Toward Big Data Manipulation for Grape Harvest Time Prediction by Intervals' Numbers Techniques

V. G. Kaburlasos[1*], E. Vrochidou[1], C. Lytridis[1], G. A. Papakostas[1], T. Pachidis[1], M. Manios[1], S. Mamalis[2], T. Merou[2], S. Koundouras[2], S. Theocharis[2], G. Siavalas[3], C. Sgouros[3], P. Kyriakidis[4]

[1]Human-Machines Interaction (HUMAIN) Lab, International Hellenic University (IHU), Kavala, 65404 Greece
[2]School of Agricultural Biotechnology and Oenology, International Hellenic University (IHU), Drama, 66100 Greece
[3]Euroaction, Vasilissis Olgas 281, Thessaloniki, 55133 Greece
[4]Ktima Pavlidis, Kokinogia, Drama, 66100 Greece
[*]Corresponding author email: vgkabs@teiemt.gr

***Abstract*—The automation of agricultural production calls for accurate prediction of the harvest time. Our interest in particular here is in grape harvest time. Nevertheless, the latter prediction is not trivial also due to the scale of data involved. We propose a novel neural network architecture that processes whole histograms induced from digital images. A histogram is represented by an Intervals' Number (IN); hence, all-order data statistics are represented. In conclusion, the proposed "IN Neural Network", or INNN for short, emerges with the capacity of predicting an IN from past INs. We demonstrate a "proof-of-concept", preliminary application on a time series of digital images of grapes taken during their growth to maturity. Compared to a conventional Back Propagation Neural Network (BPNN), the results by INNN are superior not only in terms of prediction accuracy but also because the BPNN predicts only first-order data statistics, whereas the INNN predicts all-order data statistics.**

***Index Terms*—Autonomous Robot, Big Data, Dexterous Farming, Grape Harvest, Prediction Model, Neural Computing**

I. Introduction

Automation of agricultural production is required in order to meet an ever increasing demand for food, worldwide. In this context, there is a number of technological challenges. For instance, the scale of the data needed to make farm machines intelligent, toward automating agricultural production, is daunting [1]. Our interest is in developing a (semi-)autonomous robot grape harvester for *dexterous farming* tasks; the latter are defined here as farming tasks that call for manual dexterity such as harvesting, pruning, etc.

In the above context, a critical decision regards the accurate prediction of harvest time because the right mass of both human labor and equipment should be timely engaged. Nevertheless, an automation of the aforementioned decision is difficult also due to the scale of data available in agriculture.

Based on recent advances in machine vision in agriculture, there are several reviews that focus on this subject [2]–[5]. In particular, note that in various agricultural applications, computer vision enables both visual inspection and measurements because it is objective, consistent, rapid and inexpensive [6], [7]. Color sorting is reported as an efficient, non-invasive method for monitoring fruit ripening, including grape ripening [8]. For instance, regarding red grapes, the color changes from green to either purple or black during ripening. In particular, color changes are related to chemical composition changes, especially regarding phenolics, the latter are important compounds of grapes related to the intensity as well as the stability of red wine color [9]. Therefore, the potential of computer vision to determine the phenolic maturity level of grapes has been considered toward establishing an objective ripening index by techniques of image analysis [8]. Various methods have been proposed for estimating the maturity level of red grapes [8], [9]. However, the latter methods cannot predict the time grapes reach full maturity. It turns out that the prediction of grape harvest time in non-trivial because it depends on chaotic factors such as the weather.

Our proposal here is to pursue prediction of grape harvest time based on histograms derived from digital images. In particular, a histogram is represented by an *Intervals' Number*, or *IN* for short. Recall that an IN is an established mathematical object [10]–[14] that may represent either a fuzzy interval or a distribution of numbers. INs have already been applied to a number of neural/fuzzy systems [10], [11], [14]–[16]. An N-tuple of INs here represents the maturity of grape. A time series of N-tuples of INs is considered. Our objective is to predict future N-tuples of INs from past N-tuples of INs. The aforementioned employment of INs is a novelty of this work that may engage all-order statistics of big data, using orders of magnitude fewer numbers, toward predicting all-order statistics of big data.

The techniques proposed here fall within the *Lattice Computing* (LC) information processing paradigm, which has been defined as "an evolving collection of tools and methodologies that process lattice ordered data including logic values, numbers, sets, symbols, graphs, etc" [17], [18].

The layout of this paper is as follows. Section II outlines the mathematical background. Section III presents the proposed, IN-based neural network. Section IV presents preliminary application results. Finally, section V concludes by summarizing our contribution as well as by discussing ongoing work.

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T1EDK-00300).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

II. The Mathematical Background

Recall that a *lattice* is a partially ordered set $(\mathbb{L}, \sqsubseteq)$ with the additional property that any two elements $x, y \in \mathbb{L}$ have both a *greatest lower bound*, namely *infimum*, denoted by $x \sqcap y$ and a *least upper bound*, namely *supremum*, denoted by $x \sqcup y$. If in a lattice $(\mathbb{L}, \sqsubseteq)$ every pair (x, y) satisfies either $x \sqsubseteq y$ or $x \sqsupset y$ then we say that lattice $(\mathbb{L}, \sqsubseteq)$ is *totally-ordered*. A lattice $(\mathbb{L}, \sqsubseteq)$ is called *complete* if and only if each of its subsets X has both an infimum and a supremum in $\mathbb{L}$ (therefore, taking $X = \mathbb{L}$, it follows that a complete lattice has both a *least* element and a *greatest* element) [19].

