INFORMATICA, 2023, Vol. 0, No. 0, 1–24

DOI: https://doi.org/10.15388/23-INFOR530

A New Decision Making Method for Selection of Optimal Data Using the Von Neumann-Morgenstern Theorem

Julia GARCÍA CABELLO*

Department of Applied Mathematics
Andalusian Research Institute in Data Science and Computational Intelligence
University of Granada, Spain
e-mail: cabello@ugr.es

Received: April 2023; accepted: September 2023

Abstract. The quality of the input data is amongst the decisive factors affecting the speed and effectiveness of recurrent neural network (RNN) learning. We present here a novel methodology to select optimal training data (those with the highest learning capacity) by approaching the problem from a decision making point of view. The key idea, which underpins the design of the mathematical structure that supports the selection, is to define first a binary relation that gives preference to inputs with higher estimator abilities. The Von Newman Morgenstern theorem (VNM), a cornerstone of decision theory, is then applied to determine the level of efficiency of the training dataset based on the probability of success derived from a purpose-designed framework based on Markov networks. To the best of the author's knowledge, this is the first time that this result has been applied to data selection tasks. Hence, it is shown that Markov Networks, mainly known as generative models, can successfully participate in discriminative tasks when used in conjunction with the VNM theorem.

The simplicity of our design allows the selection to be carried out alongside the training. Hence, since learning progresses with only the optimal inputs, the data noise gradually disappears: the result is an improvement in the performance while minimising the likelihood of overfitting.

Key words: data selection, prior probability, Markov networks, Von Neumann-Morgenstern Expected Utility theorem.

1. Introduction

The superiority of artificial neural networks (ANNs) in various tasks (classification, pattern identification, prediction, etc.) has led researchers to focus much of their efforts on the study of the functioning of their components from a theoretical perspective, see Higham and Higham (2019), Smale *et al.* (2010). It is well known that ANNs have a high capacity for learning, the effectiveness of which depends on many factors. Amongst them, the problem complexity influences to a high degree the ANN performance, which depends

*Correspondence to: Department of Applied Mathematics (University of Granada) Faculty of Economic and Business Sciences, Campus Cartuja s/n, 18071 Granada, Spain.

not only on the ANN architecture, but also on the accurate and sufficient training data and the efficiency that datasets show throughout the process. Training in recurrent neural networks (RNNs) – ANNs that stand out for their high capacity for learning and recognition of temporal patterns – depends to a large extent on size, type and structure of the selected training sets (Chen, 2006; Zhang and Suganthan, 2016; Zapf and Wallek, 2021): this is such a central point that decisively influences both the speed and the ability to learn. During the training phase, where the unknown parameters are to be determined, the quality and learning capacity of the selected training datasets are of key importance (Mirjalili *et al.*, 2012).

The main objective of this paper is to provide a robust methodology to select optimal training datasets (those with the highest learning capacity) that can be used in any context to maximise the performance of the trained models. This methodology has been designed to run in parallel with RNN learning so that, while the RNN learning evolves progressively only with the optimal training inputs, the data noise gradually disappears. This has a positive impact on the quality of the RNN results while minimising the likelihood of occurrence of overfitting.[1] The key idea in the design of the mathematical structure that supports this selection is to define a binary relation that gives preference to those datasets with higher estimator abilities by using Utility theory. A second contribution of our work is to have designed our methodology based on tools that have not been used previously for this. This novelty lies in showing Markov Networks (MNs), widely known as generative models (Gordon and Hernandez-Lobato, 2020), as models with a real discriminative capacity when used in conjunction with the Von Newman Morgenstern theorem (VNM theorem), a cornerstone of Game Theory with an extensive background also in Decision Theory (Machina, 1982; Delbaen *et al.*, 2011). In order to faithfully model the RNN reality, we have used the dynamic version of the MNs (TD-MRFs). It is worth noting the versatility of our proposal, which can be also applied to other data-driven methodologies provided that they are regulated by dynamical systems.

Markov Random Fields (MRFs) are also known as Markov Networks (MNs) in those contexts that require highlighting the undirected graph condition (Dynkin, 1984). MRF-type graphical models have experienced a resurgence in recent years. In its origins, they exclusively performed functions related to image processing such as restoration or reconstructing. Later works such as García Cabello (2021) or Wang *et al.* (2022) have acknowledged their high predictive capability due to the equivalence between MRFs and Gibbs distributions, which provides an explicit expression of the prior likelihood after appropriate choice of the energy functions. MRF solutions are widely regarded as generative models as opposed to discriminative approaches, more related to tasks which involve classification.

Regarding the literature review, the selection of optimal training sets has not been studied in a general framework so far. To this author's knowledge, this is the first analysis that aims to provide guidance for a general context. Published papers have studied this issue

[1] Overfitting in data-driven learning models (which extract a predictive model from a data set) is the flaw of failing to generalise the features/patterns present in the training dataset. It occurs in models which extract features from datasets having too much noise.

either only in contexts of ANN classification tasks or in very precise scenarios (electrical, financial or chemical engineering) taking advantage of their specific techniques. Within the first category, the genetic algorithm (GA) is widely used as a tool to create high-quality training sets as a the first step in designing robust ANN classifiers, see Reeves and Taylor (1998), Reeves and Bush (2001) or more recently, the paper (Nalepa *et al.*, 2018). Disadvantages of using GA, apart from slowness, include that it is computationally expensive and too sensitive to the initial conditions. In our proposal, however, the calculation of the probabilities associated with the utility (i.e. efficiency as estimators) of the inputs is very simple and therefore does not add computational cost.

Within the second category, in the paper (Zapf and Wallek, 2021), the authors made a comparison between existing methods in the area of chemical process modelling in order to split a training set from a given data set. In Wong *et al.* (2016), the authors proposed a data selection for statistical machine translations, based on recursive neural networks which can learn representations of bilingual sentences. The paper (Fernandez Anitzine *et al.*, 2012) analyses through a very context-specific instrument (ray-tracing) the ANN optimal selection of training set in the context of predicting the received power/path loss in both outdoor and indoor links. In Kim (2006), authors propose a GA approach for ANN instance selection for financial data mining.

As for the use of MNs/MRFs (prior probability) for problems which involve probability a posteriori, in the literature the terms "MNs/MRFs" and "discriminative" appear together only and exclusively to refer to discriminative random fields (DRFs) or equivalently conditional random fields (CRFs), both type of random fields which provide by definition a posterior probability.

The rest of the paper is structured as follows: preliminaries of Section 2 include basic knowledge on preference relations and VNM theorem, MNs and RNN functioning. Section 3 structures the steps to be followed to reach a solution to the proposed problem. The design of an abstract TD-MRF-based framework is performed in Section 4 which will subsequently allow the computation of prior probabilities associated with the VNM theorem. A TD-MRF structure for the input sets is also provided here. In Section 5, the expected utility theorem is applied after proving that the conditions for doing so are met. Section 6 highlights (and proves) the main results of our work. In Section 7, an example of the method application is developed. Section 8 finally concludes the paper.

2. Preliminaries

2.1. *The Von Neumann-Morgenstern Theorem*

When facing a situation of uncertainty (known as lottery), there is a set X which contains all possible outcomes (results) after the process has been completed. Each of these has associated a probability p of occurrence. The tools for managing the idea of "preferring" one outcome over another and the "benefit associated with a preference" are related to the definition of preference relation (see Jiang and Liao, 2022) and utility functions respectively.

Mathematically, a preference relation is a binary relation $\succeq$ in a set X of possible outcomes, such which is rational, i.e. that it satisfies the following properties:

- completeness: for all $x_i, x_j \in X$, either $x_i \succ x_j$ or $x_j \succ x_i$ or $x_i \sim x_j$ (indifference) and
- transitivity: for all $x_i, x_j, x_l \in X$ if $x_i \succeq x_j$ and $x_j \succeq x_l$, thus $x_i \succeq x_l$.

The instrument that allows to quantify the benefit of each possible scenario is the utility function u: they assign a numerical label to each outcome so that outcomes can be compared to make a decision.

The Von Neumann-Morgenstern expected utility theorem (VNM theorem), (Yang and Qiu, 2005; Pollak, 1967) is a simple and very efficient result in Decision Theory which allows to compare numerically (through a utility function) the possible outcomes resulting from a process under uncertainty (Van Den Brink and Rusinowska, 2022). Under some axioms the ordinal preference relation is representable by a cardinal (expected) utility function, known as VNM utility function. Moreover, the VNM theorem shows that the expected utility of a lottery can be computed as a linear combination of the corresponding utilities by using the probabilities as linear coeficients:

Theorem 2.1 (VNM Expected Utility). *Let X be a set of outcomes and a preference relation $\succeq$ on X that satisfies the hypothesis of*

- *Continuity. The following formulations of continuity are equivalent*:
  - *if each element x_n of a sequence of outcomes is $x_n \succeq x$, thus* $\lim_{n\to\infty} x_n \succeq x$,
  - $\forall x_1, x_2, x_3 \in X$ *with* $x_1 \succ x_2 \succ x_3 \Longrightarrow \exists p \in [0,1] \ni x_2 \sim [p : x_1; 1-p : x_3]$,
  - $\forall x_1, x_2, x_3 \in X$ *with* $x_1 \succ x_2 \succ x_3 \Longrightarrow \exists p \in [0,1]$ *such that* $x_2 \sim px_1 + (1-p)x_3$;
- *Independence (convex combination)*: $x_i \succeq x_j \Leftrightarrow \alpha x_i + (1-\alpha)x_l \succeq \alpha x_j + (1-\alpha)x_l$, $\forall \alpha \in (0,1]$ *and* $\forall x_l \in X$.

