Portugal	0.00	1.00	0.04	0.17	0.00	0.61	0.00	0.98	0.04	0.56	0.00	0.27
Gr. Britain	0.01	0.79	0.13	0.31	0.02	0.19	0.05	0.88	0.01	0.08	0.12	0.21
Austria	0.16	0.40	0.01	0.08	0.03	0.32	0.14	0.53	0.02	0.13	0.02	0.50
Finland	0.00	0.92	0.55	0.09	0.00	0.75	0.03	0.92	0.00	0.00	0.29	0.79
Island	0.00	0.07	0.98	0.30	0.01	0.32	0.99	0.05	0.00	0.03	0.97	0.01
Nanay	0.00	0.84	0.57	0.37	0.00	0.60	0.24	0.35	0.01	0.00	0.96	0.10
Sweden	0.01	0.70	0.11	0.05	0.07	0.31	0.02	0.95	0.00	0.00	0.82	0.54

Fig. 16 shows a little sample of the Zoo dataset as explained by the 9 variables of Food.

Obviously, once the variables of another dataset are enhanced by the previously shown procedure, it is possible to classify the new records, i.e., to determine the unknown relevant targets. For example, it will be possible to associate each country with its target: what animal would each country be? Many algorithms can be chosen for this task. In this case, two different approaches have been used: a well known machine learning algorithm, the k Nearest Neighbour (kNN), and a powerful artificial neural network, SineNet.[15] Tables XVI and XVII show the results of classification. The two Machine Learning Systems (MLS) basically seem to be in agreement, although the different nature of their functioning leads to different outputs. kNN produces a hard-edged result, while the one of SineNet is fuzzy and easier to interpret. So,

TABLE XIII. Gang from the point of view of food.

	Cereals	Rice	Potatoes	Sugar	Vegetables	Meat	Milk	Butter	Eggs
ART	61.92	1.98	119.15	53.85	48.65	106.95	197.43	2.25	8.99
AL	55.00	5.68	126.37	29.38	62.78	87.07	151.97	3.02	11.48
SAM	58.14	1.92	41.42	47.93	63.83	89.40	67.27	7.83	15.25
CLVDE	57.94	1.92	77.62	53.44	48.32	75.35	206.60	3.28	13.25
MIKE	54.77	2.14	54.42	41.32	48.46	58.27	143.09	1.04	11.82
JIM	54.95	4.21	93.15	27.31	186.34	102.86	116.81	0.74	15.28
GREG	109.64	3.89	117.29	32.86	197.53	107.65	66.95	6.17	12.98
JOHN	57.29	4.59	148.72	33.84	64.46	107.75	190.12	6.46	15.02
DOUG	83.03	2.88	40.54	27.76	52.73	65.47	90.69	0.94	13.28
LANCE	57.29	4.55	148.72	33.84	64.46	107.75	190.12	6.46	15.02
GEORGE	54.95	4.21	93.15	27.31	186.34	102.86	116.81	0.74	15.28
PETE	102.89	1.94	55.48	30.77	53.40	100.25	115.82	5.00	15.27
FRED	106.90	2.15	75.25	31.42	83.60	110.08	81.47	3.39	14.35
GENE	61.54	2.00	49.47	46.99	184.14	108.78	63.61	5.98	14.83
RALPH	55.05	2.82	60.29	38.25	57.80	92.46	72.15	0.75	8.79
PHIL	83.45	4.94	81.93	50.89	205.12	73.64	64.08	6.19	8.18
IKE	54.77	2.14	54.42	41.32	48.46	58.27	143.09	1.04	11.82
NICK	101.42	3.80	41.24	30.17	96.52	105.66	67.80	2.24	9.13
DON	55.88	5.68	55.52	38.57	129.35	62.27	98.52	8.18	12.54
NED	59.47	4.40	52.19	44.21	107.02	55.17	65.45	4.19	10.62
KARL	109.76	3.18	99.34	44.49	53.96	64.36	166.47	7.88	12.06
KEN	76.71	2.39	46.26	28.85	49.66	109.67	205.31	7.83	15.28
EARL	101.07	5.27	74.47	41.38	50.61	94.71	207.42	8.53	12.27
RICK	86.83	5.72	40.23	26.92	135.07	85.33	181.07	0.85	13.81
OL	84.45	4.94	81.93	50.89	205.12	73.64	64.05	6.19	8.18
NEAI.	83.03	2.88	40.54	27.76	52.73	65.47	90.69	0.94	13.28
DAVE	107.93	5.58	43.86	27.08	211.61	84.68	68.21	0.63	8.81

TABLE XIV. Dataset food with targets.

	Cereals	Rice	Potatoes	Sugar	Vegetables	Meat	Milk	Butter	Eggs	Center	North	South
Belgium	72.2	4.2	98.8	40.4	103.2	102	80	7.7	14.2	1	0	0
Denmark	70.5	2.2	57	39.5	50	105.8	145.2	4.1	14.3	0	1	0
Germany	71.3	2.3	74.1	37.1	83.1	97.2	90.7	6.9	14.8	1	0	0
Greece	109.8	5.4	90	30	229.5	77.1	63.1	0.9	11.3	0	0	1
Spain	71.4	5.8	107.8	26.8	191.7	102.1	98.4	0.6	15.3	0	0	1
France	73	4.3	78.2	34.1	95	110.5	98.9	8.9	15	1	0	0
Ireland	93.4	3.2	151.5	34.8	55	105	185.9	3.4	11.4	1	0	0
Italy	110.2	4.8	38.6	27.9	181.9	88	65	2.4	11.1	0	0	1
Netherland	54.6	5	86.7	39.7	99	89.4	136.2	5.4	10.7	1	0	0
Portugal	86	5.7	106.6	29.4	100	75.5	96	1.5	7.7	0	0	1
Gr. Britain	74.3	4.5	94.1	39.8	60	74.4	129.3	3.2	10.8	1	0	0
Austria	68.7	4.2	62.6	37.1	81.9	93.4	121.3	4.3	13.4	1	0	0
Finland	70.1	5.4	61.6	35.7	52.6	65	208.4	5.8	10.9	0	1	0
Island	79.7	1.9	50.2	54.9	50	71.7	205.6	4.6	11.3	0	1	0
Norway	76.9	3.5	73.2	37.3	48.3	54.9	176.5	2.1	11.3	0	1	0
Sweden	69.3	4.3	70	37.5	49.5	60.5	154.1	5.7	12.9	0	1	0

although for both MLS no one is a bird, SineNet suggests who is that, in a fuzzy way, approaches it the most. It seems therefore that, if Ireland were an animal, it would be a mammal, whereas Finland would be an amphibian. It should be stressed that this type of transformation does not have a semantic value, we just can say, that relations are maintained. To prove this last statement, we use a multidimensional scaling technique (MDS). The goal is to qualitatively evaluate the differences in the results of the same MDS algorithm applied to the source dataset and the dataset rewritten with the attributes of the other(s). It is surprisingly noted that even with respect to the new variables, records are clustered in the same way. Thus, whatever the transfer from one world to another, the geometry of the original dataset is preserved in the new world. We show this feature in a case where clustering is natural and the number of records is acceptable: the Food dataset. Figure 17(a) shows how the original dataset records (9 dimensions) projected into a space

TABLE XV. Zoo dataset.