A. A Hierarchy of Lattices

This section presents a three level hierarchy of lattices resulting in the lattice of Intervals' Numbers (INs, for short). The basic theory has been presented in [11], [14], [19], [20].

a) Level-0; the lattice $(\overline{\mathbb{R}}, \leq)$ of real numbers: Consider the totally-ordered, complete lattice $(\overline{\mathbb{R}}, \leq)$, where $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ with least and greatest elements denoted by $-\infty$ and ∞, respectively. The corresponding infimum and supremum operations are denoted by $\wedge$ and $\vee$, respectively.

b) Level-1; the lattice $(\mathbb{I}_1, \subseteq)$ of Type-1 intervals: Given $a_1, a_2 \in \overline{\mathbb{R}}$ such that $a_1 \leq a_2$, a (Type-1) *interval* $A = [a_1, a_2]$ is defined as $[a_1, a_2] = \{x : x \in \overline{\mathbb{R}} \text{ and } a_1 \leq x \leq a_2\}$. We denote the collection of (Type-1) intervals in $\overline{\mathbb{R}}$ (including the empty interval) by $\mathbb{I}_1(\overline{\mathbb{R}})$, or simply by $\mathbb{I}_1$ – The empty set ($\emptyset$) is also considered an interval, and it is represented by $[\infty, -\infty]$. It turns out that $(\mathbb{I}_1, \subseteq)$ is a complete lattice, where $\subseteq$ is the set theoretic inclusion.

Given intervals $A = [a_1, a_2]$ and $B = [b_1, b_2]$ in $\mathbb{I}_1$, their infimum ($\cap$) and supremum ($\dot{\cup}$), respectively, are given by

$$A \cap B = \begin{cases} [a_1 \vee b_1, a_2 \wedge b_2], & \text{iff } a_1 \vee b_1 \leq a_2 \wedge b_2 \\ \emptyset, & \text{otherwise} \end{cases} \quad (1)$$

$$A \,\dot{\cup}\, B = [a_1 \wedge b_1, a_2 \vee b_2]. \quad (2)$$

c) Level-2; the lattice $(\mathbb{F}_1, \preceq)$ of Type-1 Intervals' Numbers (INs): A (Type-1) *Intervals' Number*, or *IN* for short, is defined as a function $F : [0, 1] \to \mathbb{I}_1$ which satisfies

$$h_1 \geq h_2 \Rightarrow F_{h_1} \subseteq F_{h_2}, \text{ and}$$
$$\forall X \subseteq [0, 1] : \bigcap_{h \in X} F_h = F_{\vee X}.$$

We denote the family of Type-1 INs by $\mathbb{F}_1$. It turns out that $(\mathbb{F}_1, \preceq)$ is a complete lattice, where $\preceq$ is defined as follows:

$$F \preceq G \Leftrightarrow (\forall h \in [0, 1] : F_h \subseteq G_h). \quad (3)$$

It is known that an IN F represents a (convex) fuzzy interval $F(x)$ defined as $\forall x : F(x) = \sup\{h : x \in F_h\}$; in particular, F_h denotes the well known "α-cut" of $F(x)$ for $h = \alpha$. Furthermore, there is an *order isomorphism* between $(\mathbb{F}_1, \preceq)$ and the lattice of (convex) fuzzy intervals.

Given INs F and G in $\mathbb{F}_1$, their infimum ($\curlywedge$) and supremum ($\dot{\curlyvee}$), respectively, are given $\forall h \in [0, 1]$ by

$$(F \curlywedge G)_h = F_h \cap G_h \text{ and } (F \,\dot{\curlyvee}\, G)_h = F_h \,\dot{\cup}\, G_h \quad (4)$$

B. Algebraic Operations and Metric Distances in $\mathbb{F}_1$

This section presents algebraic operations in the set $\mathbb{I}_1$ of (Type-1) intervals and then it extends them to the set $\mathbb{F}_1$ of (Type-1) INs. Likewise, it presents metric distances.

1) Algebraic operations: Given intervals $[a, b], [c, d] \neq \emptyset$, their *addition* is defined as $[a, b] + [c, d] = [a + c, b + d]$; moreover, the *multiplication* of an interval $[a, b]$ by a (non-negative) number $\lambda \in \mathbb{R}$ is defined as $\lambda[a, b] = [\lambda a, \lambda b]$ [15]. Furthermore, given INs F and G, the corresponding *addition* and *multiplication* (by a non-negative number $\lambda \in \mathbb{R}$) are defined as $(F + G)_h = F_h + G_h$ and $(\lambda F)_h = \lambda F_h$, respectively, $\forall h \in [0, 1]$.