Thus, there exists a continuous (*utility*) *function* $u : X \to [0,1]$ *with the following properties*:

1. $x_1 \succeq x_2$ *iff* $u(x_1) \geqslant u(x_2)$;
2. $u([p_1 : x_1; p_2 : x_2; \ldots; p_m : x_m]) = \sum_{j=1}^{m} p_j u(x_j)$.

Many authors have shown, however, that in practice the axiom of independence is not fulfilled (the top paper (Machina, 1982) talks about a "systematic violation in practice" of the axiom of independence, with the famous "Allais Paradox" as example).

In the paper (Machina, 1982), it is also shown that there are weaker conditions that lead to the same results as those stated in the VNM theorem. There, continuity is replaced by the weak convergence topology, which is the weakest topology for which the expected utility functional is continuous (see also Delbaen *et al.*, 2011). On the other hand, the axiom of independence is replaced by the Fréchet differentiable condition on the functional form which defines the preferences (Machina, 1982).

2.2. *Basic Knowledge of Markov Networks MRFs*

Let $X = \{X_{s_i} | s_i \in S, i \in \mathbb{N}\}$ be a set of random variables which take X_{s_i} for any site[2] $s_i \in S$. X is known as a *stochastic process* or a *graphical model* GM with underlying set of sites S. Both X and S are used interchangeably to represent a GM.

Graphical Models are commonly used to visually describe the probabilistic relationships amongst stochastic variables. Basic knowledge on GMs comprises the concepts of neighbourhood of a site and clique: sites s_i and s_j are *adjacent*, $s_i \sim s_j$, if there is at least one edge that links them. GMs are called connected if for any two sites there is a path –a sequence of edges–, which connect them. Neighbourhood of a site s_i, denoted by $\mathcal{N}(s_i)$, is the set of sites which are adjacent to s_i: $\mathcal{N}(s_i) = \{s_j \in S | s_j \sim s_i\}$. Cliques are maximally connected subgraphs of the underlying graph S in the usual topological sense: they are connected and no more sites can be added and still be connected. Markov random fields (MRF's) are GMs whose underlying graph is *undirected*.

Dynamic graphical models (DGMs) are the time-varying version of GMs. The set of dynamic stochastic variables will be denoted by $X^t = \{X^t_{s_i} | s_i \in S, i \in \mathbb{N}\}$: these are node-dynamic graphs (the ones considered here) although edge-dynamic graphs could be also taken into account or even the possibility of both S and E varying over time. TD-MRFs are defined from a generalization of the usual markovian property. It states that the global probability of occurrence may be deduced from a local probability, when "local" refers to the neighbouring system: X^t is said to be a TD-MRF if

$$P\left[X^t_{s_i} = x_{s_i} \middle| X^t_{S-\{s_i\}} = x\right] = P\left[X^t_{s_i} = x_{s_i} \middle| X^t_{\mathcal{N}(s_i)} = x\right],$$

where $P[X^t] = P[\{X^t_{s_i} | s_i \in S, i \in \mathbb{N}\}] = \{P[X^t_{s_i} = x_{s_i}] | s_i \in S, i \in \mathbb{N}\}$ denote the joint distribution of X^t. The Hammersley-Clifford theorem, under the "positivity condition", sets the equivalence between MRF and Gibbs distribution, i.e. a joint distribution function which may be expressed in terms of functions ψ of $X^t = \{X^t_{s_i} | s_i \in S, i \in \mathbb{N}\}$, which takes values only on the cliques C, written as $\psi_C(X^t)$. The *energy functions* $\psi_C(X^t)$ determine in a clear-cut manner the joint distribution:

$$P\left[X^t\right] = \frac{1}{Z} \exp\left[-\sum_{\text{cliques } C} \psi_C\left(X^t\right)\right] = \exp\left[-\sum_{\text{cliques } C} \psi_C\left(X^t\right) - \ln Z\right],$$

such that all ψ_C depend on the clique C but have a common domain. When former expression is transformed into

$$P\left[X^t\right] = \frac{1}{Z} \prod_{\text{cliques } C} \exp\left[-\psi_C\left(X^t\right)\right], \tag{1}$$

[2]We are using the term "site" instead of node for its additional connotation of location, which allows us to simulate that each of the nodes is a different geographical place that could generate its own estimate of the variable as explained below, in Proposition 4.7.

energy functions $\psi_C(X^t)$ are called *clique potentials* when viewed as factors ($\exp[-\psi_C(X^t)]$). The function Z is known as "the partition function" which acts as a normalizing constant to ensure that the distribution sums up to 1. We shall refer to (1) as a Gibbs distribution.

2.3. *Functioning of Recurrent Neural Networks RNNs*

Neural networks (NNs) have a general functional definition as composition of parametric functions which disaggregates the linear component of the non-linear activation function (see García Cabello, 2023). Recurrent Neural networks, RNNs (Chou *et al.*, 2022; Zhang *et al.*, 2014), are a particular case of NNs which operate on time sequences and exhibit a special ability for learning lengthy-time period dependencies. Their functioning lies in an intermediary layer $h = (h^0, \ldots, h^T)$ of hidden states h^t in such a way that the data cycle through a loop to this layer. The output is reached by recursive applying the following functions:

$$\begin{aligned} h^1 &= \sigma\left(W^{xh}x^1 + W^{hh}h^0 + b^h\right) \\ \vdots \quad &\quad \vdots \\ h^t &= \sigma\left(W^{xh}x^t + W^{hh}h^{t-1} + b^h\right) \\ z^t &= \tilde{\sigma}\left(W^{hz}h^t + b^z\right) \end{aligned} \tag{2}$$

for weighted matrices, W^{xh}, W^{hh}, W^{hz}, and bias vectors, b^h, b^z, and where z^t is the final output and $\sigma, \tilde{\sigma}$ are point-wise nonlinearities (activation functions which are applied component-wise). In RNNs, activation $\tilde{\sigma}$ is the sigmoid, monotonically increasing with range (0, 1). For any RNN input x^t, its corresponding output is $z[x^t] \in (0, 1)$, denoted as z^t for simplicity.

The loss function $loss$ is chosen depending on the RNN task: usually it is Mean Square Error MSE $= \frac{1}{n}\sum_{i=1}^{n}(z_i - y_i)^2$, often used as $SE = \frac{1}{2n}\sum_{i=1}^{n}(z_i - y_i)^2$ for cancelling the constant when computing the gradient, where z_i and y_i represent the RNN output and the target value respectively. RNN functioning is shown in Fig. 1.

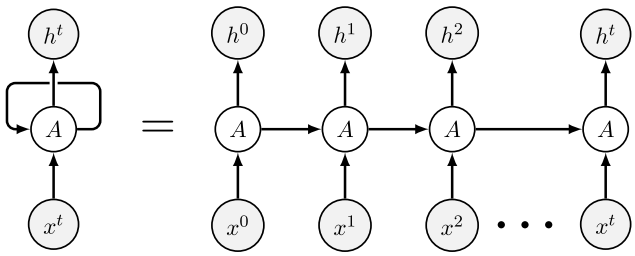

Fig. 1. RNN learning process.

3. Problem Formulation

In this section we will develop the theoretical framework that will capacitate Markov Network-methodology to perform discriminative tasks based on prior probability and the VNM Theorem 2.1. Specifically, our aim is to design a mathematical model which enables MNs to identify the optimal RNN training sets, i.e. those that produce better estimates in forecasting taks. For a better understanding, we will make reference to a real example of a RNN forecasting process:

Objective
To discriminate/select which RNN training sets will produce better estimates.
Example of RNN learning process
Let us suppose that we have to forecast the electrity price 1 month ahead ($t = +1$) and as training data we have the data of the previous 10 months ($t = -9, \ldots, -2, -1, 0$).

Recall that, in the networks as a whole, Recurrent Neural Networks, RNNs, stand out for their predictive abilities based on their potential in processing temporal data. Thus, we are facing some time-dependent learning process by using RNNs where the superscript t denotes time in all cases (no distinction will be made between matrices and their transpose in order to avoid confusion between t transpose and t time).

As is well known, in RNN prediction tasks, training sets are fed with temporal sequences composed with pairs of previous data of the form (input, labels) with the objective of predicting the future, referred to as future target value, y. Thus, X^t is the time-varying variable which needs to be estimated, Z^t is the set of RNN outputs and Y^t the set of labels (i.e. past target values for training). In terms of the illustrative example: to predict the electrity price one month ahead (target value y, $t = +1$), the training sets are made up of temporal sequences (x_i^t, y_i^t) whose first component contains different factors that influence electricity prices (fuel prices, transmission and distribution costs, weather conditions, etc.) and whose second component is the electrity price, both x_i^t, y_i^t in site i and prior to present time ($t = -9, \ldots, -2, -1, 0$).

Let In be the (time-dependent) input RNN dataset formed by n vectors:

$$In = \left\{x_i^t = \left(x_i^{1,t}, x_i^{2,t}, \ldots, x_i^{d,t}\right),\ i = 1, \ldots, n\right\},$$

each of them contains d features: $x_i^{j,t} \in \mathbb{R}$, $i = 1, \ldots, n$; $j = 1, \ldots, d$:

RNN input	RNN output	Labels	Training set Tr
x_i^t	z_i^t	y_i^t	$\{(x_i^t, y_i^t);\ i = 1, \ldots, D\}$

$Z^t[In]$ will stand for the set of outputs which is generated after the completed RNN process corresponding the the set of inputs In.

The objective is to select from the n inputs in In, $D < n$ vectors that will make up the training set: the selection criterion is to choose those which produce the best estimates in

the RNN learning process. The choice will be conducted by only using prior probability –instead of posterior– and the VNM Theorem 2.1.