	Hair	Feathers	Eggs	Milk	Airborne	Aquatic	Predator	Toothed	Backbone	Breathes	Venomous	Fins	Legs	Tail	Domestic	Catsize	Class	Mammal	Bird	Reptile	Fish	Amphibian	Bug	Invertebrate
Aardvark	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1	1	0	0	0	0	0	0
Antelope	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1	1	0	0	0	0	0	0
Bass	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4	0	0	0	1	0	0	0
Bear	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	1	1	0	0	0	0	0	0
Boar	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	1	1	0	0	0	0	0	0
Buffalo	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	1	1	0	0	0	0	0	0
Calf	1	0	0	1	0	0	0	1	1	1	0	0	4	1	1	1	1	1	0	0	0	0	0	0
Carp	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	4	0	0	0	1	0	0	0
Catfish	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	4	0	0	0	1	0	0	0

of just two dimensions look like. Figure 17(b) shows the projection on a two-dimensional space of the Food dataset, seen from the Zoo dataset point of view, that is through 16 dimensions. Looking at Fig. 17(a), which is based on real data, it is noted that Denmark, although belonging to the North group, is located close to the Central nations. In the case of the transferred data, however, Denmark is well placed and all clusters are well identified.

TIW does not preserve the distances between hyper-points in the passage from a world to another, but maintains the relationships between them. For this reason, it could be defined as a "meta-isometry". As a quantitative measure of what has been said, it is possible to calculate the correlation matrix between the matrix of the distances of records, computed before and after the transfer. Let us denote as $\mathcal{S}$ the source dataset, and as $\mathcal{T}$ the same dataset after the transfer. Let $N_{\mathcal{S}}$ and $N_{\mathcal{T}}$ be the numbers of variables of the source and transferred worlds, and M the number of their records (hyper-points). Then, $\mathcal{D}^{[\mathcal{S}]} = \left\{d_{ij}^{[\mathcal{S}]}\right\}_{ij}$ is the $M \times M$ squared matrix of distances among the source world records and $\mathcal{D}^{[\mathcal{T}]} = \left\{d_{ij}^{[\mathcal{T}]}\right\}_{ij}$ is the analogous $M \times M$ matrix calculated from the transferred world data. The matrix $\rho = \mathcal{R}(\mathcal{D}^{[\mathcal{S}]}, \mathcal{D}^{[\mathcal{T}]})$ represents the extent to which the distance between the i-th record of $\mathcal{S}$ and all the others is correlated to the distance between the j-th record of $\mathcal{T}$ and all the others. The main diagonal of that matrix, i.e., $\rho_{ii} = \mathcal{R}(\mathcal{D}^{[\mathcal{S}]}, \mathcal{D}^{[\mathcal{T}]})_{ii}$, provides a measure of the relationship between the distances between the original points and the transferred ones.

In the case where $\mathcal{S} = \text{Food}$ and $\mathcal{T} = \text{Zoo}$, ρ is 16×16 as shown in Table XVIII. It is easy to see that the diagonal has extremely high correlation values. The experiments that have been carried out seem to suggest the following conjecture:

Conjecture 1.

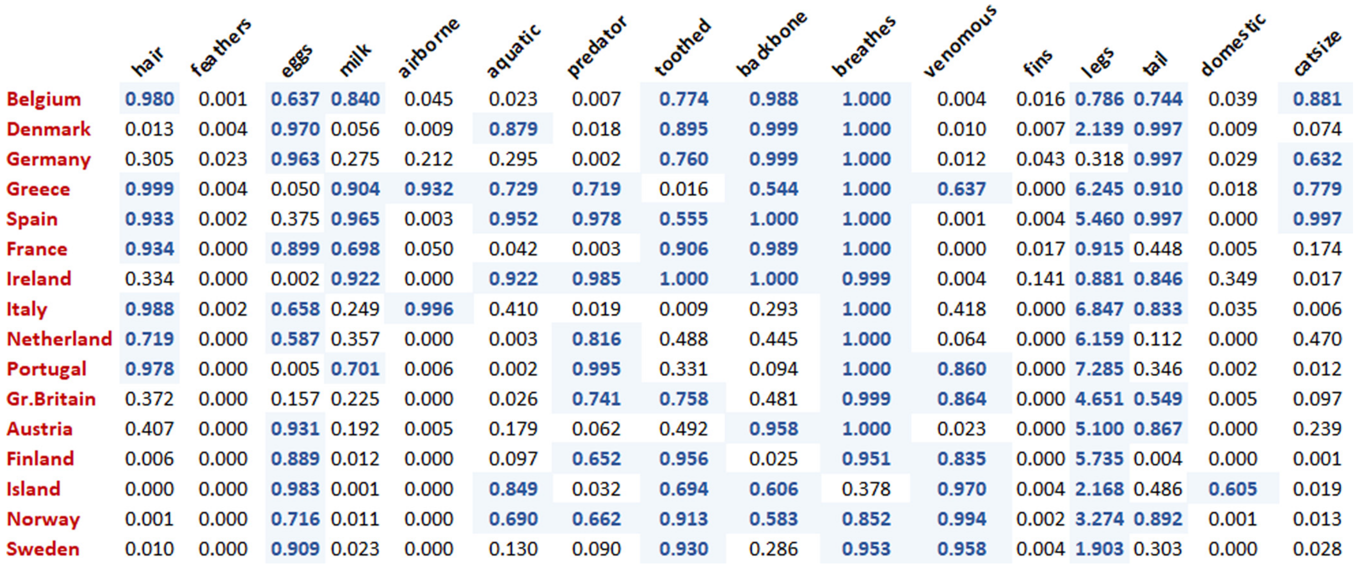
$$\max_j \mathcal{R}\left(\mathcal{D}^{[\mathcal{S}]}, \mathcal{D}^{[\mathcal{T}]}\right)_{ij} = \max_j \mathcal{R}\left(d_{ij}^{[\mathcal{S}]}, d_{ij}^{[\mathcal{T}]}\right) =, \tag{24}$$