2) Metric distances: Recall that a *metric* distance in a set S is a function $d : S \times S \to \mathbb{R}_0^+$ which satisfies the three conditions: 1) $d(x, y) = 0 \Leftrightarrow x = y$, 2) $d(x, y) = d(y, x)$, and 3) $d(x, z) \leq d(x, y) + d(y, z)$.

A metric distance can be introduced in a general lattice $(\mathbb{L}, \sqsubseteq)$ by a *positive valuation* (real) function $v : \mathbb{L} \to \mathbb{R}$ which, by definition, satisfies both $v(x) + v(y) = v(x \sqcap y) + v(x \sqcup y)$ and $x < y \Rightarrow v(x) < v(y)$.

The absence of a positive valuation function in the lattice $(\mathbb{I}_1, \subseteq)$ is treated by considering the lattice $(\overline{\mathbb{R}} \times \overline{\mathbb{R}}, \geq \times \leq)$ of *generalized intervals*. Based on two functions $v : \overline{\mathbb{R}} \to [0, +\infty)$ and $\theta : \overline{\mathbb{R}} \to \overline{\mathbb{R}}$ that satisfy the following properties

- **A1** The function $v : \overline{\mathbb{R}} \to [0, +\infty)$ is strictly increasing such that both $v(-\infty) = 0$ and $v(\infty) < +\infty$, and
- **A2** The function $\theta : \overline{\mathbb{R}} \to \overline{\mathbb{R}}$ is strictly decreasing such that both $\theta(-\infty) = \infty$ and $\theta(\infty) = -\infty$

there follows a positive valuation function $v(a, b) = v(\theta(a)) + v(b)$ in lattice $(\overline{\mathbb{R}} \times \overline{\mathbb{R}}, \geq \times \leq)$; hence, a metric distance $d_{\overline{\mathbb{R}} \times \overline{\mathbb{R}}} : (\overline{\mathbb{R}} \times \overline{\mathbb{R}}) \times (\overline{\mathbb{R}} \times \overline{\mathbb{R}}) \to \mathbb{R}_0^+$ is given by

$$d_{\overline{\mathbb{R}} \times \overline{\mathbb{R}}}([a, b], [c, d]) = v(\theta(a \wedge c)) - v(\theta(a \vee c)) + v(b \vee d) - v(b \wedge d) \quad (5)$$

Since lattice $(\mathbb{I}_1, \subseteq)$ is embedded in lattice $(\overline{\mathbb{R}} \times \overline{\mathbb{R}}, \geq \times \leq)$, it follows that (5) is a metric distance in $(\mathbb{I}_1, \subseteq)$. We point out that given $v(-x) = -v(x)$ and $\theta(x) = -x$, there follows

$$d_{\overline{\mathbb{R}} \times \overline{\mathbb{R}}}([a, a], [b, b]) = 2(v(a \vee b) - v(a \wedge b)) \quad (6)$$

For $v(x) = x$ it follows $d_{\overline{\mathbb{R}} \times \overline{\mathbb{R}}}([a, a], [b, b]) = 2|a - b|$.

The metric $d_{\overline{\mathbb{R}} \times \overline{\mathbb{R}}}(., .)$ is extended to $(\mathbb{F}_1, \preceq)$ as follows.

$$d_{\mathbb{F}_1}(F, G) = \int_0^1 d_{\overline{\mathbb{R}} \times \overline{\mathbb{R}}}(F_h, G_h) dh \quad (7)$$

C. Explanations /Interpretations of Practical Significance

Various authors have studied comparatively probability and possibility distributions [21], [22]. Note that, first, a possibilistic interpretation was proposed for an IN [12], [23] followed by a probabilistic interpretation [24]. In conclusion, an IN was established as a mathematical object that may be interpreted either probabilistically or possibilistically [10]. Note that different authors have considered distance functions between fuzzy numbers [25]. In the same direction this work promotes a *parametric, metric* distance function $d_{\mathbb{F}_1}(., .)$.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

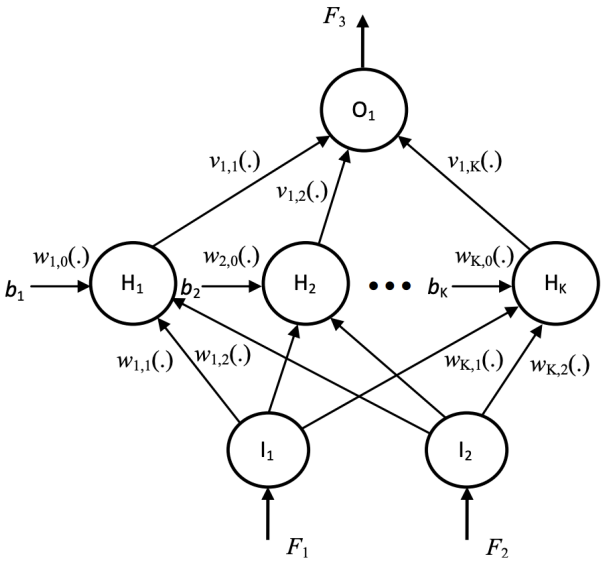


Fig. 1. The 3-layer $2 \times K \times 1$ feed-forward neural architecture in this figure processes Intervals' Numbers (INs) all along, where an IN represents a distribution of image samples regarding the maturity of a grape. In general, the weights $w_{i,j}(.), i \in \{1, \dots K\}, j \in \{1, 2\}$ and $v_{1,j}(.), j \in \{1, \dots K\}$ are strictly increasing functions. The shown architecture is trained to learn a difference equation in order to predict one future IN F_3 from two past INs F_1 and F_2 toward decision-support regarding grape harvest time prediction.