We will list the steps to be taken:

- Main lottery (in In). The key idea is to adapt the scenario of the RNN learning process (inputs, outputs, labels, target value) to the situation described in the VNM-Theorem 2.1, and to rely on the TD-MRF methodology for computing the probabilities. To do so, we consider the process of selecting $D < n$ vectors as a lottery whose outcomes are the D vectors we have chosen for the training set (main lottery). Each of these D vectors has its own likelihood of being chosen.
- Consider the RNN process as composed by several uncertain processes, RNN_i, one for each of the input vectors x_i^t that make up the set In, $i = 1, \ldots, n$ (Fig. 1). After the RNN learning is completed, a set of n outputs $Z^t[In]$ is generated. Note that, since RNN is *deterministic* (the same set of inputs will produce the same set of outputs), input x_i^t has a uniquely associated output z_i^t.
- Secondary lottery (for each $z_i^t \in Z^t[In]$). A second lottery arises *for each* output $z_i^t \in Z^t[In]$ when the selection criterion is applied: how good the estimate z_i^t is. If M denotes the threshold below which the loss function is considered acceptable, the secondary lottery is whether the loss function $\text{loss}(z_i^t) < M$ or not.

  To ensure that our selection rule is applied, we define a preference relation on $In = \{x_i^t,\ i = 1, \ldots, n\}$. Due to the unique existing direction from input x_i^t to RNN output z_i^t, the preference relation can be considered defined on both the set of inputs In and the set of outputs $Z^t[In]$. Moreover, for the evident bidirection between outcome z_i^t and associated loss $\text{loss}(z_i^t)$, the distinction that many authors make between defining a preference relation on the set of outcomes X or defining it on the set of lotteries over X, named ΔX, does not apply in our case:

  Definition 3.1. For $z_i^t, z_j^t \in Z^t[In]$, $z_i^t \prec z_j^t \Leftrightarrow \text{loss}(z_j^t) < \text{loss}(z_i^t)$, i.e. z_j^t is preferred to z_i^t if z_j^t is a better estimate (smaller associated loss). Indifference comes with the equality: for $z_i^t, z_j^t \in Z^t[In]$ $z_i^t \sim z_j^t \Leftrightarrow \text{loss}(z_i^t) = \text{loss}(z_j^t)$.

  In later sections we will show that this preference relation verifies the properties necessary for the application of the VNM theorem.
- We thus apply Theorem 2.1 for the main lottery: the expected utility of training set Tr is given by

$$u[Tr] = u\left[\left\{x_1^t, \ldots, x_D^t\right\}\right] = \sum_{i=1}^{D} po\left[X^t = x_i^t\right] \cdot u\left(x_i^t\right). \tag{3}$$

- In order to compute the (prior) probabilities $po[X^t = x_i^t]$, we shall equip the initial RNN data set with an undirected graph structure –with appropriate node and edge definitions– which will later be shown to be an MRF by application of central Theorem 4.4. This will be the development of Section 4.
- In order to compute the utility $u(x_i^t)$, we shall apply again Theorem 2.1 for the secondary lottery. This will be performed in Sections 5 and 6.

4. The Abstract TD-MRF-Based Framework

Here, we first design an abstract graph-based framework that shall provide a model for a dynamic context of k sites and r filters for data discrimination (corresponding to r cliques) for which it will be shown that it is an TD-MRF under certain mild conditions. Such TD-MRF will be the core in the computation of prior probabilities $po[X^t = x_i^t]$ of the VMN Theorem 2.1.

Let $\mathcal{S}$ be a set of k sites s_i, $\mathcal{S} = \{s_i | i = 1, 2, \ldots, k\}$ such that the variable may take different values depending on the site, $X_{s_i}^t$. For each site s_i, the variable $X_{s_i}^t$ can be disaggregated into a set of d characteristics which fully describes $X_{s_i}^t$ in the instant of time t: $X_{s_i}^t = (X_{s_i}^{1,t}, X_{s_i}^{2,t}, \ldots, X_{s_i}^{d,t})$, with lower case $x_{s_i}^t = (x_{s_i}^{1,t}, x_{s_i}^{2,t}, \ldots, x_{s_i}^{d,t})$ for the feature vector (i.e. a numerical vector corresponding to a realisation of the variable). We shall use indistinctly $x_{s_i}^t = (x_{s_i}^{1,t}, x_{s_i}^{2,t}, \ldots, x_{s_i}^{d,t})$ or $x_i^t = (x_i^{1,t}, x_i^{2,t}, \ldots, x_i^{d,t})$ either for upper and lower case. In a compact form, $x_i^{j,t} \in \mathbb{R}, i = 1, \ldots, k;\ j = 1, \ldots, d$.

Remember that two random variables are equivalent if they have identical distribution. Then, we will define the edges by equivalently defining the neighbourhood $\mathcal{N}$ of a site: $\mathcal{N}(X_{s_i}^t) = \{X_{s_j}^t |\ X_{s_j}^t, X_{s_i}^t \text{ are equivalent}\} = \{X_{s_j}^t | P[X_{s_j}^t \leqslant x] = P[X_{s_i}^t \leqslant x]\, \forall x\}$.

Definition 4.1 (*TD-DGM*). A DGM $(\mathcal{S}^t, \mathcal{N}^t)$ is defined as follows: sites are the elements of the set $\mathcal{S}^t$ through the identification $s_i^t \sim X_{s_i}^t$. Edges are defined through the neighbourhood $\mathcal{N}$ of a site s_i^t as $\mathcal{N}(X_{s_i}^t) = \{X_{s_j}^t | P[X_{s_j}^t \leqslant x] = P[X_{s_i}^t \leqslant x]\, \forall x\}$.

Remark 4.2 (*Clean data*). It is worth highlighting that Definition 4.1 (that makes equal all random variables with identical probability distribution) avoids duplicates. This is particularly important when applied to the graphical model resulting from an RNN input dataset (clean data).

Recall that marginal distribution is also known as prior probability in contrast with the posterior distribution (the conditional one). From the former definition, sites in the same neighbourhood have the same prior probability. Moreover,

Proposition 4.3. *Sites which belong to the same neighbourhood have identical probability a posteriori.*

Proof. Let s_i, s_j be two sites which belong to the same neighbourhood. From the Bayes's theorem, one has

$$P\left[X_{s_i}^t | X_{s_j}^t\right] = \frac{P[X_{s_j}^t | X_{s_i}^t] \cdot \cancel{P[X_{s_i}^t]}}{\cancel{P[X_{s_j}^t]}} = P\left[X_{s_j}^t | X_{s_i}^t\right]. \qquad \square$$

The following theorem proves then that the DGM defined in Definition 4.1 is a TD-MRF by equivalently showing that the Markov condition:

Theorem 4.4 (The TD-MRF model). *DGMs as in Definition 4.1 are TD-MRFs.*

Proof. We shall prove that the local dynamic Markov property is verified, i.e. the probability of X_{s_i} conditioned to the remaining random variables, X_{s_j}, $j \neq i$, is equal to the probability of X_{s_i} conditioned to the random variables in the same neighbouring system $X_{\mathcal{N}(s_i)}$. The following development is considered in a specific time instant t_0. By assuming that the s_i-site estimation takes value $x_{s_i}^{t_0}$ while the rest does not (thus, their estimation is some $x \neq x_{s_i}^{t_0}$), we partition the set $\mathcal{S}$ as $\mathcal{S} - \{s_i\} = \mathcal{N}(s_i) - \{s_i\} \cup \mathcal{N}_{ot}(s_i)$, where $\mathcal{N}_{ot}(s_i)$ denotes those sites which are not in $\mathcal{N}(s_i)$. Thus,

$$P\big[X_{s_i}^{t_0} = x_{s_i}^{t_0} \big| X_{\mathcal{S}-\{s_i\}}^{t_0} = x \neq x_{s_i}^{t_0}\big] = \frac{P[(X_{s_i}^{t_0} = x_{s_i}^{t_0}) \cap (X_{\mathcal{S}-\{s_i\}}^{t_0} = x \neq x_{s_i}^{t_0})]}{P[X_{\mathcal{S}-\{s_i\}}^{t_0} = x \neq x_{s_i}^{t_0}]} =$$

$$= \frac{P[(X_{s_i}^{t_0} = x_{s_i}^{t_0}) \cap (X_{\mathcal{N}(x_{s_i}^{t_0})}^{t_0} = x \neq x_{s_i}^{t_0})]}{P[X_{\mathcal{N}(x_{s_i}^{t_0})}^{t_0} = x \neq x_{s_i}^{t_0}]} =$$

$$= P\big[X_{s_i}^{t_0} = x_{s_i}^{t_0} \big| X_{\mathcal{N}(x_{s_i}^{t_0})}^{t_0} = x \neq x_{s_i}^{t_0}\big]. \qquad \square$$

Insofar as the distribution function is the tool used to make estimates, previous Theorem 4.4 provides a joint measure of how close the variable is to taking a particular value.

Corollary 4.5. *The corresponding Gibbs (joint) probability distribution at a time instant t provides that the likelihood of reaching a concrete value x is* $P[X^t = x] = \frac{1}{Z}\prod_{\substack{cliques\ C_j \\ j=1}}^{r} e^{-\psi_C(X^t=x)}$.

Remark 4.6. Under Definition 4.1, neighbours and cliques are essentially the same and equal to the set of random variables with identical prior distribution. Moreover, according to Proposition 4.3, variables in a clique have also the same posterior distribution.

Each clique has its own common estimation function:

Proposition 4.7 (The cliques). *Each clique of the TD-MRF has its own estimation function given by the clique potentials* $e^{-\psi_{C_j}([.])}$, $j = 1, \ldots, r$: $P_{C_j}[X^t = x] = \frac{1}{Z} e^{-\psi_{C_j}(X^t=x)}$.