$$= \mathcal{R}\left(d_{ii}^{[\mathcal{S}]}, d_{ii}^{[\mathcal{T}]}\right) =, \tag{25}$$

$$= \rho_{ii}. \tag{26}$$

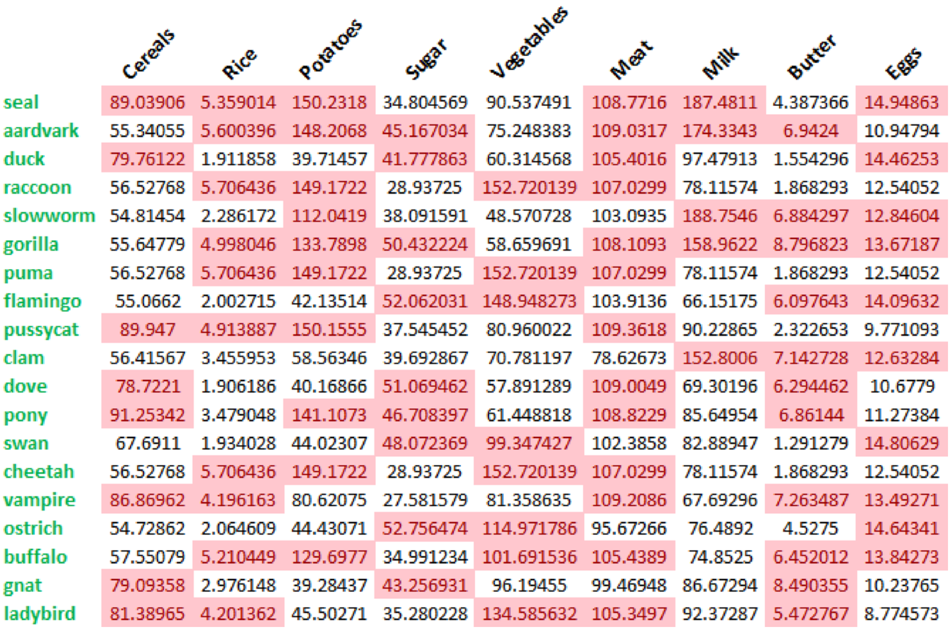
E. Step 5: The records fusion

We concatenate the hidden units of each dataset in one new dataset. This is possible, since all the datasets included into the experimentation were trained with the same number of Hidden units, as explained in Sec. II A 2. Thus, we may use a further auto-encoder, the Principal Component Analysis or an analogous algorithm to calculate the distance and/or the strength of association among the records of all the datasets, using the hidden units of each one as coordinates in a H-dimensional space (H = Number of Hidden units used during the previous training phase). At the end of this procedure, we have a square matrix of the similarity/distance between each pair of records, whether they are records from the same dataset or not.



	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize
Belgium	0.980	0.001	0.637	0.840	0.045	0.023	0.007	0.774	0.988	1.000	0.004	0.016	0.786	0.744	0.039	0.881
Denmark	0.013	0.004	0.970	0.056	0.009	0.879	0.018	0.895	0.999	1.000	0.010	0.007	2.139	0.997	0.009	0.074
Germany	0.305	0.023	0.963	0.275	0.212	0.295	0.002	0.760	0.999	1.000	0.012	0.043	0.318	0.997	0.029	0.632
Greece	0.999	0.004	0.050	0.904	0.932	0.729	0.719	0.016	0.544	1.000	0.637	0.000	6.245	0.910	0.018	0.779
Spain	0.933	0.002	0.375	0.965	0.003	0.952	0.978	0.555	1.000	1.000	0.001	0.004	5.460	0.997	0.000	0.997
France	0.934	0.000	0.899	0.698	0.050	0.042	0.003	0.906	0.989	1.000	0.000	0.017	0.915	0.448	0.005	0.174
Ireland	0.334	0.000	0.002	0.922	0.000	0.922	0.985	1.000	1.000	0.999	0.004	0.141	0.881	0.846	0.349	0.017
Italy	0.988	0.002	0.658	0.249	0.996	0.410	0.019	0.009	0.293	1.000	0.418	0.000	6.847	0.833	0.035	0.006
Netherland	0.719	0.000	0.587	0.357	0.000	0.003	0.816	0.488	0.445	1.000	0.064	0.000	6.159	0.112	0.000	0.470
Portugal	0.978	0.000	0.005	0.701	0.006	0.002	0.995	0.331	0.094	1.000	0.860	0.000	7.285	0.346	0.002	0.012
Gr.Britain	0.372	0.000	0.157	0.225	0.000	0.026	0.741	0.758	0.481	0.999	0.864	0.000	4.651	0.549	0.005	0.097
Austria	0.407	0.000	0.931	0.192	0.005	0.179	0.062	0.492	0.958	1.000	0.023	0.000	5.100	0.867	0.000	0.239
Finland	0.006	0.000	0.889	0.012	0.000	0.097	0.652	0.956	0.025	0.951	0.835	0.000	5.735	0.004	0.000	0.001
Island	0.000	0.000	0.983	0.001	0.000	0.849	0.032	0.694	0.606	0.378	0.970	0.004	2.168	0.486	0.605	0.019
Norway	0.001	0.000	0.716	0.011	0.000	0.690	0.662	0.913	0.583	0.852	0.994	0.002	3.274	0.892	0.001	0.013
Sweden	0.010	0.000	0.909	0.023	0.000	0.130	0.090	0.930	0.286	0.953	0.958	0.004	1.903	0.303	0.000	0.028

FIG. 15. How Zoo sees Food. The values higher than 0.5 have been highlighted.