III. IN-based Neural Network (INNN)

The proposed IN-based Neural Network, or INNN for short, is shown in Fig.1. INNN is a 3-layer feed-forward neural network as an extension of a conventional feed-forward neural network from the Euclidean space $\mathbb{R}^N$ to the space $\mathbb{F}_1^N$ of INs.

a) INNN Training: The training phase of INNN, described by Algorithm 1, is based on a genetic algorithm.

Algorithm 1 INNN Training Phase

1: Randomly select a population of parameter sets individuals, where a parameter set consists of (a) a set of weights for the hidden layer as well as for the output layer, (b) parameters for the activation function of each neuron, and (c) a set of biases for each neuron;
2: **for** g generations **do**
3: Evaluate individuals by calculating the distance between the INNN computed output IN estimate and the real output IN for each parameter set;
4: Apply genetic operators;
5: **end for**

b) INNN Testing: The testing phase is straightforward, therefore it is omitted.

An INNN bias is a trivial IN, that is a vector of 32 equal real numbers. At the end of the training phase, optimal parameters are estimated to be used in the testing phase.

We remark that the architecture in Fig.1 can be interpreted as a fuzzy inference system [12].

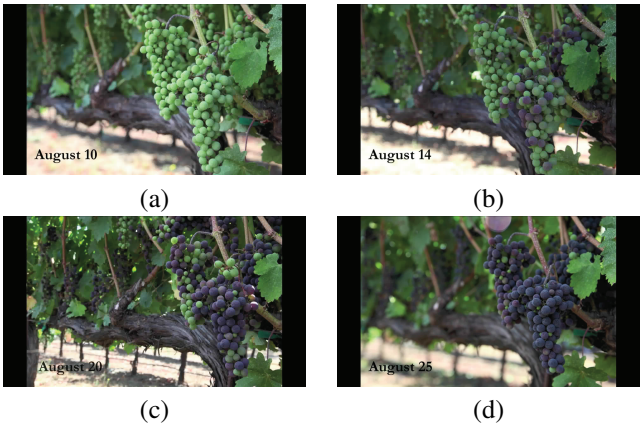


Fig. 2. Pictures of grapes taken from a fixed distance, at the same time of the day on (a) August 10, (b) August 14, (c) August 20, and (d) August 25. The color of grapes progressively turns from green to black as the grapes ripen.

IV. Preliminary Application Results

Data acquisition is critical for developing an effective, grape maturity prediction model. In this preliminary study, we carried out data acquisition as explained next.

A. Data acquisition

In this pilot study, data (i.e., images) acquisition was based on a veraison to harvest time lapse video downloaded from the web [26]. The video presents the ripeness stages of a variety of red grapes (Cabernet Sauvignon) demonstrating their change in color from August 10 to August 25, in 13 successive days from a fixed view angle. In all, 13 video snapshots were taken based on illumination criteria so that all the selected images were under similar lighting conditions (Fig.2). Note that both the illumination intensity and the angle between the camera and light source affect the histogram shape, therefore they need to be constant. The images had a resolution of 1920×1080 pixels and 24-bit color depth stored in .tiff format.

A straightforward approach to describe intensity variations of grapes is a color histogram. Based on grape variety, the suitable color channel of a RGB image was used to calculate intensity distributions as explained below. More sophisticated histogram-based representations [27], e.g. the Local Binary Patterns (LBP), can also be used to encode the local intensity variations toward describing the micro-structure of grape surface. Next, a histogram was represented by an IN.

B. Image Preprocessing

Following image acquisition, each image was manually cropped using Matlab segmentation algorithms [28], [29] toward isolating the region of interest on an image, i.e. the grapes, and exclude its background (Fig.3). Often, automated segmentation algorithms are used because manual segmentation takes time; nevertheless, manual segmentation is more precise. Since in our study, only a small number of frames was involved, we carried out segmentation manually. We remark that the proposed INNN results do not depend on the segmentation algorithm.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

(a) (b) (c)

Fig. 3. Manual image segmentation. (a) Original image of a grape. (b) A mask defined manually by a user. (c) The segmented image of interest includes only the grape in black background.

Color histograms were induced from all segmented images. As expected, the data showed that grape color was changing gradually from green to red/black; in particular, the green channel histograms have demonstrated the greatest variations during the ripening process as shown in Fig.4. Therefore, in this pilot study, the green color channel histograms were selected and post-processed.

C. IN Induction

Green channel histograms were represented by INs by algorithm *distIN* [17]. In conclusion, for the 13 selected snapshots of grape ripeness stages, the corresponding INs were induced.