Proof. It is straightforward from definition of clique. $\square$

In discrimination/classification works, the commonly used probability is the conditional or posterior probability $P[X^t = x|y] = \frac{1}{Z}\prod_{\substack{\text{cliques } C_j \\ j=1}}^{r} e^{-\psi_C(X^t=x|y)}$ where x stands for RNN input, y represents the corresponding target value and $\psi_C(X^t = x|y)$ means that the energy function ψ_C only assigns to those inputs x that correspond to y.

4.1. *Visual Flowchart of the TD-MRF Operational Process*

Inputs for a TD-MFRs are specific values of the stochastic variable X^t in a particular time instant t and a site s_i, $x_{s_i}^t$. Depending on the context, X^t can be considered as disaggre-

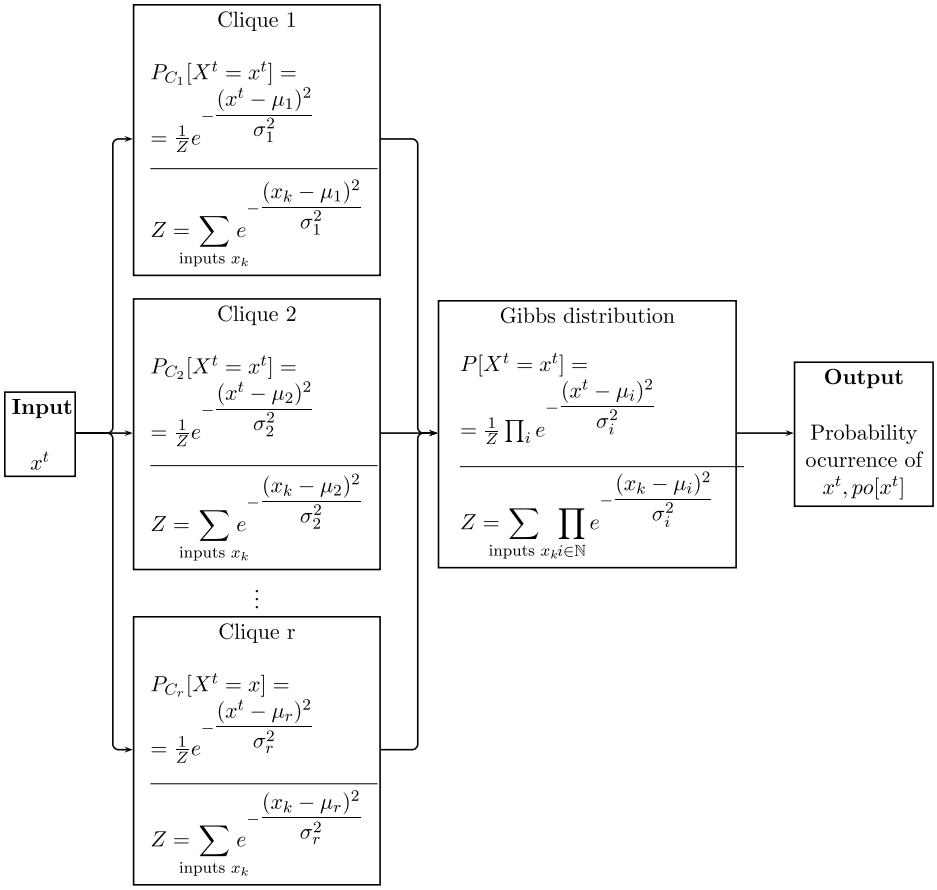

Fig. 2. TD-MRF operational process.

gated into a set of d features for each site s_i. Thus, each input is $x_i^t = (x_i^{1,t}, x_i^{2,t}, \ldots, x_i^{d,t})$, $x_i^{j,t} \in \mathbb{R}, i = 1, \ldots, k, j = 1, \ldots, d$ such that each univariate component $x_i^{j,t} \in \mathbb{R}$ can be regarded as input of a single-variable for d processes. The TD-GDM operational process is as follows: by assuming there are r cliques, $C_1, \ldots, C_r$, each input is processed in a clique $\mathcal{C}_j$, $j = 1, \ldots, r$, through the corresponding clique potential $e^{-\psi_{C_j}([.])}$, $j = 1, \ldots, r$, such that the output is the estimate made by the corresponding clique (according to Proposition 4.7). Each of these is thus aggregated in a final output by means of the expression given in Corollary 4.5. If, on the other hand, the variable is not considered disaggregated, multivariate energy functions will process the information in a single multivariate execution.

Figure 2 provides a visual representation of a TD-MRF operation in the univariate scenario, where the energy functions ψ are of Gaussian type, i.e. $\psi_{C_j} = \frac{([.]-\mu_j)^2}{\sigma_j^2}$, $j = 1, \ldots, r$. Note that for $x_i^t = (x_i^{1,t}, x_i^{2,t}, \ldots, x_i^{d,t})$, the corresponding probability $po[x_i^t]$ is $po[x_i^t] = (po[x_i^{1,t}], po[x_i^{2,t}], \ldots, po[x_i^{d,t}])$.

Moreover, the operation of an MRF is equally applied in the calculation of following probabilities:

$$P_{C_j}\big[X^t \leqslant x^t\big] = \sum_{x \leqslant x^t} P\big[X^t = x\big],$$
$$P_{C_j}\big[x_{pr} < X^t \leqslant x_{im}\big] = \sum_{x \leqslant x_{im}} P\big[X^t = x\big] - \sum_{i \leqslant x_{pr}} P\big[X^t = x\big],$$
$$P_{C_j}\big[X^t > x^t\big] = 1 - P_{C_j}\big[X^t \leqslant x^t\big] = 1 - \sum_{x \leqslant x^t} P\big[X^t = x\big].$$

4.2. *TD-MRF Structure for the RNN Input Set*

Our goal here is to provide a graph-structure (which will become TD-MRF structure according to Theorem 4.4) for the set *In* of RNN inputs $In = \{x_i^t = (x_i^{1,t}, x_i^{2,t}, \ldots, x_i^{d,t}), i = 1, \ldots, n\}$, each $x_i^{j,t} \in \mathbb{R}$, $i = 1, \ldots, n$; $j = 1, \ldots, d$.

To achieve this, the steps to follow are listed below:

1. **The pre-processing phase**. Before training/testing an RNN, data must suffer some preprocessing steps for removing duplicates, unnecessary information and simulating missing data. Depending on the context, data must also be normalised with feature scaling. We shall assume that RNN input datasets have completed this phase.
2. **The input set graph structure.** We endow the RNN data set with an undirected graph structure, according to Definition 4.1, where $k = D$:
   - sites s_i^t are the local version of the random variable X^t, i.e. the random variable X_i^t for which the set of inputs x_i^t are a realisation of it: $s_i^t = X_{s_i}^t$.
   - The neighbourhood of a site is defined as

$$\begin{aligned}\mathcal{N}\big(X_{s_i}^t\big) &= \big\{X_{s_j}^t \big| X_{s_j}^t, X_{s_i}^t \text{ are equivalent}\big\} \\ &= \big\{X_{s_j}^t \big| P\big[X_{s_j}^t \leqslant x\big] = P\big[X_{s_i}^t \leqslant x\big] \forall x\big\}.\end{aligned}$$

     in the sense of Definition 4.1. According to Remark 4.2, this definition is intended for discarding duplicates in the datasets.
3. **The corresponding Gibbs distribution**. According to Theorem 4.4, the graphical model formed by the RNN input data viewed as a time varying random variable X^t is TD-MRF, with a Gibbs distribution $P[X^t] = \frac{1}{Z} \prod_{\text{cliques } C} \exp[-\psi_C(X^t)]$.
4. **Likelihood of occurrence of an output.** Recall that from Definition 4.1, neighbours are equal to cliques and both consist of those random variables which have equal prior distributions (and identical posterior distribution in consequence). Hence, a probability $po[X^t = x_i^t]$ is biunivocally determined for each input x_i^t. We also refer to Fig. 2 to reproduce the operation of an TD-MRF in a single multivariate execution.

   Note that the deterministic nature of RNN, which always assigns the same z_i^t to the same x_i^t, allows also defining the probability of occurrence of an RNN output $po[z_i^t]$

as

$$po\left[z_i^t\right] := po\left[X^t = x_i^t\right].$$

5. Application of the VNM Utility Theorem

In this section, we will further develop equation (3),

$$u[Tr] = u\left[\left\{x_1^t, \ldots, x_D^t\right\}\right] = \sum_{i=1}^{D} po\left[X^t = x_i^t\right] \cdot u\left(x_i^t\right),$$

by giving the explicit calculation of $u(x_i^t)$, $\forall i, t$. To do so, we will again apply the VMN theorem to the secondary lottery.

Recall that in the preceding sections we have considered a second lottery which arises naturally when we test how good the ouput $z_i^t \in Z^t[In]$ is as estimate. If M is the threshold below which the loss function is considered acceptable, the secondary lottery is whether the loss function $\text{loss}(z_i^t) < M$ or not. With the intuitive notation of the paper (Machina, 1982), our secondary lottery is a probability distribution function over the interval $[0, M]$ (the set of all lotteries over $[0, M]$ is denoted in Machina (1982) by $D[0, M]$) and the preference functional $V(\cdot)$ on $D[0, M]$ in our case is $\text{loss}(\cdot)$. Note that Definition 3.1 of preference over the set $Z^t[In]$ can be additionally considered over In for the deterministic assignment $x_i^t \to z_i^t$: for $x_i^t, x_j^t \in In$

$$x_i^t \prec x_j^t \Leftrightarrow z_i^t \prec z_j^t \Leftrightarrow \text{loss}\left(z_j^t\right) < \text{loss}\left(z_i^t\right),$$

where the loss function is considered in a squared-form, $\text{loss}(z_i^t) = (z_i^t - y_i)^2$ which in the context or real positive numbers is equivalent to $\text{loss}(z_i^t) = |z_i^t - y_i|$. The objective here is to prove that this binary relation satisfies the necessary conditions that enable the application of the Von Neumann-Morgenstern (VNM) theorem.