	Cereals	Rice	Potatoes	Sugar	Vegetables	Meat	Milk	Butter	Eggs
seal	89.03906	5.359014	150.2318	34.804569	90.537491	108.7716	187.4811	4.387366	14.94863
aardvark	55.34055	5.600396	148.2068	45.167034	75.248383	109.0317	174.3343	6.9424	10.94794
duck	79.76122	1.911858	39.71457	41.777863	60.314568	105.4016	97.47913	1.554296	14.46253
raccoon	56.52768	5.706436	149.1722	28.93725	152.720139	107.0299	78.11574	1.868293	12.54052
slowworm	54.81454	2.286172	112.0419	38.091591	48.570728	103.0935	188.7546	6.884297	12.84604
gorilla	55.64779	4.998046	133.7898	50.432224	58.659691	108.1093	158.9622	8.796823	13.67187
puma	56.52768	5.706436	149.1722	28.93725	152.720139	107.0299	78.11574	1.868293	12.54052
flamingo	55.0662	2.002715	42.13514	52.062031	148.948273	103.9136	66.15175	6.097643	14.09632
pussycat	89.947	4.913887	150.1555	37.545452	80.960022	109.3618	90.22865	2.322653	9.771093
clam	56.41567	3.455953	58.56346	39.692867	70.781197	78.62673	152.8006	7.142728	12.63284
dove	78.7221	1.906186	40.16866	51.069462	57.891289	109.0049	69.30196	6.294462	10.6779
pony	91.25342	3.479048	141.1073	46.708397	61.448818	108.8229	85.64954	6.86144	11.27384
swan	67.6911	1.934028	44.02307	48.072369	99.347427	102.3858	82.88947	1.291279	14.80629
cheetah	56.52768	5.706436	149.1722	28.93725	152.720139	107.0299	78.11574	1.868293	12.54052
vampire	86.86962	4.196163	80.62075	27.581579	81.358635	109.2086	67.69296	7.263487	13.49271
ostrich	54.72862	2.064609	44.43071	52.756474	114.971786	95.67266	76.4892	4.5275	14.64341
buffalo	57.55079	5.210449	129.6977	34.991234	101.691536	105.4389	74.8525	6.452012	13.84273
gnat	79.09358	2.976148	39.28437	43.256931	96.19455	99.46948	86.67294	8.490355	10.23765
ladybird	81.38965	4.201362	45.50271	35.280228	134.585632	105.3497	92.37287	5.472767	8.774573

FIG. 16. How Food sees Zoo (just a little sample is shown). The values higher than the mean for each variable have been highlighted.

In our experiment, we have a dataset made up of $16 + 99 = 115$ records and 25 virtual variables.

F. Step 6: The graph fusion

We can apply a graph filter (e.g., Minimum Spanning Tree,[18] Maximally Regular Graph (MRG),[11,12] or similar) to the new square matrix of similarities to generate a weighted tree or graph among all the records simultaneously, as shown in Sec. III B. Again, the matrix of similarities was generated using the AutoCM NN.[15] Figure 18 shows the MRG related to the fusion of the Zoo and Food datasets.

1. The forced graph fusion

Since, obviously, records belonging to the same dataset tend to be more clustered, an additional graphic filter has been elaborated to be applied to the similarity/distance matrix. To force each element of a dataset to connect to an element of the other, the matrix of similarity/distances has been modified by setting at 0 or $+\infty$, respectively, the value of similarity/distance of records belonging to the same dataset as shown in Fig. 19. The forced similarity graph of the FD (see Definition 8) pertaining to the Zoo and Food datasets is shown in Figs. 20 and 21.

G. Step 7: The quantitative validation of the trans world data transferring

In the case of datasets including target columns, a further quantitative validation of the transfer from one world to another can be carried out (see Sec. II B). We can measure the effectiveness of the trans world operation with a K-Fold Cross Validation protocol comparing the accuracy obtained in the three cases: source dataset (default measure), slight

TABLE XVI. Which animal represents each country according to the well-known k Nearest Neighbour (kNN) algorithm.

kNN	Mammal	Bird	Rept.	Fish	Amphibian	Bug	Inverteb.
Belgium	1	0	0	0	0	0	0
Denmark	0	0	0	0	1	0	0
Germany	0	0	1	0	0	0	0
Greece	1	0	0	0	0	0	0
Spain	1	0	0	0	0	0	0
France	1	0	0	0	0	0	0
Ireland	1	0	0	0	0	0	0
Italy	0	0	0	0	0	1	0
Netherland	1	0	0	0	0	0	0
Portugal	0	0	0	0	0	0	1
Gr. Britain	0	0	0	0	0	0	1
Austria	0	0	1	0	0	0	0
Finland	0	0	0	0	0	1	0
Island	0	0	0	0	1	0	0
Norway	0	0	1	0	0	0	0
Sweden	0	0	1	0	0	0	0

TABLE XVII. Which animal represents each country according to SineNet.

Sine net	Mammal	Bird	Rept.	Fish	Amphibian	Bug	Inverteb.
Belgium	0.97	0.01	0.01	0.02	0.02	0.01	0.01
Denmark	0.08	0.03	0.08	0.06	0.82	0.05	0.01
Germany	0.49	0.01	0.13	0.13	0.10	0.03	0.01
Greece	0.74	0.02	0.16	0.01	0.01	0.14	0.03
Spain	0.96	0.01	0.02	0.04	0.01	0.01	0.00
France	0.93	0.00	0.02	0.08	0.04	0.02	0.01
Ireland	0.84	0.01	0.02	0.05	0.06	0.02	0.01
Italy	0.28	0.04	0.12	0.01	0.03	0.79	0.21
Netherland	0.84	0.00	0.09	0.01	0.02	0.08	0.07
Portugal	0.52	0.01	0.11	0.05	0.07	0.19	0.09
Gr. Britain	0.52	0.01	0.48	0.01	0.09	0.17	0.25
Austria	0.50	0.01	0.30	0.03	0.20	0.14	0.02
Finland	0.06	0.01	0.11	0.01	0.13	0.40	0.84
Island	0.03	0.07	0.31	0.14	0.51	0.15	0.06
Norway	0.07	0.04	0.46	0.06	0.63	0.13	0.06
Sweden	0.10	0.02	0.38	0.03	0.32	0.18	0.40

dataset (Definition 6) and combined dataset (Definition 7). Several Machine Learning Systems (MLS) were used to perform this type of verification, to show that the results are not dependent on the choice of method. Tables XIX and XX show the results. As already highlighted in Sec. III A 1, the prediction carried out by considering only the slight dataset provides an improvement in some cases only, whereas the classification built upon the combined dataset never yields a worse performance.

V. CONCLUSION

In this paper, an innovative theory for the fusion of different datasets is proposed. Our theory focuses upon the possibility to consider how very different datasets may be put in relation to one another in order to improve our overall knowledge about both.