A main advantage of an IN is its capacity to represent *all order* data statistics using fairly few, namely L, numbers that define L intervals [17]. In particular, note that a histogram corresponds to a *probability density function* (pdf), whereas an IN corresponds to a *probability distribution function* (PDF). Fig.5 illustrates both the membership-function-representation and the interval-representation of INs for $L = 32$ levels equally spaced over the interval [0, 1], for two images corresponding to different ripeness levels.

D. Experiments and Results

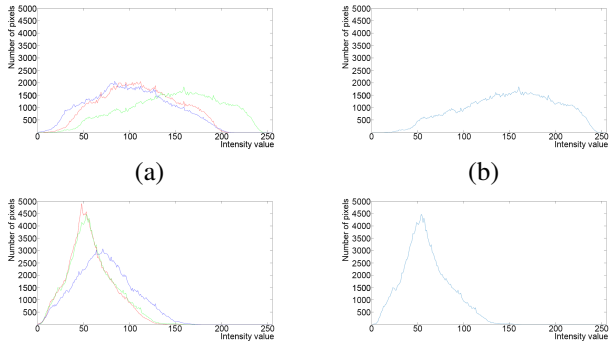
We employed the neural architecture of Fig.1 with two input INs corresponding to the green color histogram of two successive samples. The corresponding output IN was the one induced from the green color histogram of the very following day. A number $K = 15$ of hidden layer neurons was selected by trial-and-error.

A weight either $v_{1,j}(.), j \in \{1, \dots K\}$ or $w_{i,j}(.), i \in \{1, \dots K\}, j \in \{1, 2\}$ was a linear function ax, where $a \in [1, 10]$. Moreover, the activation function of a hidden/output layer neuron was a sigmoid function

$$\frac{A}{1 + e^{-\lambda(x-m)}} \tag{8}$$

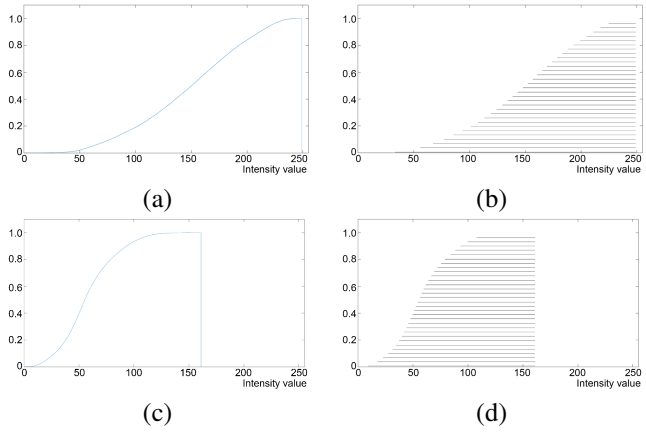
where $A \in [0, 255]$, $\lambda \in [0, 10]$, $m \in [-50, 300]$. The proposed biases used were "trivial INs", i.e. vectors of 32 equal numbers selected randomly in the range [-100, 100].

When a pair (F_1, F_2) of INs was applied to the input of the neural network of Fig.1, an output IN $\widehat{F}_3$ was computed as an estimate of the real (i.e., measured) output IN F_3. The distance $d_{\mathbb{F}_1}(\widehat{F}_3, F_3)$ was used here as the cost function to



(a) (b) (c) (d)

Fig. 4. Histograms calculated from digital images of grapes indicating the number of pixels per intensity level $0 - 255$. (a) RGB histograms calculated from an image taken on the first day. (b) The G histogram of the first day is displayed alone. (c) RGB histograms calculated from an image taken on the last day. Note that the histograms have moved to the left, moreover they have become sharper. (d) The G histogram of the last day is displayed alone.



(a) (b) (c) (d)

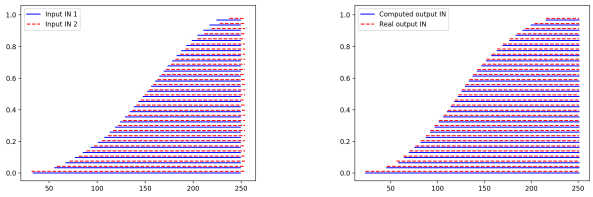
Fig. 5. Intervals' Numbers (INs) computed from the G histograms in Fig.4. (a) "Membership-function-representation" of the IN induced in Fig.4(b). (b) "Interval-representation" of the IN in Fig.5(a). (c) "Membership-function-representation" of the IN induced in Fig.4(d). (d) "Interval-representation" of the IN in Fig.5(c).

be minimized by evolutionary optimization techniques toward calculating optimal neural network weights.

The neural network of Fig.1 was trained in three different experiments E1, E2 and E3, respectively. More specifically, in experiment E1, only the first data sample "$(F_1, F_2) \rightarrow F_3$" was used for training, whereas all the remaining data samples were used for testing; in experiment E2, every other data sample was used for training, whereas all the remaining data samples were used for testing; finally, in experiment E3, the first half of the data samples were used for training, whereas all the remaining data samples were used for testing.