First of all, it has to be shown that it is a preference relation:

Proposition 5.1. *The relation defined in Definition 3.1 is a preference relation.*

Proof. The standard axiom for a preference relation is rationality which includes both completeness and transitivity.

- Completeness: for all $z_i^t, z_j^t \in Z^t[In]$, either $z_i^t \preceq z_j^t$ or $z_i^t \succeq z_j^t$. This property comes from the fact that $\mathbb{R}$ has a total order when applying to $|z_i^t - y_i|, |z_j^t - y_i| \in \mathbb{R}$.
- Transitivity: for all $z_i^t, z_j^t, z_l^t \in Z^t[In]$, if $z_i^t \preceq z_j^t$ and $z_j^t \preceq z_l^t$, thus $z_i^t \preceq z_l^t$. This property holds by the transitivity in the order of $\mathbb{R}$: $|z_i^t - y_i| \leqslant |z_j^t - y_i| \leqslant |z_l^t - y_i|$ $\Rightarrow |z_i^t - y_i| \leqslant |z_l^t - y_i|$. □

Moreover, former preference relation satisfies the following conditions:

Proposition 5.2. *The preference relation in Definition 3.1 satisfies*:

- *Continuity: if z_n^t of a sequence of outcomes with $z_n^t \succeq z$, thus* $\lim_{n\to\infty} z_n^t \succeq z$.
- *Fréchet differentiable*: *the functional form which defines de preferences* (*the loss function*) *must be Fréchet differentiable.*

Proof. Continuity: let us prove that if each element z_n^t of a sequence of outcomes is $z_n^t \succeq z$, thus it must be $\lim_{n\to\infty} z_n^t \succeq z$. First, we have that

$$\begin{aligned}
\left(z_n^t - y_n\right)^2 &= \left(z_n^t\right)^2 - 2z_n^t \cdot y_n + y_n^2 \Rightarrow \\
\lim_{n\to\infty}\left(z_n^t - y_n\right)^2 &= \lim_{n\to\infty}\left(z_n^t\right)^2 - \lim_{n\to\infty}\left(2z_n^t \cdot y_n\right) + \lim_{n\to\infty} y_n^2 = \\
&= \lim_{n\to\infty}\left(z_n^t\right)^2 - 2\lim_{n\to\infty} z_n^t \cdot \lim_{n\to\infty} y_n + \lim_{n\to\infty} y_n^2 = \\
&= \left(\lim_{n\to\infty} z_n^t - \lim_{n\to\infty} y_n\right)^2.
\end{aligned}$$

Let us suppose that z_n^t is a sequence of outcomes such that $z_n^t \succeq z$, that is, output z is preferred to outputs z_n^t. That means that $(z-y)^2 < (z_n^t - y_n)^2$ where y, y_n are the corresponding target values for the inputs that correspond to z and z_n^t, respectively.

By assuming that both limits exist, the inequality $(z-y)^2 < (z_n - y_n)^2$ is preserved in "less than or equal" form: $\lim_{n\to\infty}(z-y)^2 = (z-y)^2 \leqslant \lim_{n\to\infty}(z_n^t - y)^2$. Hence,

$$(z-y)^2 \leqslant \lim_{n\to\infty}\left(z_n^t - y\right)^2 = \left(\lim_{n\to\infty} z_n^t - \lim_{n\to\infty} y_n\right)^2 \Rightarrow \lim_{n\to\infty} z_n^t \succeq z.$$

Fréchet differentiable. Recall that the Fréchet derivative in finite-dimensional spaces is the usual derivative. Thus, the loss function $\text{loss}(z_i^t) = (z_i^t - y)^2$ verifies this hypothesis. □

Therefore, the preference relation stated in Definition 3.1 verifies the conditions required for the application of the VNM Theorem 2.1. Thus, there exists an utility function $u : In \to [0,1]$ which quantifies the preferences: $x_1^t \succeq x_2^t$ iff $u(x_1^t) \geqslant u(x_2^t)$. Moreover, the Von Neumann-Morgenstern Expected Utility theorem also provides guidance for computing the utility of the set of inputs through the explicit formula given in equation (3):

$$u[ii] = u\left[\left\{x_1^t, \ldots, x_D^t\right\}\right] = \sum_{i=1}^{D} po\left[X^t = x_i^t\right] \cdot u\left(x_i^t\right).$$

6. Main Results

The objective of this section is to highlight (after proving) the main results of our proposal. First of all, we define the level of efficiency of an input set *In*, Eff[*In*]. Thus, next Theorem 6.2 proves the existence of a formula for determining the efficiency of a training set while Theorem 6.3 gives an explicit expression for computing Eff[*In*].

DEFINITION 6.1. We define the level of efficiency of an input set *In*, Eff[*In*] as the expected utility of training set *Tr*: $\text{Eff}[In] := u[\{x_1^t, \ldots, x_D^t\}] = \sum_{i=1}^{D} po[X^t = x_i^t] \cdot u(x_i^t)$.

Theorem 6.2 (Existence). *For any input set In, there exists an utility function* $u : In \to [0, 1]$ *which allows to quantify its level of efficiency Eff*[*In*] *such that this can be computed as*

$$Eff[In] = \sum_{i=1}^{D} po\big[X^t = x_i^t\big] \cdot u\big(x_i^t\big).$$

Proof. Following Proposition 5.2, the preference relation stated in Definition 3.1 verifies the necessary conditions for applying the VNM Theorem 2.1. According to this, there exists an utility function $u : In \to [0, 1]$, which quantifies the preferences over input vectors: $x_1^t \succeq x_2^t$ iff $u(x_1^t) \geqslant u(x_2^t)$. □

Theorem 6.3 (How to compute the level of efficiency of *In*). *We assume that all inputs are equally distributed. Thus, for any RNN input set In, its level of efficiency can be explicitely computed as*

$$Eff[In] = p \cdot \sum_{i=1}^{D} po\big[X^t = x_i^t\big], \text{ where } p \text{ is the probability that } |z_i^t - y| < M, \ \forall z_i^t.$$

Proof. We start from expression $\text{Eff}[In] = \sum_{i=1}^{D} po[X^t = x_i^t] \cdot u(x_i^t)$ derived from Theorem 6.2. On one hand, the explicit computation of probabilities $\{po[X^t = x_i^t]\}_{i=1}^{D}$ is given by the TD-MRF operational process defined in Section 4 (see Fig. 2, which visually describes this). On the other hand, in order to compute the utilities $\{u(x_i^t)]\}_{i=1}^{D}$, we shall apply again the VNM Theorem 2.1 for the secondary lottery. Recall that the secondary lottery for each z_i^t is whether the loss function $|z_i^t - y| < M$ or not, for a given threshold M below which the loss function is considered acceptable. Recall also that the probability of occurrence of an RNN output $po[z_i^t]$ is $po[z_i^t] = po[X^t = x_i^t]$. To overcome a continuous space of outcomes $\{|z_i^t - y| \in \mathbb{R}\}_{i=1}^{D}$ for the lottery associated to each z_i^t, we represent it in a binary form:

$$\begin{cases} \text{outcome 1} & \text{if } |z_i^t - y| < M, \\ \text{outcome 0} & \text{if } |z_i^t - y| \geqslant M. \end{cases}$$

Since the utility function u represents a set of outcomes in the sense of VNM-theorem 2.1 ($x_1 \succeq x_2$ iff $u(x_1) \geqslant u(x_2)$), we can conclude that

$$\begin{cases} \text{outcome 1} & \text{if } |z_i^t - y| < M \Rightarrow u(1) = 1, \\ \text{outcome 0} & \text{if } |z_i^t - y| \geqslant M \Rightarrow u(0) = 0. \end{cases}$$

Then, by application of VNM-Theorem 2.1 on each secondary lottery, one has that

$$\begin{aligned} u\big(x_i^t\big) &= u\big(z_i^t\big) \\ &= \sum_{i=1}^{2} p[\text{outcome}_i] \cdot u(\text{outcome}_i) = p[\text{outcome 1}] \cdot 1 = p[\text{outcome 1}], \\ &\forall i = 1, \ldots, D. \end{aligned}$$

The formula comes then by substituting:

$$\begin{aligned} \text{Eff}[In] &= \sum_{i=1}^{D} po\big[X^t = x_i^t\big] \cdot u\big(x_i^t\big) = \\ &= \sum_{i=1}^{D} po\big[X^t = x_i^t\big] \cdot p[\text{outcome 1}] = \\ &= p[\text{outcome 1}] \cdot \sum_{i=1}^{D} po\big[X^t = x_i^t\big] = \\ &= p \cdot \sum_{i=1}^{D} po\big[X^t = x_i^t\big]. \end{aligned}$$

□

7. A Case Application

This section is aimed at developing an example of the method application. In order to make a choice between two real data sets, their level of efficiency will be computed. As stated in Definition 6.1, the level of efficiency of an input set Eff[*In*] measures the learning capacity of its inputs on the basis of the preference relation of Definition 3.1 that prioritises those inputs whose outputs have a lower associated loss (better estimates). To compute Eff[I_n] we shall apply the formula proved in Theorem 6.3 which follows from Theorem 6.2, in which the existence of an utility function u which quantifies the preference relation was proved. This is

$$\text{Eff}[In] = \sum_{i=1}^{D} po\big[X^t = x_i^t\big] \cdot u\big(x_i^t\big) = p \cdot \sum_{i=1}^{D} po\big[X^t = x_i^t\big],$$

where p is the probability that $|z_i^t - y| < M, \forall z_i^t$ for a given threshold M below which the loss function is considered acceptable.