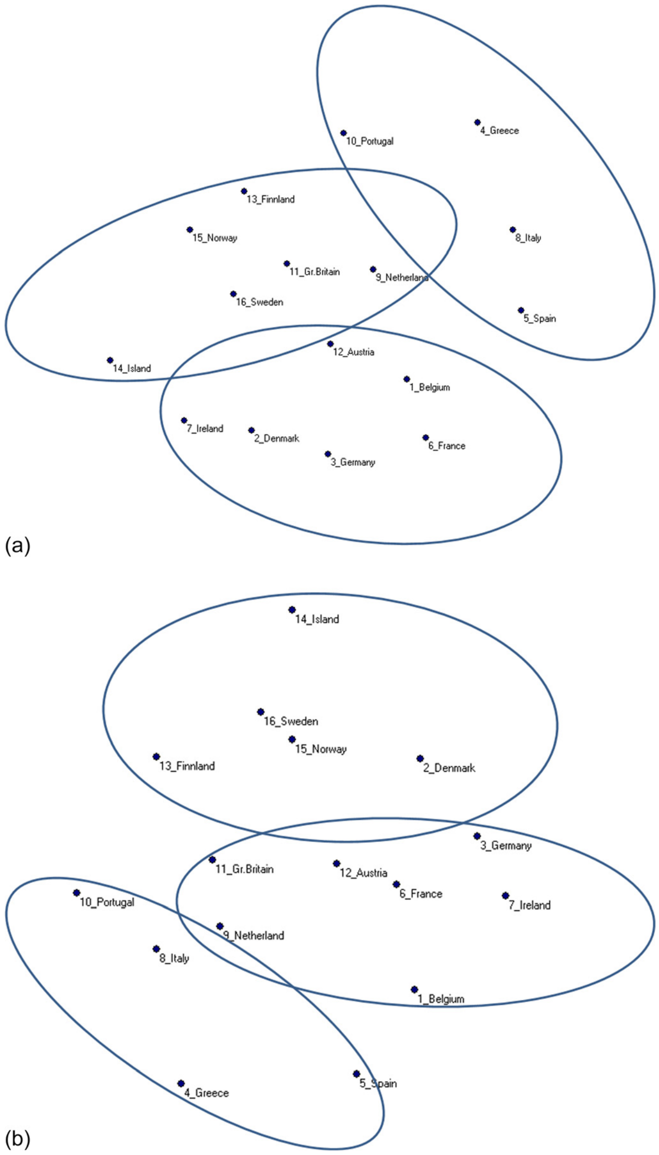


FIG. 17. Multi Dimensional Scaling technique performed on real and transformed data. (a) MDS applied to Food source dataset. (b) MDS applied to how Zoo sees Food.

The experimentations reported in Secs. III A 1 and III A 2 show that, in some cases, TIW-based predictions can improve upon those based upon original data and that, in any case, the fused data from S + TIW are conducive to performance gains in classification tasks.

The experimentation in Sec. III A 3 shows that considering the fused dataset can be useful in terms of ease of convergence. This aspect is highlighted in Sec. III A 4, where a quicker convergence occurred when S + TIW was taken into account. This experimentation also shows the composite learning strategy of a fused dataset. Specifically, we see how part of the input specializes in coding the affirmative “it is” property, whereas another in coding the negative “it is not” one.

The experimentation III B paves the way to a possible use of TIW for unsupervised tasks, as the knowledge contained in the source data seems to be preserved after the fusion.

TABLE XVIII. The correlation matrix in the case where $\mathcal{S}$ = Food and $\mathcal{T}$ = Zoo.

	Belgium	Denmark	Germany	Greece	Spain	France	Ireland	Italy	Netherland	Portugal	Gr. Britain	Austria	Finland	Island	Norway	Sweden
Zoo (Belgium)	**0.89**	0.31	0.57	−0.06	0.41	0.81	−0.02	−0.04	0.34	−0.18	0.02	0.51	−0.15	−0.29	−0.28	0.05
Zoo (Denmark)	0.43	**0.88**	0.64	−0.43	−0.04	0.49	0	−0.24	0.27	−0.42	0.21	0.69	0.19	0.29	0.3	0.42
Zoo (Germany)	0.64	0.64	**0.9**	−0.25	0.1	0.66	−0.03	−0.12	0.15	−0.43	−0.04	0.54	−0.11	0.01	−0.09	0.16
Zoo (Greece)	−0.23	−0.57	−0.43	**0.96**	0.55	−0.25	−0.34	0.74	−0.28	0.44	−0.26	−0.33	−0.46	−0.72	−0.41	−0.52
Zoo (Spain)	0.18	−0.18	−0.06	0.41	**0.95**	0.15	−0.06	0.19	−0.02	0.13	−0.19	0.02	−0.42	−0.66	−0.41	−0.38
Zoo (France)	0.81	0.45	0.61	−0.27	0.18	**0.91**	−0.03	−0.14	0.46	−0.21	0.16	0.64	0.11	−0.11	−0.07	0.29
Zoo (Ireland)	0.08	0.05	0	−0.19	0.08	0.05	**0.94**	−0.34	−0.08	−0.07	−0.02	−0.11	−0.21	−0.14	−0.1	−0.19
Zoo (Italy)	−0.19	−0.31	−0.26	0.66	0.2	−0.13	−0.51	**0.94**	−0.13	0.34	−0.11	−0.06	−0.16	−0.44	−0.17	−0.2
Zoo (Netherland)	0.24	−0.06	−0.02	0.1	0.25	0.2	−0.15	0.1	**0.78**	0.48	0.53	0.4	0.44	−0.2	0.19	0.35
Zoo (Portugal)	−0.2	−0.46	−0.44	0.42	0.2	−0.24	−0.11	0.32	0.31	**0.92**	0.43	−0.07	0.25	−0.37	0.16	0.07
Zoo (Gr. Britain)	0.05	0	−0.09	0.03	0.04	0	0	0.03	0.61	0.57	**0.86**	0.37	0.63	0.03	0.61	0.58
Zoo (Austria)	0.55	0.57	0.52	−0.12	0.22	0.58	−0.16	0.08	0.54	−0.06	0.37	**0.88**	0.27	0.02	0.21	0.43
Zoo (Finland)	−0.05	0.08	−0.07	−0.25	−0.26	−0.03	−0.15	−0.15	0.58	0.27	0.67	0.34	**0.94**	0.28	0.68	0.77
Zoo (Island)	−0.23	0.23	−0.04	−0.57	−0.66	−0.23	−0.2	−0.47	−0.03	−0.44	0.1	−0.01	0.26	**0.91**	0.41	0.3
Zoo (Norway)	−0.05	0.34	0.07	−0.3	−0.24	−0.05	−0.05	−0.22	0.36	0.06	0.64	0.4	0.66	0.43	**0.9**	0.71
Zoo (Sweden)	0.15	0.34	0.2	−0.38	−0.31	0.18	−0.14	−0.23	0.56	0.04	0.67	0.52	0.84	0.41	0.74	**0.93**