Tables I, II and III show the corresponding training/testing errors, where an error was computed by equations (6) and (7) as the distance between a measured IN and a predicted IN. We point out that due to (6) only half of the errors in Tables I, II and III should be considered as detailed in [23]. When all the data where used for training the average error was 10.39.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) (b)

Fig. 6. (a) Two inputs to the neural network of Fig.1, INs F_1 and F_2 in their "interval-representation" with solid and dashed lines, respectively. (b) The corresponding, computed output IN $\widehat{F_3}$ versus the real output IN F_3 in their "interval-representation" with solid and dashed lines, respectively.

TABLE I
TRAINING/TESTING DISTANCE ERROR AVERAGE IN 10 RANDOM TRIALS. ONLY THE FIRST DATA SAMPLE WAS USED FOR TRAINING

Data samples	***Training error***	***Testing error***
$(F_1, F_2) \rightarrow F_3$	0.20	
$(F_2, F_3) \rightarrow F_4$		49.39
$(F_3, F_4) \rightarrow F_5$		59.04
$(F_4, F_5) \rightarrow F_6$		27.74
$(F_5, F_6) \rightarrow F_7$		9.34
$(F_6, F_7) \rightarrow F_8$		14.66
$(F_7, F_8) \rightarrow F_9$		4.87
$(F_8, F_9) \rightarrow F_{10}$		7.75
$(F_9, F_{10}) \rightarrow F_{11}$		11.18
$(F_{10}, F_{11}) \rightarrow F_{12}$		27.69
$(F_{11}, F_{12}) \rightarrow F_{13}$		17.33
Grand average	**0.20**	**22.90**

We repeated the aforementioned experiments using a conventional Back Propagation Neural Network (BPNN) [30], [31] having the same architecture as the INNN shown in Fig.1. The latter was achieved by replacing an IN by a single number, namely the mean value of the IN's corresponding PDF. Table IV shows the computed numerical data. First, we trained the BPNN using every other numerical data sample for training; the corresponding testing data (average difference) error was 13.93 to be compared with the 21.08 error of Table II. Second, we trained the BPNN using the first half data samples for training; the corresponding testing data (average difference) error was 18.45 to be compared with the 13.68 error of Table III. Third, we trained the BPNN using all the data for training; the corresponding testing data (average difference) error was 7.53 to be compared with the aforementioned 10.39 error.

E. Discussion

The employment of INNN resulted in a clearly smaller prediction error compared to the error of a BPNN. Recall that due to equation (6), only half of the errors shown in Tables I, II and III should be considered. Moreover, by predicting an IN, the proposed INNN can predict all-order data statistics.

Since an IN represents a distribution, it follows that an IN can represent *all order* data statistics [24] using only L real numbers, that is an advantage of using INs in big data applications. In addition, note that no *ad hoc* feature extraction was employed here. The latter is a remarkable advantage of

TABLE II
TRAINING/TESTING DISTANCE ERROR AVERAGE IN 10 RANDOM TRIALS. EVERY OTHER DATA SAMPLE WAS USED FOR TRAINING

Data samples	***Training error***	***Testing error***
$(F_1, F_2) \rightarrow F_3$	6.82	
$(F_2, F_3) \rightarrow F_4$		51.46
$(F_3, F_4) \rightarrow F_5$	9.29	
$(F_4, F_5) \rightarrow F_6$		12.75
$(F_5, F_6) \rightarrow F_7$	1.66	
$(F_6, F_7) \rightarrow F_8$		9.90
$(F_7, F_8) \rightarrow F_9$	8.77	
$(F_8, F_9) \rightarrow F_{10}$		10.77
$(F_9, F_{10}) \rightarrow F_{11}$	5.74	
$(F_{10}, F_{11}) \rightarrow F_{12}$		20.53
$(F_{11}, F_{12}) \rightarrow F_{13}$	6.28	
Grand average	**6.43**	**21.08**

TABLE III
TRAINING/TESTING DISTANCE ERROR AVERAGE IN 10 RANDOM TRIALS. THE FIRST HALF DATA SAMPLES WERE USED FOR TRAINING

Data samples	***Training error***	***Testing error***
$(F_1, F_2) \rightarrow F_3$	3.66	
$(F_2, F_3) \rightarrow F_4$	11.69	
$(F_3, F_4) \rightarrow F_5$	23.74	
$(F_4, F_5) \rightarrow F_6$	12.43	
$(F_5, F_6) \rightarrow F_7$	3.69	
$(F_6, F_7) \rightarrow F_8$	6.62	
$(F_7, F_8) \rightarrow F_9$		6.30
$(F_8, F_9) \rightarrow F_{10}$		8.37
$(F_9, F_{10}) \rightarrow F_{11}$		10.30
$(F_{10}, F_{11}) \rightarrow F_{12}$		26.64
$(F_{11}, F_{12}) \rightarrow F_{13}$		16.80
Grand average	**10.30**	**13.68**

TABLE IV
NUMERICAL DATA INDUCED FROM THE INS $F_1, \ldots F_{13}$ AS THE MEAN VALUES OF THE CORRESPONDING DISTRIBUTION FUNCTIONS