The data sets we shall use here contain prices (€/1 kg) over time for the most common olive oil varieties. These data are available on the websites of the Government of Spain: https://www.mapa.gob.es/es/agricultura/temas/producciones-agricolas/aceite-oliva-y-aceituna-mesa/Evolucion_precios_AO_vegetales.aspx,

where prices are published on a weekly basis. Specifically, we will consider data sets corresponding to weeks 28/2023 and 39/2022 (the reasons for this choice will be explained later). We shall thus apply the above formula taking into account the following considerations. On one hand, note that the choice of the threshold M will depend on the context. In the olive oil market scenario, assuming a deviation from the olive oil price of 10% is acceptable. Hence, $M = 0.5$. On the other hand, the value of the parameter p will depend on several market features which vary over time depending on physical (rainfall, pollen levels...) and socio-economic (Government regulations) circumstances. When real prices are subject to unusual fluctuations (due to changes in the aforementioned circumstances), such prices used in training tasks will deviate more from the real ones. In consequence, the probability p, such that $|z_i^t - y| < M, \forall z_i^t$, will decrease. Either way, it should be noticed that p is known as soon as the value of M is fixed, but it is not the same for all input sets.

From the above formula, since M is known (and therefore so is p), we must focus on computing $po[X^t = x_i^t]$ for each input x_i^t. To that end, we will follow Fig. 2 of the TD-MRF operational process given in Section 4.1. Since each input x_i^t may be disaggregated into d features $x_i^{k,t}$ $(k = 1, \ldots, d)$, $x_i^t = (x_i^{1,t}, x_i^{2,t}, \ldots, x_i^{d,t})$, the same is true for the corresponding probability $po[X^t = x_i^t] = po[x_i^t]$, which is $po[x_i^t] = (po[x_i^{1,t}], po[x_i^{2,t}], \ldots, po[x_i^{d,t}])$.

We assume that the energy functions ψ corresponding to the r cliques are of Gaussian type, i.e. $\psi_{C_j} = \frac{([.]-\mu_j)^2}{\sigma_j^2}$, $j = 1, \ldots, r$, since Gaussian distributions are suitable in the olive oil scenario. Actually, given that Gaussian distributions portray those data sets whose majority of elements revolves around the centre, energy functions of Gaussian type are particularly suited for goods whose price takes values in an interval of small length and do not suffer very sharp price variations.

In order to achieve our goal, each input x_i^t must be first processed in each clique $\mathcal{C}_j$, $j = 1, \ldots, r$, through the corresponding clique potential, whose result is

$$P_{C_j}\left[X^t = x_i^t\right] = \frac{1}{Z} e^{-\frac{(x_i^t - \mu_j)^2}{\sigma_j^2}} \Rightarrow P_{C_j}\left[X^t = x_i^t\right] = e^{-\frac{(x_i^t - \mu_j)^2}{\sigma_j^2}}, \quad \text{when } Z = 1.$$

Once the processing in the cliques has been completed, the required probability is obtained as

$$po\left[x_i^t\right] = P\left[X^t = x_i^t\right] = \frac{1}{Z} \prod_j^r P_{C_j}\left[X^t = x_i^t\right] \Rightarrow P\left[X^t = x_i^t\right] = \prod_j^r e^{-\frac{(x_i^t - \mu_j)^2}{\sigma_j^2}},$$

when $Z = 1$.

The computation of "the partition function" Z entails certain difficulties in practice. For this reason, we shall adopt the view commonly taken in the literature that $Z = 1$.

From the TD-MRF structure proved in Theorem 4.4, cliques gather those random variables with identical prior and posterior distribution (see Remark 4.6). This theoretical description fits with the specialist major retailers in the olive oil context. In this

Table 1
In_1: Processing in the clique C_1.

x_i^t	$(x_i^t - 3.988)$	$(x_i^t - 3.988)^2$	$\frac{(x_i^t-3.988)^2}{0.172}$	$-\frac{(x_i^t-3.988)^2}{0.172}$	$e^{-\frac{(x_i^t-3.988)^2}{0.172}}$
3.16	−0.828	0.685584	3.985953488	−3.985953488	0.018574725
5.92	1.932	3.732624	2.86683871	−2.86683871	0.056878452
6.26	2.272	5.161984	3.96465745	−3.96465745	0.018974535
6.52	2.532	6.411024	4.923981567	−4.923981567	0.007270127
7.1	3.112	9.684544	7.438205837	−7.438205837	0.00058834

Table 2
In_1: Processing in the clique C_2.

x_i^t	$(x_i^t - 3.884)$	$(x_i^t - 3.884)^2$	$\frac{(x_i^t-3.884)^2}{0.156}$	$-\frac{(x_i^t-3.884)^2}{0.156}$	$e^{-\frac{(x_i^t-3.884)^2}{0.156}}$
3.16	−0.724	0.524176	3.360102564	−3.360102564	0.034731697
5.92	2.036	4.145296	26.57241026	−26.57241026	2.88236E−12
6.26	2.376	5.645376	36.18830769	−36.18830769	1.9214E−16
6.52	2.636	6.948496	44.54164103	−44.54164103	4.52701E−20
7.1	3.216	10.342656	66.29907692	−66.29907692	1.60945E−29

practical case, the level of efficiency shall be computed through $P_{C_j}[X^t = x_i^t]$ of cliques C_j, $j = 1, \ldots, 7$ supported by the information given in Table 8 below (source: https://www.olimerca.com/precios/tipoInforme/3).

As discussed before, there are multiple factors (physical and socio-economic) that influence the price. Such factors are the features $x_i^{k,t}$ $(k = 1, \ldots, d)$ of each input $x_i^t = (x_i^{1,t}, x_i^{2,t}, \ldots, x_i^{d,t})$. For simplicity, we focus on just one of them in order to choose the two input sets In_1, In_2: the hydrographic index, which shows the average rainfall over a certain period of time. In this line, In_1 is an input set of prices (€/1 kg) under severe drought conditions (and therefore, with unusual fluctuations in price as product shortages raise the prices[3]) while In_2 is an input set of prices which correspond to a usual period of rainfall:

$$In_1 = \{3.16, 5.92, 6.26, 6.52, 7.10\}, \ p = 0.23, \quad \text{week } 28/2023,$$
$$In_2 = \{2.71, 3.78, 3.80, 3.86, 3.96\}, \ p = 0.57, \quad \text{week } 39/2022.$$

With this choice of In_1 and In_2, it is to be expected that the inputs in In_1 will produce worse estimators (higher associated loss) since such inputs reflect prices with unusual fluctuations (therefore with higher deviation from the mean). Hence, it is is to be expected that $\text{Eff}[In_2] > \text{Eff}[In_1]$.

The computation of level of efficiency In_1 is supported by the information provided in Tables 1–7.

From the information provided by the above tables, finally the required probability is computed (see Table 9).

[3]In dry seasons, water shortages lead to a drop in the olive production and, therefore, in the olive oil production. Hence, under severe drought conditions, olive oil prices skyrocket.

Table 3
In_1: Processing in the clique C_3.

x_i^t	$(x_i^t - 3.851)$	$(x_i^t - 3.851)^2$	$\frac{(x_i^t-3.851)^2}{0.143}$	$-\frac{(x_i^t-3.851)^2}{0.143}$	$e^{-\frac{(x_i^t-3.851)^2}{0.143}}$
3.16	−0.691	0.477481	3.339027972	−3.339027972	0.03547142
5.92	2.069	4.280761	29.93539161	−29.93539161	9.98216E−14
6.26	2.409	5.803281	40.58238462	−40.58238462	2.37298E−18
6.52	2.669	7.123561	49.81511189	−49.81511189	2.32045E−22
7.1	3.249	10.556001	73.81818881	−73.81818881	8.73309E−33

Table 4
In_1: Processing in the clique C_4.

x_i^t	$(x_i^t - 3.858)$	$(x_i^t - 3.858)^2$	$\frac{(x_i^t-3.858)^2}{0.108}$	$-\frac{(x_i^t-3.858)^2}{0.108}$	$e^{-\frac{(x_i^t-3.858)^2}{0.108}}$
3.16	−0.698	0.487204	4.511148148	−4.511148148	0.01098584
5.92	2.062	4.251844	39.36892593	−39.36892593	7.98533E−18
6.26	2.402	5.769604	53.42225926	−53.42225926	6.29517E−24
6.52	2.662	7.086244	65.61337037	−65.61337037	3.19503E−29
7.1	3.242	10.510564	97.32003704	−97.32003704	5.42556E−43

Table 5
In_1: Processing in the clique C_5.

x_i^t	$(x_i^t - 4.343)$	$(x_i^t - 4.343)^2$	$\frac{(x_i^t-4.343)^2}{0.472}$	$-\frac{(x_i^t-4.343)^2}{0.472}$	$e^{-\frac{(x_i^t-4.343)^2}{0.472}}$
3.16	−1.183	1.399489	2.965019068	−2.965019068	0.051559486
5.92	1.577	2.486929	5.268917373	−5.268917373	0.005149182
6.26	1.917	3.674889	7.78578178	−7.78578178	0.000415602
6.52	2.177	4.739329	10.04095127	−10.04095127	4.35783E−05
7.1	2.757	7.601049	16.10391737	−16.10391737	1.01428E−07

Table 6
In_1: Processing in the clique C_6.

x_i^t	$(x_i^t - 3.878)$	$(x_i^t - 3.878)^2$	$\frac{(x_i^t-3.878)^2}{0.134}$	$-\frac{(x_i^t-3.878)^2}{0.134}$	$e^{-\frac{(x_i^t-3.878)^2}{0.134}}$
3.16	−0.718	0.515524	3.84719403	−3.84719403	0.021339531
5.92	2.042	4.169764	31.11764179	−31.11764179	3.06041E−14
6.26	2.382	5.673924	42.34271642	−42.34271642	4.08124E−19
6.52	2.642	6.980164	52.09077612	−52.09077612	2.38376E−23
7.1	3.222	10.381284	77.47226866	−77.47226866	2.26059E−34

Table 7
In_1: Processing in the clique C_7.

x_i^t	$(x_i^t - 3.916)$	$(x_i^t - 3.916)^2$	$\frac{(x_i^t-3.916)^2}{0.117}$	$-\frac{(x_i^t-3.916)^2}{0.117}$	$e^{-\frac{(x_i^t-3.916)^2}{0.117}}$
3.16	−0.756	0.571536	4.884923077	−4.884923077	0.007559705
5.92	2.004	4.016016	34.32492308	−34.32492308	1.23844E−15
6.26	2.344	5.494336	46.96013675	−46.96013675	4.03155E−21
6.52	2.604	6.780816	57.95569231	−57.95569231	6.76336E−26
7.1	3.184	10.137856	86.64834188	−86.64834188	2.33939E−38

Table 8
Mean and variance of cliques $C_1 - C_7$.