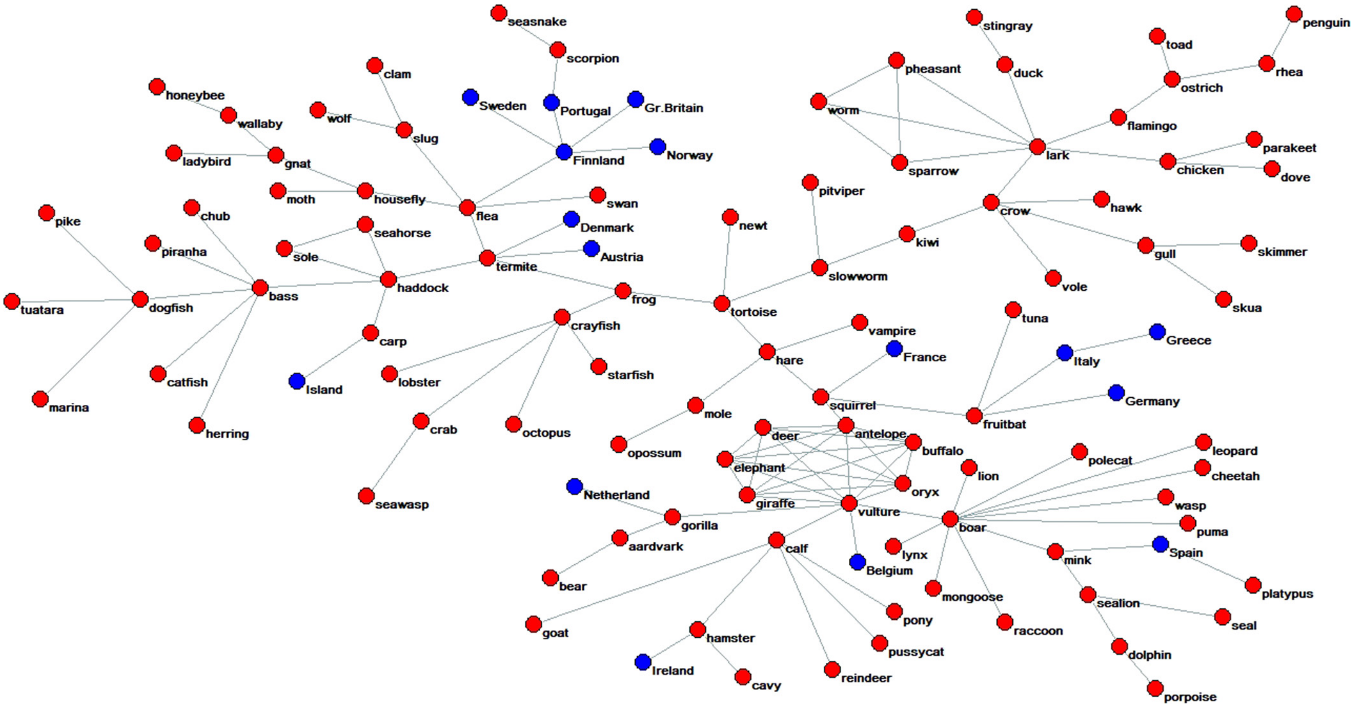
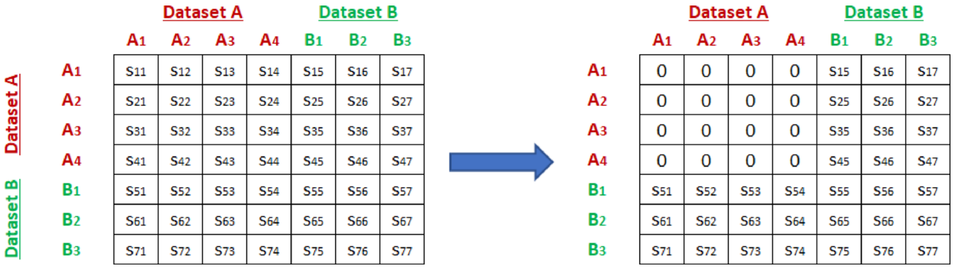


FIG. 18. The MRG related to the fused dataset of the Zoo and Food datasets, obtained through the AutoCM auto-encoder.



		Dataset A				Dataset B		
		A1	A2	A3	A4	B1	B2	B3
Dataset A	A1	S11	S12	S13	S14	S15	S16	S17
	A2	S21	S22	S23	S24	S25	S26	S27
	A3	S31	S32	S33	S34	S35	S36	S37
	A4	S41	S42	S43	S44	S45	S46	S47
Dataset B	B1	S51	S52	S53	S54	S55	S56	S57
	B2	S61	S62	S63	S64	S65	S66	S67
	B3	S71	S72	S73	S74	S75	S76	S77

	Dataset A				Dataset B		
	A1	A2	A3	A4	B1	B2	B3
A1	0	0	0	0	S15	S16	S17
A2	0	0	0	0	S25	S26	S27
A3	0	0	0	0	S35	S36	S37
A4	0	0	0	0	S45	S46	S47
B1	S51	S52	S53	S54	S55	S56	S57
B2	S61	S62	S63	S64	S65	S66	S67
B3	S71	S72	S73	S74	S75	S76	S77

FIG. 19. Transformation of a similarity matrix into a forced similarity matrix. Two toy datasets has been used to show the forcing procedure. *A* which has 4 records and *B* which has 3 records. In this case, dataset *A* is forced to connect to dataset *B*.

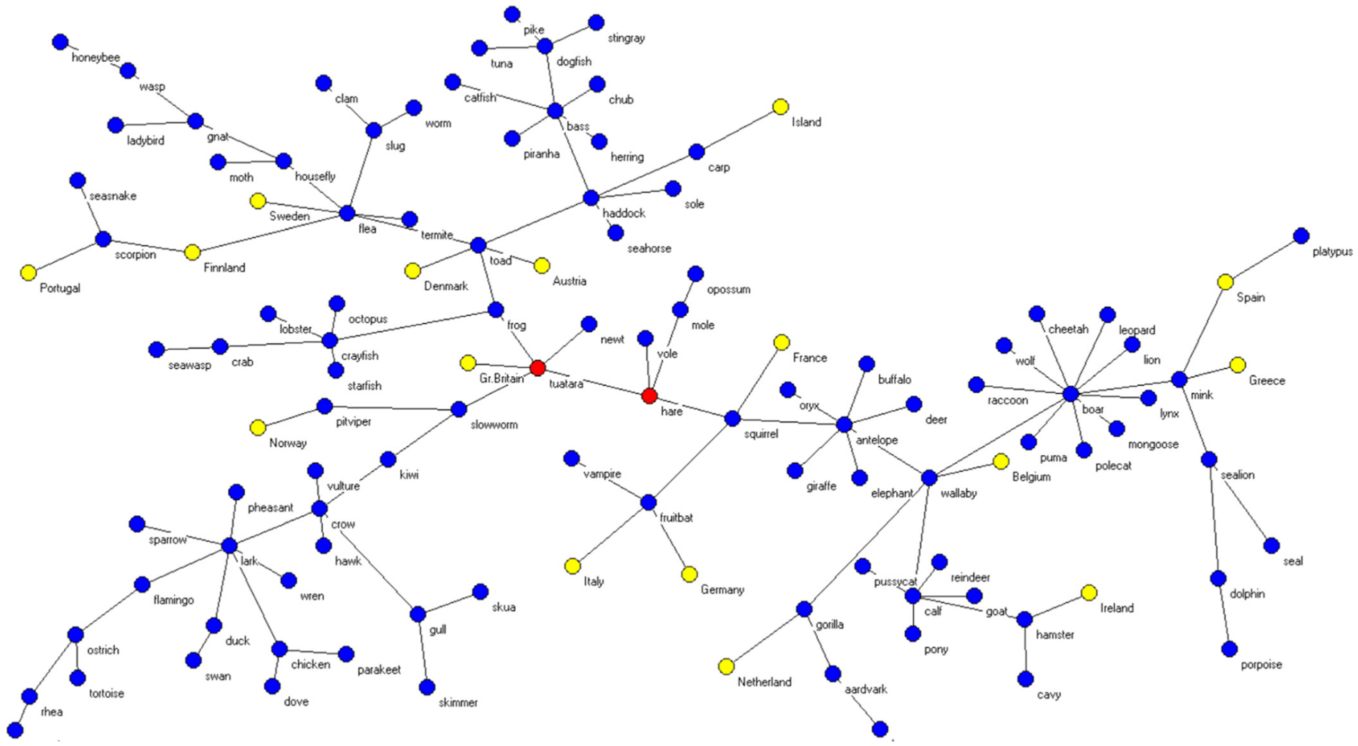


FIG. 20. Graph where countries have been forced towards animals.

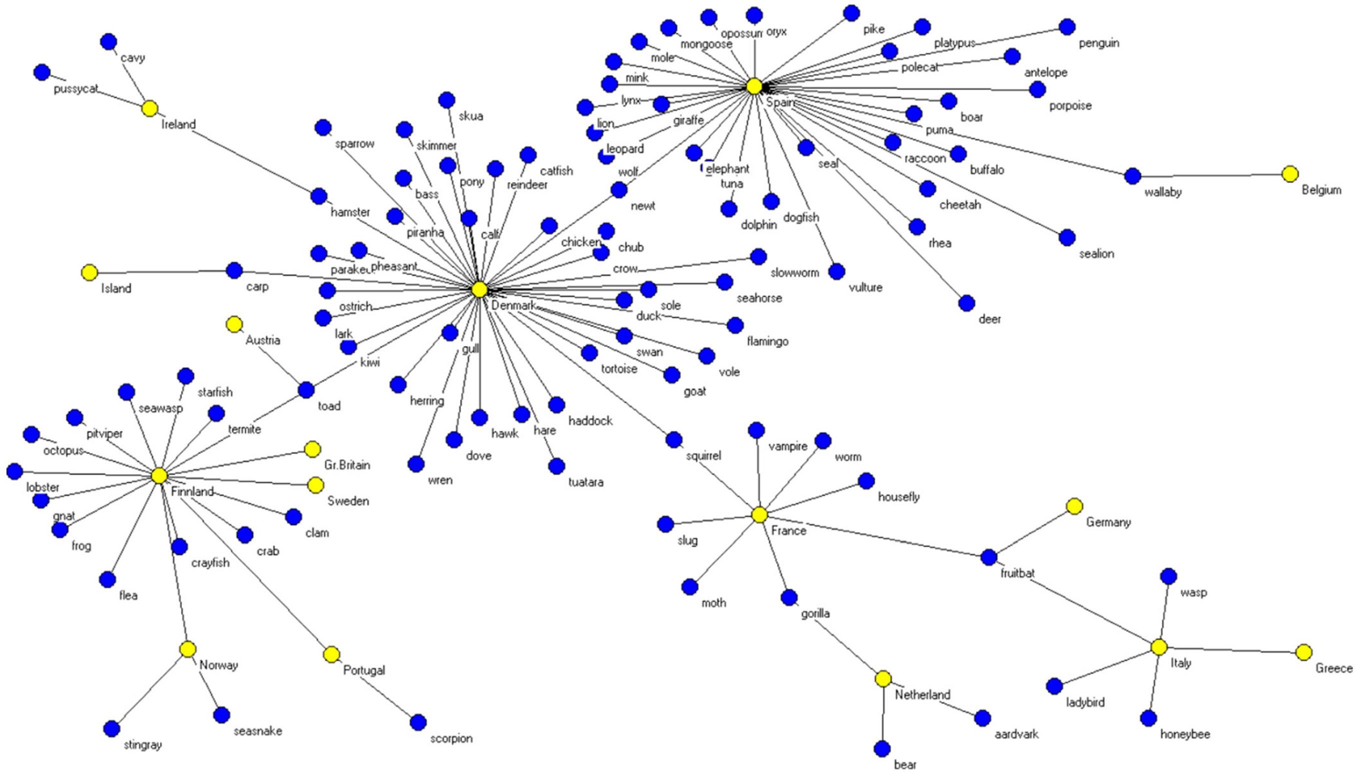


FIG. 21. Graph where animals have been forced towards countries.

TABLE XIX. Accuracy obtained in the classification of the Zoo dataset.

MLS	Dataset	Center (%)	North (%)	South (%)	Acc (%)	Err
Logistic	Source (16 × 9 × 3)	57.14	60.00	50.00	56.25	7
	Slight (16 × 16 × 3)	57.14	80.00	50.00	62.50	6
	Combined (16 × 25 × 3)	57.14	80.00	75.00	68.75	5
Radial function	Source (16 × 9 × 3)	85.71	80.00	50.00	75.00	4
	Slight (16 × 16 × 3)	85.71	100.00	25.00	75.00	4
	Combined (16 × 25 × 3)	85.71	100.00	50.00	81.25	3
Backpropagation	Source (16 × 9 × 3)	100.00	40.00	75.00	75.00	4
	Slight (16 × 16 × 3)	71.43	80.00	50.00	68.75	5
	Combined (16 × 25 × 3)	85.71	80.00	75.00	81.25	3

TABLE XX. Accuracy obtained in the classification of the food dataset, using Adaptive Vector Quantization (AVQ) and Backpropagation.

MLS/datasets	Mammal (%)	Bird (%)	Reptile (%)	Fish (%)	Amphibian (%)	Bug (%)	Invertebrate (%)	Acc (%)	Err
AVQ advanced									
Source (99 × 16 × 7)	97.50	95.00	60.00	100.00	66.67	87.50	90.00	92.93	7
Slight (99 × 9 × 7)	100.00	100.00	0.00	92.31	0.00	87.50	90.00	88.89	11
Comb (99 × 25 × 7)	97.14	100.00	33.33	100.00	100.00	83.33	100.00	95.96	4
Backpropagation									
Source (99 × 16 × 7)	100.00	100.00	87.50	100.00	50.00	70.00	69.05	91.96	8
Slight (99 × 9 × 7)	97.37	100.00	50.00	100.00	0.00	80.00	71.43	85.86	14
Comb (99 × 25 × 7)	100.00	100.00	62.50	100.00	100.00	80.00	69.05	91.96	8

Finally, the step-by-step example detailed in Sec. IV presents a thorough application of the method, to facilitate understanding and replication.

[1] See http://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval) for "Australian credit approval," (Accessed 2018 January 17).

[2] See https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data) for "German credit data," (Accessed 2018 January 17).

[3] H. Abdi and L. J. Williams, "Principal component analysis," Wiley Interdiscip. Rev. Comput. Stat. **2**(4), 433–459 (2010).

[4] F. Alexander Bais and J. Doyne Farmer, "The physics of information," preprint arXiv:0708.2837 (2007).

[5] Y. Bengio *et al.*, "Learning deep architectures for ai," Found. Trends Mach. Learn. **2**(1), 1–127 (2009).

[6] C. Benzmüller and L. C. Paulson, "Quantified multimodal logics in simple type theory," Logica Universalis **7**(1), 7–20 (2013).

[7] "The frame problem in artificial intelligence," in *Proceedings of the 7th WSEAS International Conference on Evolutionary Computing, Cavtat, Croatia, 12–14 June 2006*, edited by F. M. Brown (2014), pp. 12–19.

[8] M. Buscema, Spin Net. Spin Net is Implemented in Modular Associative ANNs Semeion Software.

[9] M. Buscema, "Recirculation neural networks," Subst. Use Misuse **33**(2), 383–388 (1998).

[10] M. Buscema, "The general philosophy of the artificial adaptive systems," in *Applications of Mathematics in Models, Artificial Neural Networks and Arts* (Springer, 2010), pp. 197–226.

[11] M. Buscema, M. Asadi-Zeydabadi, W. Lodwick, and M. Breda, "The h0 function, a new index for detecting structural/topological complexity information in undirected graphs," Physica A **447**, 355–378 (2016).

[12] M. Buscema and P. L. Sacco, "Auto-contractive maps, the h function, and the maximally regular graph (MRG): A new methodology for data mining," in *Applications of Mathematics in Models, Artificial Neural Networks and Arts* (Springer, 2010), pp. 227–275.

[13] M. Buscema and S. Terzi, "A new evolutionary approach to topographic mapping," *Proceedings of the 7th WSEAS International Conference on Evolutionary Computing, Cavtat, Croatia, June 12–14*, (2006), pp. 12–19.

[14] M. Buscema and S. Terzi, "Pst*: An evolutionary approach to the problem of multi dimensional scaling," WSEAS Trans. Inf. Sci. Appl. **3**(9), 1704–1710 (2006).

[15] P. M. Buscema, G. Massini, M. Breda, W. A. Lodwick, F. Newman, and M. Asadi-Zeydabadi, *Artificial Adaptive Systems Using Auto Contractive Maps: Theory, Applications and Extensions* (Springer, 2018), Vol. 131.

[16] P. M. Buscema and W. J. Tastle, "Artificial neural network what-if theory," Int. J. Inf. Syst. Soc. Change (IJISSC) **6**(4), 52–81 (2015).

[17] E. F. Codd, "A relational model of data for large shared data banks," Commun. ACM **13**(6), 377–387 (1970).

[18] T. H. Cormen, *Introduction to Algorithms* (MIT Press, 2009).

[19] R. Forsyth. https://archive.ics.uci.edu/ml/datasets/Zoo for "Zoo data set in 1990-05-15. UCI machine learning repository," (Accessed 2018 April 16).

[20] G. E. Hinton and J. L. McClelland, "Learning representations by recirculation," in *Neural Information Processing Systems* (American Institute of Physics, 1988), pp. 358–366.

[21] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science **313**(5786), 504–507 (2006).

[22] M. Jago, "Against yagisawa's modal realism," Analysis. **73**(1), 10–17 (2013).

[23] M. Jago, "Impossible worlds," Noûs **49**(4), 713–728 (2015).

[24] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," Inf. Fusion **14**(1), 28–44 (2013).

[25] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI* (Montreal, Canada, 1995), Vol. 14, pp. 1137–1145.

[26] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2013), pp. 8595–8598.

[27] D. Lewis, *On the Plurality of Worlds* (Oxford, 1986), Vol. 322.

[28] MultiMedia LLC, http://archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits for "Optical recognition of handwritten digits data set," 1998 (Accessed 2017 October 19).

[29] R. N. Mantegna and H. Eugene Stanley, *Introduction to Econophysics: Correlations and Complexity in Finance* (Cambridge University Press, 1999).

[30] P. Mishra and M. H. Eich, "Join processing in relational databases," ACM Comput. Surv. **24**(1), 63–113 (1992).

[31] D. P. Nolan, *Topics in the Philosophy of Possible Worlds* (Taylor & Francis, 2002).

[32] A. Pietarinen, "Impossible worlds and logical omniscience: A note on macpherson's logic of belief," in *The Tenth White House Papers Graduate Research in Cognitive and Computing Sciences at Sussex* (1998), p. 8.

[33] G. Ferilli, P. L. Sacco, and G. Tavano Blessi, *Culture 3.0. Cultural Participation as a New Driver of Social and Economic Value Creation in European Cohesion Policies* (IULM University, Milan, 2018).

[34] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on Machine Learning* (ACM, 2009), pp. 873–880.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature **323**(6088), 533 (1986).

[36] Semeion Research Centre of Sciences of Communication, http://archive.ics.uci.edu/ml/datasets/semeion+handwritten+digit for Hand-written digits 16 × 16, 2008 (Accessed 2017 October 19).

[37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," J. Mach. Learn. Res. **11**, 3371–3408 (2010).

[38] E. N. Zalta *et al.*, "A classically-based theory of impossible worlds," Notre Dame J. Formal Logic **38**(4), 640–660 (1997).

[39] J. Zhao, M. Mathieu, R. Goroshin, and Y. Lecun, "Stacked what-where auto-encoders," preprint arXiv:1506.02351 (2015).

[40] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," preprint arXiv:1609.03126 (2016).