IN data samples	**Corresponding numerical data samples**
$(F_1, F_2) \rightarrow F_3$	$(146.89, 150.51) \rightarrow 126.73$
$(F_2, F_3) \rightarrow F_4$	$(150.51, 126.73) \rightarrow 116.04$
$(F_3, F_4) \rightarrow F_5$	$(126.73, 116.04) \rightarrow 100.43$
$(F_4, F_5) \rightarrow F_6$	$(116.04, 100.43) \rightarrow 89.01$
$(F_5, F_6) \rightarrow F_7$	$(100.43, 89.01) \rightarrow 89.19$
$(F_6, F_7) \rightarrow F_8$	$(89.01, 89.19) \rightarrow 83.75$
$(F_7, F_8) \rightarrow F_9$	$(89.19, 83.75) \rightarrow 75.85$
$(F_8, F_9) \rightarrow F_{10}$	$(83.75, 75.85) \rightarrow 71.95$
$(F_9, F_{10}) \rightarrow F_{11}$	$(75.85, 71.95) \rightarrow 61.22$
$(F_{10}, F_{11}) \rightarrow F_{12}$	$(71.95, 61.22) \rightarrow 65.60$
$(F_{11}, F_{12}) \rightarrow F_{13}$	$(61.22, 65.60) \rightarrow 59.56$

IN-based techniques in a wide range of applications. Furthermore, there are instruments for optimizing performance by parameter optimization of the real functions $v(.)$ and $\theta(.)$ in equation (5). In other words, functions $v(.)$ and $\theta(.)$ may introduce useful Riemannian space distortions while retaining the statistical interpretations of INs. Finally, note that this work has demonstrated a "proof-of-concept" regarding the INNN. An advantage of INNN over a convolutional neural network (CNN) is that the INNN is based on sound mathematical theory, it can explain its answers using logic, moreover it represents semantics by partial-order [14], [32].

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

V. Conclusion

This work has paved the way toward engaging IN-based neural networks in agricultural applications. The proposed techniques are low-cost for fast decision making during harvest time based on distributions induced from color histograms of vineyard images. The proposed method is not meant as a substitute of traditional chemical analysis, but rather it is meant for decision-support complement toward an efficient prediction of grape harvest time.

Computer vision provides fast, accurate, direct and non-invasive intervention for predicting grape maturity time. Color features derived from other color spaces (HIS, YIQ, CIELAB, etc.) may be investigated in future work. Moreover, in addition to color, other characteristics such as texture, shape and homogeneity may be considered. Our proposed methodology can thus be extended to other grape varieties, including green ones, as well as to other fruits. Future work plans include acquisition of images of various grapes under constant lighting conditions for an extended period of time until a desired maturity level is reached.

References

[1] T. S. Perry. (2019, 04 October) Want a really hard machine learning problem? Try agriculture, says John Deere labs. [Online]. Available: https://spectrum.ieee.org/view-from-the-valley/robotics/artificial-intelligence/want-a-really-hard-machine-learning-problem-try-agriculture-say-john-deere-labs-leaders

[2] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, no. April, pp. 70–90, 2018.

[3] D. I. Patrício and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," *Computers and Electronics in Agriculture*, vol. 153, no. October, pp. 69–81, 2018.

[4] G. Pajares, I. García-Santillán, Y. Campos, M. Montalvo, J. M. Guerrero, L. Emmi, J. Romeo, M. Guijarro, and P. Gonzalez-de-Santos, "Machine-vision systems selection for agricultural vehicles: A guide," *Journal of Imaging*, vol. 2, no. 34, pp. 1–32, 2016.

[5] A. Bechar and C. Vigneault, "Agricultural robots for field operations: Concepts and components," *Biosystems Engineering*, vol. 149, no. September, pp. 94–111, 2016.

[6] I. R. Donis-González, D. E. Guyer, G. A. Leiva-Valenzuela, and J. Burns, "Assessment of chestnut (Castanea spp.) slice quality using color images," *Journal of Food Engineering*, vol. 115, no. 3, pp. 407–414, April 2013.

[7] E. Mavridou, E. Vrochidou, G. A. Papakostas, T. Pachidis, and V. G. Kaburlasos, "Machine vision systems in precision agriculture for crop farming," *Journal of Imaging*, vol. 5, no. 89, pp. 1–32, 2019.

[8] F. J. Rodríguez-Pulido, R. Ferrer-Gallego, M. L. González-Miret, J. C. Rivas-Gonzalo, M. T. Escribano-Bailón, and F. J. Heredia, "Preliminary study to determine the phenolic maturity stage of grape seeds by computer vision," *Analytica Chimica Acta*, vol. 732, pp. 78–82, 30 June 2012.

[9] F. J. Rodríguez-Pulido, L. Gómez-Robledo, M. Melgosa, B. Gordillo, M. L. González-Miret, and F. J. Heredia, "Ripeness estimation of grape berries and seeds by image analysis," *Computers and Electronics in Agriculture*, vol. 82, pp. 128–133, March 2012.

[10] S. E. Papadakis and V. G. Kaburlasos, "Piecewise-linear approximation of nonlinear models based on probabilistically/possibilistically interpreted Intervals' Numbers (INs)," *Information Sciences*, vol. 180, no. 24, pp. 5060–5076, 2010.

[11] V. G. Kaburlasos, S. E. Papadakis, and G. A. Papakostas, "Lattice computing extension of the FAM neural classifier for human facial expression recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1526–1538, 2013.