Mean, Variance	C_1 Ahorramas	C_2 Alcampo	C_3 Carrefour	C_4 Dia	C_5 Hipercor	C_6 Lidl	C_7 Mercadona
μ_i	3.988	3.884	3.851	3.858	4.343	3.878	3.916
σ_i^2	0.172	0.156	0.143	0.108	0.472	0.134	0.117

Table 9
Aggregated probability for In_1.

	$po[X^t = 3.16]$	$po[X^t = 5.92]$	$po[X^t = 6.26]$	$po[X^t = 6.52]$	$po[X^t = 7.10]$
$P_{C_1}[X^t = -]$	0.018574725	0.056878452	0.018974535	0.007270127	0.00058834
$P_{C_2}[X^t = -]$	0.034731697	2.88236E−12	1.9214E−16	4.52701E−20	1.60945E−29
$P_{C_3}[X^t = -]$	0.03547142	9.98216E−14	2.37298E−18	2.32045E−22	8.73309E−33
$P_{C_4}[X^t = -]$	0.01098584	7.98533E−18	6.29517E−24	3.19503E−29	5.42556E−43
$P_{C_5}[X^t = -]$	0.051559486	0.005149182	0.000415602	4.35783E−05	1.01428E−07
$P_{C_6}[X^t = -]$	0.021339531	3.06041E−14	4.08124E−19	2.38376E−23	2.26059E−34
$P_{C_7}[X^t = -]$	0.007559705	1.23844E−15	4.03155E−21	6.76336E−26	2.33939E−38
$\prod_i^7 P_{C_i}[X^t = -]$	2.09102E−12	2.55039E−74	3.7242E−101	1.7143E−124	2.4066E−185

Table 10
In_2: Processing in the clique C_1.

x_i^t	$(x_i^t - 3.988)$	$(x_i^t - 3.988)^2$	$\frac{(x_i^t-3.988)^2}{0.172}$	$-\frac{(x_i^t-3.988)^2}{0.172}$	$e^{-\frac{(x_i^t-3.988)^2}{0.172}}$
2.71	−1.278	1.633284	9.495837209	−9.495837209	7.51641E−05
3.78	−0.208	0.043264	0.251534884	−0.251534884	0.777606331
3.8	−0.188	0.035344	0.205488372	−0.205488372	0.814249563
3.86	−0.128	0.016384	0.095255814	−0.095255814	0.909140334
3.96	−0.028	0.000784	0.00455814	−0.00455814	0.995452233

Table 11
In_2: Processing in the clique C_2.

x_i^t	$(x_i^t - 3.884)$	$(x_i^t - 3.884)^2$	$\frac{(x_i^t-3.884)^2}{0.156}$	$-\frac{(x_i^t-3.884)^2}{0.156}$	$e^{-\frac{(x_i^t-3.884)^2}{0.156}}$
2.71	−1.174	1.378276	8.835102564	−8.835102564	0.000145534
3.78	−0.104	0.010816	0.069333333	−0.069333333	0.933015623
3.8	−0.084	0.007056	0.045230769	−0.045230769	0.955776892
3.86	−0.024	0.000576	0.003692308	−0.003692308	0.9963145
3.96	0.076	0.005776	0.037025641	−0.037025641	0.963651426

Eff$[In_1] = p \cdot (po[X^t = 3.16] + po[X^t = 5.92] + po[X^t = 6.26] + po[X^t = 6.52] + po[X^t = 7.10]) = 0.23 \cdot (2.09102\text{E}{-12} + 2.55039\text{E}{-74} + 3.7242\text{E}{-101} + 1.7143\text{E}{-124} + 2.4066\text{E}{-185}) = 0.23 \cdot 2.09102\text{E}{-12} =$ 4.80935E–13

Similarly, the computation of In_2 is supported by the information provided in Tables 10–16.

Finally, the required probability is computed (see Table 17).
Eff$[In_2] = p \cdot (po[X^t = 2.71] + po[X^t = 3.78] + po[X^t = 3.80] + po[X^t = 3.86] + po[X^t = 3.96]) = 0.57(3.24825\text{E}{-30} + 0.268808808 + 0.337851876 + 0.53631732 + 0.549630567 + 1.692608571) =$ 0.964786886 .

Table 12
In_2: Processing in the clique C_3.

x_i^t	$(x_i^t - 3.851)$	$(x_i^t - 3.851)^2$	$\frac{(x_i^t-3.851)^2}{0.143}$	$-\frac{(x_i^t-3.851)^2}{0.143}$	$e^{-\frac{(x_i^t-3.851)^2}{0.143}}$
2.71	−1.141	1.301881	9.104062937	−9.104062937	0.000111213
3.78	−0.071	0.005041	0.035251748	−0.035251748	0.965362357
3.8	−0.051	0.002601	0.018188811	−0.018188811	0.981975607
3.86	0.009	8.1E−05	0.000566434	−0.000566434	0.999433727
3.96	0.109	0.011881	0.083083916	−0.083083916	0.920273918

Table 13
In_2: Processing in the clique C_4.

x_i^t	$(x_i^t - 3.858)$	$(x_i^t - 3.858)^2$	$\frac{(x_i^t-3.858)^2}{0.108}$	$-\frac{(x_i^t-3.858)^2}{0.108}$	$e^{-\frac{(x_i^t-3.858)^2}{0.108}}$
2.71	−1.148	1.317904	12.20281481	−12.20281481	5.01632E−06
3.78	−0.078	0.006084	0.056333333	−0.056333333	0.945224009
3.8	−0.058	0.003364	0.031148148	−0.031148148	0.969331958
3.86	0.002	4E−06	3.7037E−05	−3.7037E−05	0.999962964
3.96	0.102	0.010404	0.096333333	−0.096333333	0.908161245

Table 14
In_2: Processing in the clique C_5.

x_i^t	$(x_i^t - 4.343)$	$(x_i^t - 4.343)^2$	$\frac{(x_i^t-4.343)^2}{0.472}$	$-\frac{(x_i^t-4.343)^2}{0.472}$	$e^{-\frac{(x_i^t-4.343)^2}{0.472}}$
2.71	−1.633	2.666689	5.649764831	−5.649764831	0.003518344
3.78	−0.563	0.316969	0.671544492	−0.671544492	0.510918858
3.8	−0.543	0.294849	0.624680085	−0.624680085	0.535432694
3.86	−0.483	0.233289	0.494256356	−0.494256356	0.61002438
3.96	−0.383	0.146689	0.31078178	−0.31078178	0.732873786

Table 15
In_2: Processing in the clique C_6.

x_i^t	$(x_i^t - 3.878)$	$(x_i^t - 3.878)^2$	$\frac{(x_i^t-3.878)^2}{0.134}$	$-\frac{(x_i^t-3.878)^2}{0.134}$	$e^{-\frac{(x_i^t-3.878)^2}{0.134}}$
2.71	−1.168	1.364224	10.18077612	−10.18077612	3.78918E−05
3.78	−0.098	0.009604	0.071671642	−0.071671642	0.930836493
3.8	−0.078	0.006084	0.045402985	−0.045402985	0.955612307
3.86	−0.018	0.000324	0.00241791	−0.00241791	0.99758501
3.96	0.082	0.006724	0.050179104	−0.050179104	0.95105907

Table 16
In_2: Processing in the clique C_7.

x_i^t	$(x_i^t - 3.916)$	$(x_i^t - 3.916)^2$	$\frac{(x_i^t-3.916)^2}{0.117}$	$-\frac{(x_i^t-3.916)^2}{0.117}$	$e^{-\frac{(x_i^t-3.916)^2}{0.117}}$
2.71	−1.206	1.454436	12.43107692	−12.43107692	3.99256E−06
3.78	−0.136	0.018496	0.15808547	−0.15808547	0.853776806
3.8	−0.116	0.013456	0.115008547	−0.115008547	0.891358525
3.86	−0.056	0.003136	0.026803419	−0.026803419	0.973552605
3.96	0.044	0.001936	0.016547009	−0.016547009	0.983589141

Table 17
Aggregated probability for In_2.

	$po[X^t = 2.71]$	$po[X^t = 3.78]$	$po[X^t = 3.80]$	$po[X^t = 3.86]$	$po[X^t = 3.96]$
$P_{C_1}[X^t = -]$	7.51641E−05	0.777606331	0.814249563	0.909140334	0.995452233
$P_{C_2}[X^t = -]$	0.000145534	0.933015623	0.955776892	0.9963145	0.963651426
$P_{C_3}[X^t = -]$	0.000111213	0.965362357	0.981975607	0.999433727	0.920273918
$P_{C_4}[X^t = -]$	5.01632E−06	0.945224009	0.969331958	0.999962964	0.908161245
$P_{C_5}[X^t = -]$	0.003518344	0.510918858	0.535432694	0.61002438	0.732873786
$P_{C_6}[X^t = -]$	3.78918E−05	0.930836493	0.955612307	0.99758501	0.95105907
$P_{C_7}[X^t = -]$	3.99256E−06	0.853776806	0.891358525	0.973552605	0.983589141
$\prod_i^7 P_{C_i}[X^t = -]$	3.24825E−30	0.268808808	0.337851876	0.53631732	0.549630567

Hence, as expected (since the inputs of In_1 reflect prices with unusual fluctuations and therefore with higher deviation from the mean) $\text{Eff}[In_2] > \text{Eff}[In_1]$.

8. Conclusions

This paper deals with the selection of optimal training sets (those that have a higher capacity as estimators) in Recurrent Neural Networks under prediction tasks (or pattern recognition with time series as inputs), although this may also apply to other data-driven models regulated by dynamic systems. Our objective is to fill the existing gap of clear guidelines to follow for selecting optimal training sets in a general context.

We design here a novel methodology to select optimal training data sets that can be used in any context. The key idea, which underpins the design of the mathematical structure that supports the selection, is a binary relation that gives preference to inputs with higher estimator abilities. A second novelty of our approach is to use dynamic tools that have not been used previously for this purposes: dynamic Markov Networks, which are widely regarded as generative models, successfully compute the prior probabilities involved in the formula for calculating the degree of efficiency of the training set (Theorem 6.3), derived from application of the Von Neumann-Morgenstern Theorem 2.1. It is precisely the VMN theorem the instrument that confers discriminative capacity to the MNs: in this work we show that the preference relation that we define between inputs of a training set (inputs with higher learning capacities are preferred in the sense that the error function takes lower values) fulfils the necessary hypotheses to derive the existence of a simple formula for the calculation of the utility (efficiency) of a training set.

The simplicity of this calculation allows it to be carried out in parallel with the learning process without adding computational cost. Thus the optimal sets are selected as the learning process evolves, therefore the data noise gradually disappears which decreases the likelihood of overfitting occurring.

Declarations of interest: none.

Funding

Financial support from the Spanish Ministry of Universities. "Disruptive group decision making systems in fuzzy context: Applications in smart energy and people analytics" (PID2019-103880RB-I00). Main Investigator: Enrique Herrera Viedma, and Junta de Andalucía. "Excellence Groups" (P12.SEJ.2463) and Junta de Andalucía (TIC186) are gratefully acknowledged. Research partially supported by the "Maria de Maeztu" Excellence Unit IMAG, reference CEX2020-001105-M, funded by MCIN/AEI/10.13039/501100011033/.

References

Chen, A.N. (2006). Robust optimization for performance tuning of modern database systems. *European Journal of Operational Research*, 171, 412–429. https://doi.org/10.1016/j.ejor.2011.03.043.

Chou, P., Chuang, H.H., Chou, Y., Liang, T. (2022). Predictive analytics for customer repurchase: Interdisciplinary integration of buy till you die modeling and machine learning. *European Journal of Operational Research*, 296, 635–651. https://doi.org/10.1016/j.ejor.2021.04.021.

Delbaen, F., Drapeau, S., Kupper, M. (2011). A von neumann morgenstern representation result without weak continuity assumption. *Journal of Mathematical Economics*, 47, 401–408. https://doi.org/10.1016/j.jmateco.2011.04.002.

Dynkin, E. (1984). Gaussian and nongaussian random fields associated with markov processes. *Journal of Functional Analysis*, 55, 344–376. https://doi.org/10.1016/0022-1236(84)90004-1.

Fernandez Anitzine, I., Romo Argota, J.A., Fontan, F.P. (2012). Influence of training set selection in artificial neural network based propagation path loss predictions. *International Journal of Antennas and Propagation*, *2012*. https://doi.org/10.1155/2012/351487.

García Cabello, J. (2021). A novel intelligent system for securing cash levels using markov random fields. *International Journal of Intelligent Systems*, 36, 4468–4490. https://doi.org/10.1002/int.22467.

García Cabello, J. (2023). *Improved deep neural network performance under dynamic programming mode*. Preprint. https://doi.org/10.2139/ssrn.4410415.

Gordon, J., Hernandez-Lobato, J.M. (2020). Combining deep generative and discriminative models for bayesian semi-supervised learning. *Pattern Recognition*, 100, 107156. https://doi.org/10.1016/j.patcog.2019.107156.

Higham, C.F., Higham, D.J. (2019). Deep learning: an introduction for applied mathematicians. *Siam Review*, 61, 860–891. https://doi.org/10.1137/18M1165748.

Jiang, L., Liao, H. (2022). Bounded rational reciprocal preference relation for decision making. *Informatica*, 33, 731–748. https://doi.org/10.15388/23-INFOR511.

Kim, K. (2006). Artificial neural networks with evolutionary instance selection for financial forecasting. *Expert Systems with Applications*, 30, 519–526. https://doi.org/10.1016/j.eswa.2005.10.007.

Machina, M.J. (1982). Expected utility analysis without the independence axiom. *Econometrica: Journal of the Econometric Society*, 50(2), 277–323. https://doi.org/10.2307/1912631.

Mirjalili, S., Hashim, S.Z.M., Sardroudi, H.M. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, 218, 11125–11137. https://doi.org/10.1016/j.amc.2012.04.069.

Nalepa, J., Myller, M., Piechaczek, S., Hrynczenko, K., Kawulok, M. (2018). Genetic selection of training sets for (not only) artificial neural networks. In: Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., Kostrzewa, D. (Eds.), *Beyond Databases, Architectures and Structures. Facing the Challenges of Data Proliferation and Growing Variety, BDAS 2018*, Communications in Computer and Information Science, Vol. 928. Springer, Cham. https://doi.org/10.1007/978-3-319-99987-6_15.

Pollak, R.A. (1967). Additive von neumann-morgenstern utility functions. *Econometrica, Journal of the Econometric Society*, 353–4, 485–494. https://doi.org/10.2307/1905650.

Reeves, C.R., Bush, D.R. (2001). Using genetic algorithms for training data selection in RBF networks. In: Liu, H., Motoda, H. (Eds.), *Instance Selection and Construction for Data Mining, The Springer International Se-*

ries in Engineering and Computer Science, 608. Springer, Boston, MA, pp. 339–356. https://doi.org/10.1007/978-1-4757-3359-4_19.

Reeves, C.R., Taylor, S.J. (1998). Selection of training data for neural networks by a genetic algorithm. In: *Parallel Problem Solving from Nature-PPSN V: 5th International Conference Amsterdam, The Netherlands September 27–30, 1998 Proceedings 5*. Springer, pp. 633–642. https://doi.org/10.1007/BFb0056905.

Smale, S., Rosasco, L., Bouvrie, J., Caponnetto, A., Poggio, T. (2010). Mathematics of the neural response. *Foundations of Computational Mathematics*, 10, 67–91. https://doi.org/10.1007/s10208-009-9049-1.

Van Den Brink, R., Rusinowska, A. (2022). The degree measure as utility function over positions in graphs and digraphs. *European Journal of Operational Research*, 299, 1033–1044. https://doi.org/10.1016/j.ejor.2021.10.017.

Wang, L., Zhou, Y., Li, R., Ding, L. (2022). A fusion of a deep neural network and a hidden markov model to recognize the multiclass abnormal behavior of elderly people. *Knowledge-Based Systems*, 252, 109351. https://doi.org/10.1016/j.knosys.2022.109351.

Wong, D.F., Lu, Y., Chao, L.S. (2016). Bilingual recursive neural network based data selection for statistical machine translation. *Knowledge-Based Systems*, 108, 15–24. https://doi.org/10.1016/j.knosys.2016.05.003.

Yang, J., Qiu, W. (2005). A measure of risk and a decision-making model based on expected utility and entropy. *European Journal of Operational Research*, 164, 792–799. https://doi.org/10.1016/j.ejor.2004.01.031.

Zapf, F., Wallek, T. (2021). Comparison of data selection methods for modeling chemical processes with artificial neural networks. *Applied Soft Computing*, 113, 107938. https://doi.org/10.1016/j.asoc.2021.107938.

Zhang, H., Wang, Z., Liu, D. (2014). A comprehensive review of stability analysis of continuous-time recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25, 1229–1262. https://doi.org/10.1109/TNNLS.2014.2317880.

Zhang, L., Suganthan, P.N. (2016). A survey of randomized algorithms for training neural networks. *Information Sciences*, 364, 146–155. https://doi.org/10.1016/j.ins.2016.01.039.

J. García Cabello was born in Andalusia (Spain). She received the PhD degree in pure and applied mathematics from the University of Granada where she has been teaching since 1990. Prior to getting to know at the world of applied mathematics, she developed a successful career in pure algebra (known as JG Cabello). Today, she is a fully tenured professor and a full researcher at the Applied Mathematics Department of the University of Granada (Spain), where she teaches undergraduate, MBA and Executive MBA courses and conducts seminars on a wide range of mathematical business-related topics.

She is a full researcher at the Andalusian Research Institute in Data Science and Computational Intelligence. Her current research interests include the application of applied mathematics to the resolution of real problems, decision making, theoretical computer science and operational research. To this regard, her mathematical baggage (from pure algebra to applied mathematics) makes Dr. García Cabello's research characterized by using a wide range of mathematical tools, from stochastic processes to dynamic systems. Dr. Julia García Cabello is also a regular reviewer of journals *Applied Mathematics and Intelligent* and *Information Systems*.