[12] V. G. Kaburlasos and A. Kehagias, "Fuzzy inference system (FIS) extensions based on the lattice theory," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 531–546, 2014.

[13] V. G. Kaburlasos and T. Pachidis, "A lattice-computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application," *Information Fusion*, vol. 16, pp. 68–83, 2014.

[14] V. G. Kaburlasos and G. A. Papakostas, "Learning distributions of image features by interactive fuzzy lattice reasoning (FLR) in pattern recognition applications," *IEEE Computational Intelligence Magazine*, vol. 10, no. 3, pp. 42–51, 2015.

[15] V. G. Kaburlasos, G. A. Papakostas, T. Pachidis, and A. Athinellis, "Intervals' numbers (INs) interpolation /extrapolation," in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013)*, Hyderabad, India, 7-10 July 2013.

[16] X. Meng, M. Liu, H. Zhou, J. Wu, F. Xu, and Q. Wu, "Fuzzy c-means on metric lattice," *Automatic Control and Computer Sciences*, vol. 54, no. 1, pp. 30–38, 2020.

[17] V. Kaburlasos, E. Vrochidou, F. Panagiotopoulos, C. Aitsidis, and A. Jaki, "Time series classification in cyber-physical system applications by intervals' numbers techniques," in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2019)*, New Orleans, Louisiana, USA, 23-26 June 2019.

[18] P. Sussner and I. Campiotti, "Extreme learning machine for a new hybrid morphological/linear perceptron," *Neural Networks*, vol. 123, pp. 288–298, 2020.

[19] V. G. Kaburlasos, *Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory. Studies in Computational Intelligence*. Springer, 2006, vol. 27.

[20] V. G. Kaburlasos and G. A. Papakostas, *Introduction to Computational Intelligence – A Holistic Approach (in Greek)*. Hellenic Academic Ebooks (www.kallipos.gr), 2016.

[21] A. L. Ralescu and D. A. Ralescu, "Probability and fuzziness," *Information Sciences*, vol. 34, no. 2, pp. 85–92, 1984.

[22] S. Wonneberger, "Generalization of an invertible mapping between probability and possibility," *Fuzzy Sets and Systems*, vol. 64, no. 2, pp. 229–240, 1994.

[23] V. G. Kaburlasos, "FINs: lattice theoretic tools for improving prediction of sugar production from populations of measurements," *IEEE Transactions on Systems, Man and Cybernetics – Part B*, vol. 34, no. 2, pp. 1017–1030, 2004.

[24] S. Papadakis and V. G. Kaburlasos, "Induction of classification rules from histograms," in *Joint Conference on Information Sciences (JCIS 2007), Proc. 8th International Conference on Natural Computing (NC 2007)*, Salt Lake City, Utah, 18-24 July 2007, pp. 1646–1652.

[25] L. Tran and L. Duckstein, "Comparison of fuzzy numbers using a fuzzy distance measure," *Fuzzy Sets and Systems*, vol. 130, no. 3, pp. 331–341, 2002.

[26] R. Davis. (2020) Do grapes change color? understanding grape veraison. [Online]. Available: https://blog.jordanwinery.com/grape-veraison/

[27] E. Badeka, T. Kalabokas, K. Tziridis, A. Nicolaou, E. Vrochidou, E. Mavridou, G. A. Papakostas, and T. Pachidis, "Grapes visual segmentation for harvesting robots using local texture descriptors," in *Proc. 12th Intl. Conf. on Computer Vision Systems (ICVS 2019)*, Thessaloniki, Greece, 23-25 September 2019, pp. 98–109.

[28] J. Luo, Y. Wang, Q. Wang, R. Zhai, H. Peng, L. Wu, and Y. Zong, "Automatic image segmentation of grape based on computer vision," in *Recent Developments in Intelligent Systems and Interactive Applications*, F. Xhafa, S. Patnaik, and Z. Yu, Eds. Cham, Switzerland: Springer, 2017, vol. 541, pp. 365–370.

[29] N. Behroozi-Khazaei and M. Reza Maleki, "A robust algorithm based on color features for grape cluster segmentation," *Computers and Electronics in Agriculture*, vol. 142, Part A, pp. 41–49, November 2017.

[30] M. Vannucci, V. Colla, M. Sgarbi, and O. Toscanelli, "Thresholded neural networks for sensitive industrial classification tasks," in *Bio-Inspired Systems: Computational and Ambient Intelligence*, ser. Lecture Notes in Computer Science, J. Cabestany, F. Sandoval, A. Prieto, and J. M. Corchado, Eds. Berlin, Germany: Springer, 2009, vol. 5517, pp. 1320–1327.

[31] S. Ding, C. Su, and J. Yu, "An optimizing BP neural network algorithm based on genetic algorithm," *Artificial Intelligence Review*, vol. 36, no. 2, pp. 153–162, 2011.

[32] V. G. Kaburlasos, I. N. Athanasiadis, and P. A. Mitkas, "Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation," *International Journal of Approximate Reasoning*, vol. 45, no. 1, pp. 152–188, 2007.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE