IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Visualizing Classification Results: Confusion Star and Confusion Gear

Amalia Luque[1], Member, IEEE, Mirko Mazzoleni[2], Member, IEEE, Alejandro Carrasco[3], Member, IEEE, and Antonio Ferramosca[2], Member, IEEE

[1]Departamento de Ingeniería del Diseño, Escuela Politécnica Superior, Universidad de Sevilla, Sevilla, 41011 Spain
[2]Department of Management, Information and Production Engineering, University of Bergamo, Via G. Marconi 5, 24044 Dalmine (BG), Italy
[3]Departamento de Tecnología Electrónica, School of Computer Engineering, Universidad de Sevilla, Seville, Spain

Corresponding author: Amalia Luque (e-mail: amalialuque@ us.es).

ABSTRACT Recent developments in machine learning applications are deeply concerned with the poor interpretability of most of these techniques. To gain some insights in the process of designing data-based models it is common to graphically represent the algorithm's results, either in their final or intermediate stage. Specially challenging is the task of plotting multiclass classification results as they involve categorical variables (classes) rather than numeric results. Using the well-known MNIST dataset and a simple neural network as an example, this paper reviews the existing techniques to visualize classification results, from those centered on a particular instance or set of instances, to those representing an overall performance metric. As classification results are commonly summarized in the form of a confusion matrix, special attention is paid to its graphical representation. From this analysis, a new visualization tool is derived, which is presented in two forms: confusion star and confusion gear. The confusion star is centered on the classification errors, while the confusion gear focuses on the classification hits. The proposed visualization tools are also evaluated when facing: (i) balanced and imbalanced classifiers issues; (ii) the problem of representing errors with different orders of magnitude. By using shapes instead of colors to represent the value of each matrix cell, the new tools significantly improve the readability of the confusion matrices. Furthermore, we show how the area enclosed by the confusion stars and gears are directly related to standard classification metrics. The new graphic tools can be also usefully employed to visualize the performances of a sequence of classifiers.

INDEX TERMS Machine learning, classification performance, confusion matrix, data visualization, confusion star, confusion gear.

I. INTRODUCTION

Machine learning models in general, and deep learning algorithms in particular, are powerful algorithms able to provide very good results when there is a pattern to be learnt from available data, but at the cost of operating as a black-box.

On the other hand, having some insights about how they work is a key issue for several reasons: improving the interpretability and explainability of the models [1], debugging and improving architectures and algorithms [2], comparing and selecting results [3], and even for pedagogical purposes [4]. Therefore, a common approach to unveil their functioning relies on some kind of visualization of their inner operation and final results e.g., in the computer vision domain [5].

The main target audience of these tools is the model developer community [6], but also technically skilled model users [7] and even non-experts [8] can benefit of a visual description.

These users may be interested in the visual representation of different types of models' information, such as model architecture [9], neural network's weights [10], convolutional filters' values [11], neurons' activation outputs [12] or edges' backpropagation gradients [13]. However, by far the most represented information is the model's predictions either for a particular instance [14], for a group of instances [15] or for the overall dataset [16].

Many methods have been described with the aim of visualizing the prediction process. An up-to-date comprehensive survey of them, structured using the Five

W's and How questions (Why, Who, What, How, When, and Where), can be found in [17]. Also a perspective of visual analytics for understanding, diagnosing, and refining models is reviewed in [18]. Additionally, different visualizing tools integrating several approaches have been developed [19]–[23].

Focusing on how to visualize the results predicted by machine learning algorithms, different approaches should be considered depending on the type of problem addressed. The information to be represented (prediction results) is qualitatively different for tasks such as regression, classification, clustering, reinforcement learning, etc. This paper addresses the issue of *visualizing the results obtained by multiclass classification algorithms*, since this is one of the most frequent tasks in machine learning applications (for instance, around 75% of the datasets in the well-known University of California Irvine Machine Learning Repository [24] contain classification problems).

In most cases, the performance of a classifier is summarized by a single metric (accuracy, precision, etc.), but "it is important to understand both what a classification metric expresses and what it hides" [25]. For this reason, a classification metric can also be disaggregated as a set of values with the purpose of gaining better insight into the classifier's results.

As for the level of disaggregation to be used in visualizing classification results, three approaches are considered in the paper:

- Low-detailed results, using a single-valued metric for the classification of the whole dataset.
- Medium-detailed results, where the classification of the whole dataset is summarized by a small set of values.
- High-detailed results, representing classification scores for a single instance or a set of instances in the dataset.

Although the paper briefly examines how to represent low and high-detailed classification results, its main focus is on how to visualize them at a medium level of detail, which is commonly described by its multiclass confusion matrix [26].

The main contributions of this research can be summarized as follows:

- Two new approaches to visualize the results of a multiclass classifier are proposed, namely the confusion star and confusion gear graphics.
- Their use as an intuitive guideline to understand the classification behavior is explored.
- Their application to imbalanced datasets is considered.
- Their role to compare different classifiers is highlighted, as well as to understand the influence of classifier's hyperparameters.
- The relationships between the shape of these graphs and common classification metrics are derived.

The paper is organized as follows. Section II describes the structure of the dataset used in the research, defines the classification scoring procedure and formalizes the concept of confusion matrix. Then, in section III, several techniques to visualize classification scores and multiclass confusion matrices are reviewed. The extension of these ideas is addressed in section IV, where the confusion star and confusion gear concepts are presented. Later, in section V these new tools are discussed, tackling issues such as the impact of imbalanced datasets, the inner and outer areas of the graphics, the use of logarithmic scale and the visualization of evolving classifiers by means of a sequence of the new graphics. Finally, the main findings of the research are presented in the conclusion section.

II. METHODOLOGY

A. DATASET

Throughout this research the MNIST (Modified National Institute of Standards and Technology) dataset [27] has been used as the primary dataset. It contains 70,000 images, each of them representing a handwritten digit (0 to 9). The dataset is split into a 60,000 images subset that is used to train the classifier (training dataset) and a 10,000 images subset employed for generalization purposes (testing dataset). In this case there are 10 classes, one for each digit.

This dataset has been widely used as a reference to analyze different classification algorithms. Our goal in this paper is not to obtain a better classifier but, given the results of any of them, to explore how to represent its confusion matrix.

As a first example, a classifier implemented as a very simple neural network has been considered, with only an 8-neurons hidden layer and a sigmoid as activation function. The output layer contains 10 nodes (one for each class) with a *softmax* activation function. Such a network is trained during just 5 epochs, and its generalization results are evaluated on the testing dataset. These test results are used in the following to show different visualization methods.

This classifier is advisedly simple for the purpose of obtaining low performance: in this case, differences among the considered visualization techniques can be more easily appreciated. By increasing the number of hidden layers, the number of nodes per layer, and the number of training epochs, much better classification results can be obtained. As instance, using convolutional neural networks, excellent results (99.8% accuracy) have been reported [28].

In the final part of the paper, it is discussed the evolution of the classification performance as a function of the number of training instances. In this case the MNIST dataset has also been used, now raising the number of neurons in the hidden

layer up to 128-neurons and training the neural networks during 100 epochs.

To show the ability of confusion stars and gears to visualize classification results in problems with a high number of classes, a second dataset, the CIFAR-100, has also been considered [29]. This dataset consists of 60000, 32x32 color images in 100 classes, with 600 images per class, where 50000 images (83%) are used for training and 10000 for test (17%). A 6-layer Convolutional Neural Network (CNN) classifier has been employed, according to the code in [30]. This is not a very powerful classifier as it shows an accuracy of about 40%, while the state of the art classifiers for this problem reach figures over 96% [31]. However this moderate accuracy is quite convenient to depict confusion stars and gears with many classes.

Finally, to show the impact of imbalanced datasets on confusion stars and gears, a reduced version of the Abalone dataset is employed. This dataset, available in [24], derives from a non-machine-learning study [32] and contains physical measurements (height, several lengths, diameter, sex) of the abalone mollusk exemplars, along with the number of "rings" present in the shell. The number of rings is proportional to the age of the mollusk. The purpose is to classify each observation in its age class. Certain classes in the original dataset contains very few instances (some classes with one or no elements) making unaffordable any prediction. To overcome this problem a reduced dataset has been obtained by selecting only 10 classes, from class (age) 4 to 13, containing 3670 instances which represents the 88% of the total population. The resulting dataset contains the same number of classes (10) than the MNIST problem but they are highly imbalanced, which is quite convenient for the sake of comparison. A simple multiclass logistic regression has been used as classifier.

B. CLASSIFICATION SCORE MATRIX

Let us consider a statistical population $\mathcal{P}$ that contains a set of elements, usually in a large and potentially infinite number. In this population n elements are randomly sampled, obtaining a dataset $\mathcal{D} = \{d_1, d_2, \cdots, d_n\}$, where d_i represents the i-th element. Let also be a set of classes $\boldsymbol{\Theta} = \{\theta_1, \theta_2, \cdots, \theta_C\}$ where C is the number of classes, and θ_j represents the j-th class. A certain element $d \in \mathcal{D}$ is defined by a pair $\langle \boldsymbol{\Phi}, \theta \rangle$ formed by a vector $\boldsymbol{\Phi} = [\varphi_1, \varphi_2, \cdots, \varphi_F]$ that contains the F features that define the element, and the class θ to which the element belongs to. Let us call $\mathcal{P}_\Phi = \{\boldsymbol{\Phi_1}, \boldsymbol{\Phi_2}, \cdots, \boldsymbol{\Phi_n}\}$ the set containing the feature vectors $\boldsymbol{\Phi}$ of the population $\mathcal{P}$.

A classifying algorithm $\mathcal{A}$ is defined as a function from the population $\mathcal{P}$ to $\mathbb{R}^C$ (the set of real numbers of dimension C), which can be expressed as $\mathcal{A}: \mathcal{P} \to \mathbb{R}^C$. Therefore, each element belonging to $\mathcal{P}$ is associated to a scoring vector $\boldsymbol{\Psi} = [\psi_1, \psi_2, \cdots, \psi_C]$, that is, a score for each class in $\boldsymbol{\Theta}$. If the scores can be interpreted as probabilities, the algorithm is a probabilistic classifier. Otherwise, if scores are binary values (0,1) the algorithm is a hard classifier.

A decision rule $\mathcal{R}$ is defined as a function which associates a scoring vector $\boldsymbol{\Psi}$, defined in $\mathbb{R}^C$, in an estimation of the class $\hat{\theta} \in \boldsymbol{\Theta}$.

Finally, a classifier $\mathcal{C}$ is defined as an ordered pair of functions $\langle \mathcal{A}, \mathcal{R} \rangle$ indicating that it first applies the classification algorithm $\mathcal{A}$, and then the decision rule $\mathcal{R}$. So, $\mathcal{C}: \mathcal{P}_\Phi \xrightarrow{\mathcal{A}} \mathbb{R}^C \xrightarrow{\mathcal{R}} \boldsymbol{\Theta}$. Considering not the whole dataset but each single element, a classifier can be described as two sequential transformations, $\boldsymbol{\Phi} \xrightarrow{\mathcal{A}} \boldsymbol{\Psi} \xrightarrow{\mathcal{R}} \hat{\theta}$, where $\hat{\theta}$ is the class estimate.

Therefore, the result obtained applying the classifier $\mathcal{C}$ to an element in the dataset $\mathcal{D}$ is a scoring vector $[\psi_1, \psi_2, \cdots, \psi_C]$, and a class estimation $\hat{\theta}$. To measure the classifier performance, the actual class of the element must also be included. Then, the performance of a classifier, operating on a dataset with n instances, can be expressed by the *score matrix* ($\boldsymbol{SM}$), with $\boldsymbol{SM} \in \mathbb{R}^{n \times (C+2)}$, given by

$$\boldsymbol{SM} \equiv \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{1C} & \hat{\theta}_1 & \theta_1 \\ \psi_{21} & \psi_{22} & \cdots & \psi_{2C} & \hat{\theta}_2 & \theta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \psi_{n1} & \psi_{n2} & \cdots & \psi_{nC} & \hat{\theta}_n & \theta_n \end{bmatrix}. \quad (1)$$

This matrix contains the information about the performance of the classifier at its *maximum level* of disaggregation.

C. CONFUSION MATRIX

In many situations, the classifier performance is analyzed not considering the scores associated to each class, but just comparing the estimated and the actual class for each instance in the dataset. So, by discarding the first C columns of the score matrix, the more compact *estimation matrix* ($\boldsymbol{EM}$) is obtained

$$\boldsymbol{EM} \equiv \begin{bmatrix} \hat{\theta}_1 & \theta_1 \\ \hat{\theta}_2 & \theta_2 \\ \vdots & \vdots \\ \hat{\theta}_n & \theta_n \end{bmatrix}. \quad (2)$$

The estimation matrix $\boldsymbol{EM}$ is has a smaller dimension (less columns) than the score matrix $\boldsymbol{SM}$, but it still has a high level of disaggregation, since it contains information for each instance in the dataset. Therefore, it is common to summarize it using the *confusion matrix* ($\boldsymbol{CM}$) defined as

$$\boldsymbol{CM} \equiv \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1C} \\ m_{21} & m_{22} & \cdots & m_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ m_{C1} & m_{C2} & \cdots & m_{CC} \end{bmatrix}, \quad (3)$$

where m_{ij} represents the number of instances of class θ_i estimated by the classifier as belonging to the class $\hat{\theta}_j$. The results obtained classifying the MNIST dataset with the neural network previously described can be summarized in the confusion matrix shown in TABLE I. Last column also shows that there are a similar number of instances in each class, which means that classes are quite balanced.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

TABLE I
CONFUSION MATRIX OBTAINED CLASSIFYING THE MNIST DATASET WITH A VERY SIMPLE NEURAL NETWORK

		Predicted class										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Inst.**
Actual class	**0**	700	1	9	12	0	214	20	20	3	1	980
	1	1	995	4	15	0	1	8	2	108	1	1135
	2	67	184	468	38	10	52	14	152	38	9	1032
	3	102	42	20	674	1	76	6	43	41	5	1010
	4	1	0	0	7	652	10	40	50	48	174	982
	5	65	13	16	47	18	608	38	15	63	9	892
	6	27	0	1	1	1	104	742	29	28	25	958
	7	1	25	24	13	31	1	6	880	5	42	1028
	8	7	52	7	24	11	95	32	11	701	34	974
	9	5	1	0	18	156	17	15	127	34	636	1009
Estim.		976	1313	549	849	880	1178	921	1329	1069	936	10000

TABLE II
CONFUSION MATRIX FOR A MULTICLASS CLASSIFICATION

		Estimated Class				
		$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_2}$	⋯	$\boldsymbol{\theta_C}$	**Instances**
Actual Class	$\boldsymbol{\theta_1}$	$\lambda_{11}m_1$	$\lambda_{12}m_1$	⋯	$\lambda_{1C}m_1$	m_1
	$\boldsymbol{\theta_2}$	$\lambda_{11}m_1$	$\lambda_{12}m_1$	⋯	$\lambda_{1C}m_1$	m_2
	⋮	⋮	⋮	⋱	⋮	⋮
	$\boldsymbol{\theta_C}$	$\lambda_{11}m_1$	$\lambda_{12}m_1$	⋯	$\lambda_{1C}m_1$	m_C
Estimations		g_1	g_2	⋯	g_C	m

In the definition of the confusion matrix, is usual to describe m_{ij} as a fraction of the total number of instances m_i belonging to the class θ_i. By calling this ratio $\lambda_{ij} \equiv m_{ij}/m_i$, then m_{ij} can be expressed as $m_{ij} = \lambda_{ij} \cdot m_i$, and the confusion matrix can be rewritten as

$$\boldsymbol{CM} = \begin{bmatrix} \lambda_{11}m_1 & \lambda_{12}m_1 & \dots & \lambda_{1C}m_1 \\ \lambda_{21}m_2 & \lambda_{22}m_2 & \dots & \lambda_{2C}m_2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{C1}m_C & \lambda_{C2}m_C & \dots & \lambda_{CC}m_C \end{bmatrix} = \boldsymbol{\Lambda} \circ \boldsymbol{M}. \quad (4)$$

The symbol $\circ$ represents the element-wise multiplication (also called Hadamard product), $\boldsymbol{\Lambda}$ is the unit confusion matrix expressed by,

$$\boldsymbol{\Lambda} \equiv \begin{bmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1C} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{C1} & \lambda_{C2} & \dots & \lambda_{CC} \end{bmatrix}, \quad (5)$$

and $\boldsymbol{M}$ is the matrix defined by

$$\boldsymbol{M} \equiv \begin{bmatrix} m_1 & m_1 & \dots & m_1 \\ m_2 & m_2 & \dots & m_2 \\ \vdots & \vdots & \ddots & \vdots \\ m_C & m_C & \dots & m_C \end{bmatrix}. \quad (6)$$

TABLE II summarizes the main elements considered in the definition of the confusion matrix, where g_j is the number of instances estimated as belonging to the j-th class.

III. VISUALIZING CLASSIFICATION RESULTS

A. CLASSIFICATION SCORES OF INSTANCES

Fully detailed classification results regarding the i-th instance correspond to the i-th row of the score matrix (1) defined by

$$\boldsymbol{SM_i} \equiv [\psi_{i1} \quad \psi_{i2} \quad \dots \quad \psi_{iC} \quad \hat{\theta}_i \quad \theta_i]. \quad (7)$$

The classification scores for the first three instances in the MNIST testing dataset can be depicted as in Fig. 1 (in colors blue, orange and green respectively).

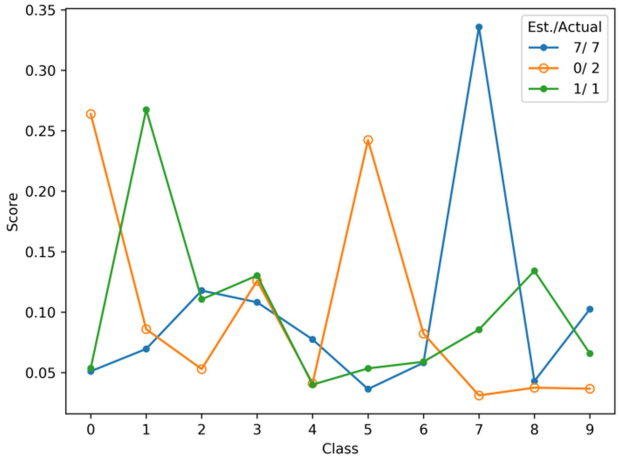

FIGURE 1. Classification scores for the first three instances in the MNIST testing dataset. (Filled dots) Correctly classified instances. (Empty dots) Not correctly classified instances.

In this classifier the scores are generated by the softmax activation function of the 10-neurons output layer, so they are in the range $[0,1]$, sum up to 1 and, therefore, they can be interpreted as probabilities. For example, the first instance (represented by a blue line) has a $(0.05,0.07,0.12,\cdots)$ probability of belonging to the class $(0,1,2,\cdots)$. Belonging to class 7 obtains the highest probability (0.34), so this is the class estimated by the classifier. In this case, the instance is classified correctly, which is indicated using filled dots. For

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

the second instance (represented by an orange line), belonging to class 0 obtains the highest probability (0.26). In this case, this is an error as the actual class is 2 which is indicated using empty dots.

This type of representation only has meaning for a single instance or for a very reduced number of them. In order to depict classification scores for many instances a scatter polar plot has been proposed [33] as it is shown in Fig. 2 for every instance in the MNIST testing dataset.

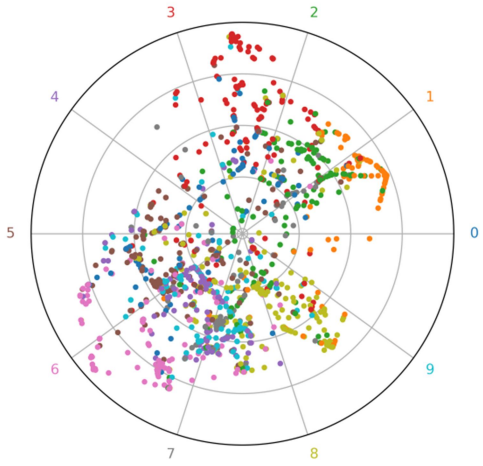

FIGURE 2. Polar representation of the classification scores for the MNIST testing dataset.

Each class is represented by a certain *angle* φ_j, which for the j-th class is defined by

$$\varphi_j = \frac{2\pi(j-1)}{C}. \tag{8}$$

The classification result for the i-th instance is depicted as a dot in a position defined by its vector

$$\vec{r}_i = \sum_{j=1}^{C} \vec{r}_{ij} = \sum_{j=1}^{C} \psi_{ij} \angle \varphi_j, \tag{9}$$

where $\vec{r}_{ij}$ is a vector of module ψ_{ij} and phase φ_j.

B. CLASSIFICATION SCORES OF CLASSES

A partial perspective of the score matrix may consider not every instance in the dataset, but only those belonging to a certain class. In this case, the scoring results of the instances belonging to the j-th class are defined by a slice of the score matrix $\boldsymbol{SM}^{(j)} \equiv \{\boldsymbol{SM_i}\}, \forall i | \theta_i = j$. As the level of disaggregation of this matrix is still very high, it is commonly summarized using some statistics for each column (mean score value, standard deviation, density function, etc.).

Fig. 3 depicts one of these summaries, here in the form of a boxplot. The i-th subplot considers the m_i instances belonging to the i-th class and the j-th box indicates the distribution of the values $\psi_{ij}, \forall i | d_i \in \theta_j$, that is the scores of elements of the i-th class that are being estimated as belonging to the j-th class.

The instances belonging to class 1, for example, are estimated as belonging to class 1 with a probability distributed as it is shown in the second box of the second plot, clearly outperforming the remaining probability distributions. Then, very good classification results should be expected for instances belonging to class 1.

Conversely, instances belonging to class 2 (third plot) have a probability of being correctly classified as it is shown in the third box. This distribution is only slightly better than the ones corresponding to the estimated classes 1 and 7, so many classification errors should be expected for instances belonging to class 2.

C. REPRESENTATION OF THE CONFUSION MATRIX

Let us now focus on how to represent the classification results using a *medium* level of detail, that is, based on its confusion matrix. The most common way to depict a certain multiclass confusion matrix is straightforwardly drawing it as a $C \times C$ colored grid where each cell has a color scaled according to its value. Sometimes the cell also contains a text with its numeric value, as it shown in Fig. 4.

In case of an imbalanced dataset, it is better to represent the *unit confusion matrix*, commonly expressed by the percentage values, as it is depicted in Fig. 5.

The confusion matrix or the unit confusion matrix can be alternatively represented as in Fig. 6 where, for each actual class, a set of C stacked bars are drawn. The height of each bar in a certain stack (actual class) is proportional to the number (or ratio) of instances estimated as belonging to each class, that is, corresponding to the values of a row in the confusion matrix. A similar stacked bar approach is used in [34].

D. REPRESENTATION OF BINARY CONFUSION MATRICES

Sometimes it is worth to assess the classification results of one class versus all the remaining ones (OvA binary classification). So, let us consider the instances belonging to the i-th class which will be denoted as the "positive" (P) class. The remaining instances belong to different classes which will be collectively denoted as the "negative" (N) class. In this way, the number of instances correctly classified as positives (TP: True Positives) is $TP = m_{ii}$. Similarly, the number of instances erroneously classified as positives (FN: False Negatives) is

$$FN = m_i - m_{ii} = \left(\sum_{\substack{k=1 \\ k \neq i}}^{C} m_{ik} \right) - m_{ii}. \tag{10}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

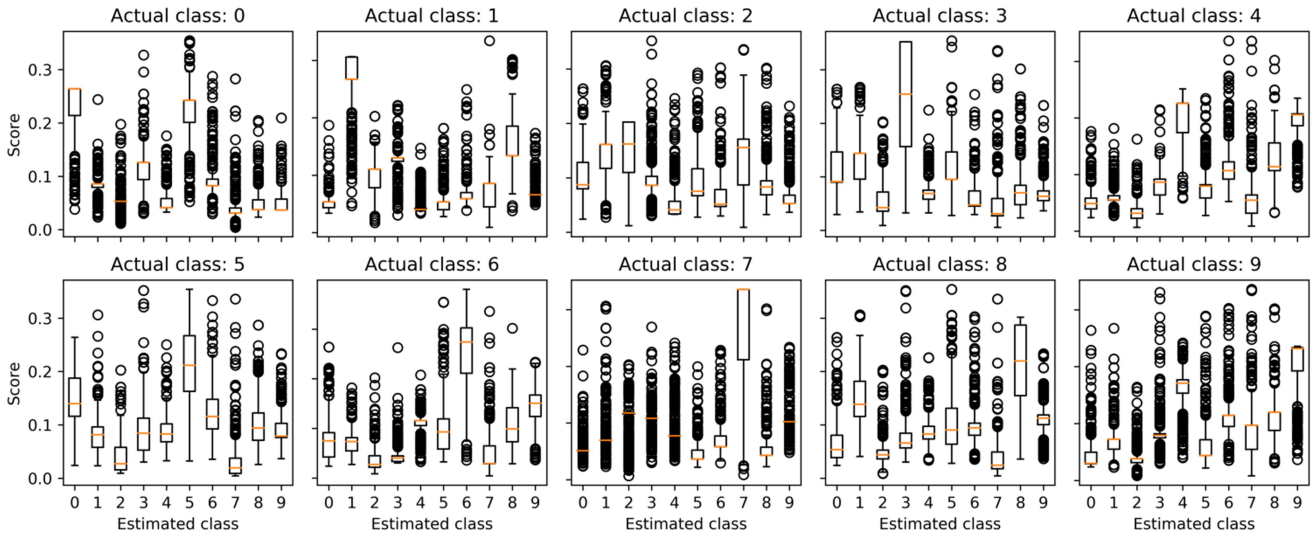

FIGURE 3. **Statistical distribution of the classification scores (probabilities) for the instances in the MNIST testing dataset belonging to a certain class.**

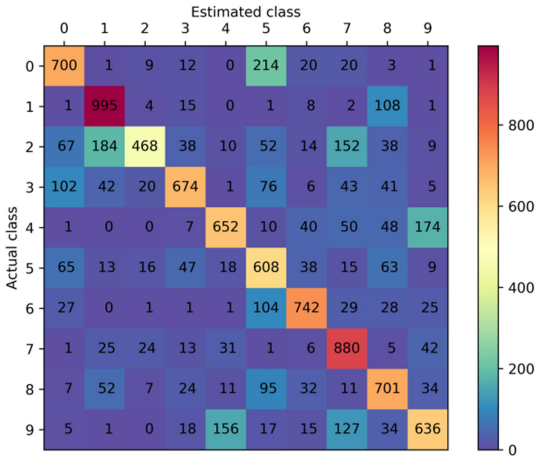

FIGURE 4. **Straightforward colored grid representation of the confusion matrix corresponding to the classification of the MNIST testing dataset.**

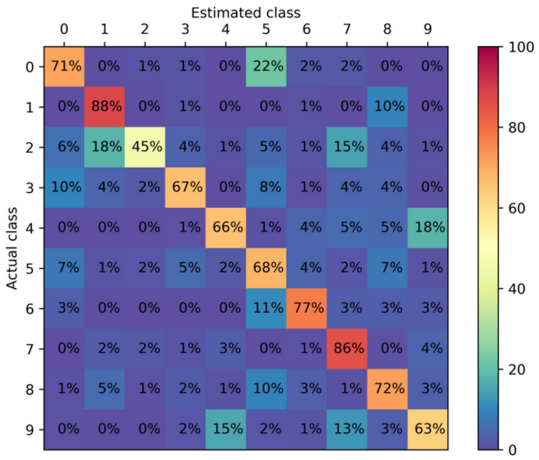

FIGURE 5. **Straightforward colored grid representation of the unit confusion matrix corresponding to the classification of the MNIST testing dataset.**

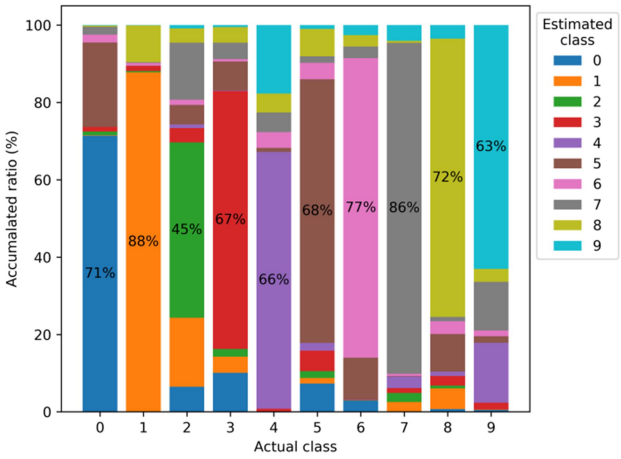

FIGURE 6. **Stacked bar representation of the unit confusion matrix corresponding to the classification of the MNIST testing dataset.**

The number of elements not belonging to the i-th class (that is, belonging to the negative class) which are erroneously classified (FP: False Positives) is

$$FP = \left(\sum_{\substack{k=1 \\ k \neq i}}^{C} m_{ki} \right) - m_{ii}. \tag{11}$$

Finally, the number of elements not belonging to the i-th class (that is, belonging to the negative class) which are correctly classified (TN: True Negatives) is

$$TN = m_N - FP = m - \sum_{\substack{k=1 \\ k \neq i}}^{C} m_{ki}. \tag{12}$$

Considering these results, the binary matrices corresponding to every class can be represented as it is shown in Fig.7. Alternatively, they can be represented using stacked bar plots, as it is depicted in Fig. 8.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

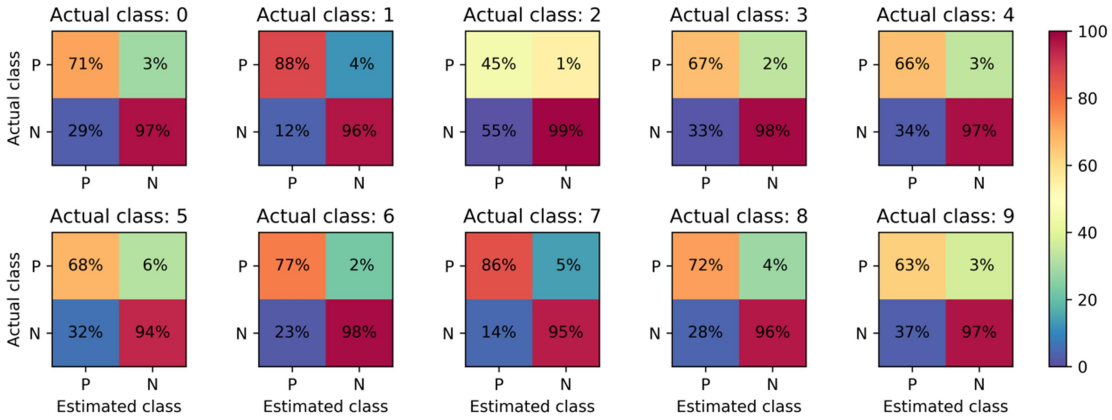

FIGURE 7. **Straightforward representation of the unit binary confusion matrices corresponding to the classification of the MNIST testing dataset.**

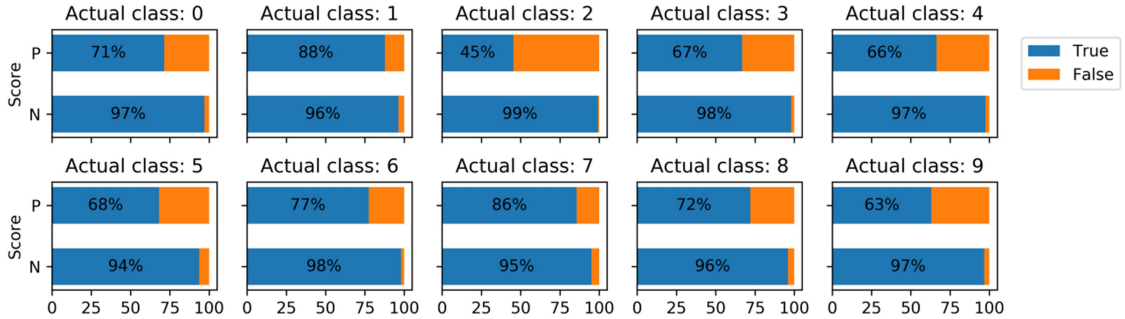

FIGURE 8. **Stacked bar representation of the unit binary confusion matrices corresponding to the classification of the MNIST testing dataset.**

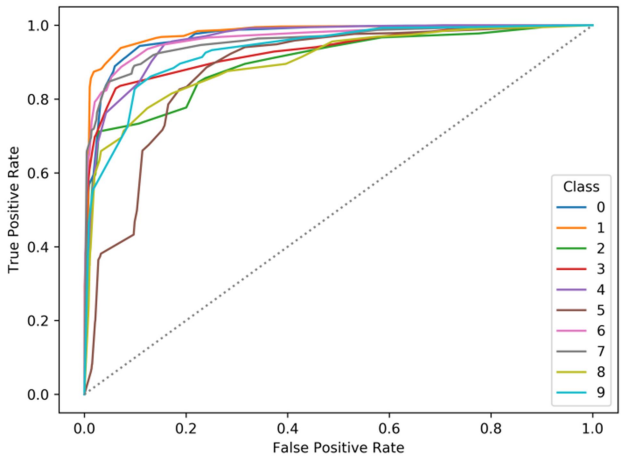

FIGURE 9. ***ROC curves corresponding to the OvA binary classification of the MNIST testing dataset.***

Binary classification results can also be analyzed using the receiver operating characteristic (ROC) curve [35]. Converting the classification scores for an instance into its estimated class requires a decision rule $\mathcal{R}$ which, in the binary case, is usually a threshold τ. If the score of belonging to the positive class ψ_{iP} is greater than the threshold, the instance is estimated as positive; otherwise as negative. So, the elements of the binary confusion matrix depend on τ, and also their related metrics. Specifically, the True Positive Rate (TPR) and the False Positive Rate (FPR) are defined as

$$TPR(\tau) \equiv \frac{TP(\tau)}{m_P}; \quad FPR(\tau) \equiv 1 - \frac{TN(\tau)}{m_N}. \tag{13}$$

The ROC is built as a parametric curve in τ, with $FPR(\tau)$ in the horizontal and $TPR(\tau)$ in the vertical axis. The resulting ROC curves for the 10 binary classifiers are depicted in Fig. 9.

FIGURE 10. **Chord diagram representation of the confusion matrix corresponding to the classification of the MNIST testing dataset.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

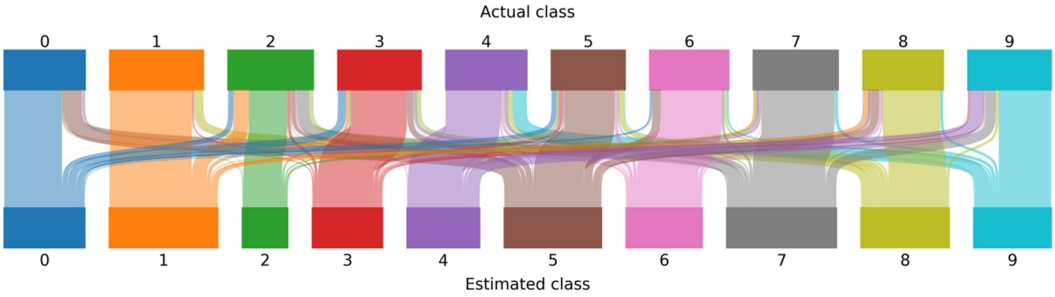

FIGURE 11. Sankey diagram representation of the confusion matrix corresponding to the classification of the MNIST testing dataset.

E. ALTERNATIVE REPRESENTIONS OF THE CONFUSION MATRIX

Some authors have proposed alternative representation for the confusion matrix such as, for instance, in [14] where a *chord diagram*, called by the authors *confusion wheel*, is used. This plot is depicted in Fig.10 where each class corresponds to a circular sector with a size proportional to the number of instances belonging to that class. Later a chord is drawn starting at the actual class sector and ending at the estimated class sector. The width of the chord at each side is proportional to the number of instances belonging to that class classified as belonging to the other side's class. The color of the chord is that of its widest side.

Also in [36] it is proposed to represent the confusion matrix using a *Sankey diagram* as in Fig. 11 . In the upper part each class (origin) is represented by a rectangle with a width proportional to the number of instances belonging to that class. In the lower part, the estimated classes (destinations) are drawn with a width proportional to the number of instances predicted as belonging to that class. The ribbons drawn in the middle represent the instance belonging to the upper side class but classified as belonging to the lower side class.

In [37] the confusion matrix is conceived as a similarity matrix between classes. Then, it is transformed in its opposite, that is, a dissimilarity or distance matrix. Finally this matrix is represented in a two-dimensional plane using the multidimensional scaling (MDS) technique.

The result is shown in Fig. 12 where each class is represented by a point in the new 2D plane. The closer a pair of classes, the more similar they are and, therefore, the more difficult is to separate them. For example, classes 4 and 9 are very close in the MDS plane, which means that it is very difficult to separate them and so a high number of classification errors should be expected.

Along with the individual representations described above, it is also common to find visual representations of the classifier results that combine several of the preceding graphs.

Although these more sophisticated graphics may seem visually very appealing, they do not necessarily provide additional information compared to the more conventional representations. Therefore, in the following section, a new graphical representation is proposed.

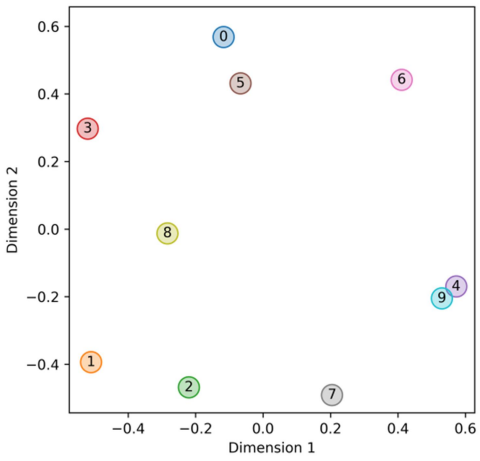

FIGURE 12. MDS transformation of the confusion matrix corresponding to the classification of the MNIST testing dataset.

IV. BEYOND CONFUSION MATRIX

A. LINEAR REPRESENTATION OF THE CONFUSION MATRIX

Let us consider the m_i instances belonging to the i-th class. The results of their classification are summarized in the i-th row of the confusion matrix, $\boldsymbol{CM_i} \equiv [m_{i1} \quad m_{i2} \quad \dots \quad m_{iC}]$, where m_{ij} represents the number of instances belonging to the i-th class, estimated as belonging to the j-th class.

Then, it is possible to represent the confusion matrix as a sequence of C lines, each of them corresponding to a row $\mathbf{CM_i}$. Every line is defined by C values, corresponding to each m_{ij} elements. The result is depicted in Fig. 13. A similar approach is used in [38].

In a good classifier most instances are correctly estimated as belonging to its actual class, so $m_{ii} \approx m_i$; $m_{ij} \approx 0, \forall j \neq i$. That is, a single very high value escorted by the remaining very low values. This important imbalance in the values of each row is clearly seen in the plot and it makes difficult its interpretation. This is also the reason why such graphic is not commonly used to represent the confusion matrix.

To overcome the issues raised in the previous representation, the $\boldsymbol{CM_i}$ containing the classification results corresponding to the i-th class is transformed into a new vector $\boldsymbol{EM_i} \equiv [e_{i1} \quad e_{i2} \quad \dots \quad e_{iC}]$, where its elements are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

defined as $e_{ii} = m_i - m_{ii}$ and $e_{ij} = m_{ij}, \forall j \neq i$. Then, for a perfect classification, $e_{ij} = 0, \forall j$. The matrix $\boldsymbol{EM} = \{e_{ij}\}$ is denominated the error matrix.

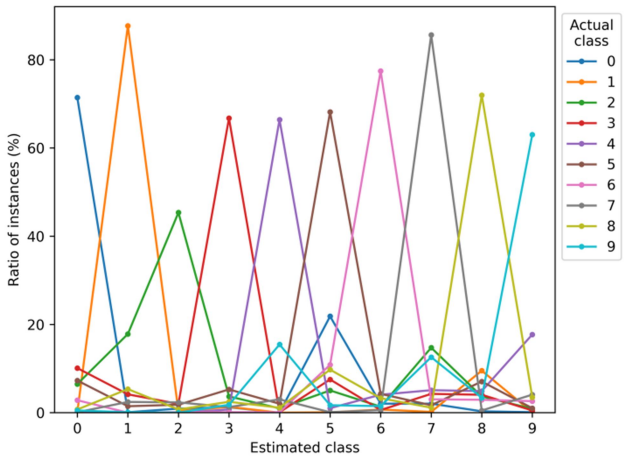

FIGURE 13. Linear representation of the confusion matrix corresponding to the classification of the MNIST testing dataset.

The i-th row of this matrix can also be formulated in terms of the ratio over the total number of instances belonging to the i-th class, $\boldsymbol{EM_i} = [\epsilon_{i1} m_i \quad \epsilon_{i2} m_i \quad \ldots \quad \epsilon_{iC} m_i]$, where the ratio $\epsilon_{ij} = e_{ij}/m_i$. The matrix $\mathbf{E} = \{\epsilon_{ij}\}$ is denominated the unit error matrix and can be represented as a sequence of C lines, as it is shown in Fig. 14.

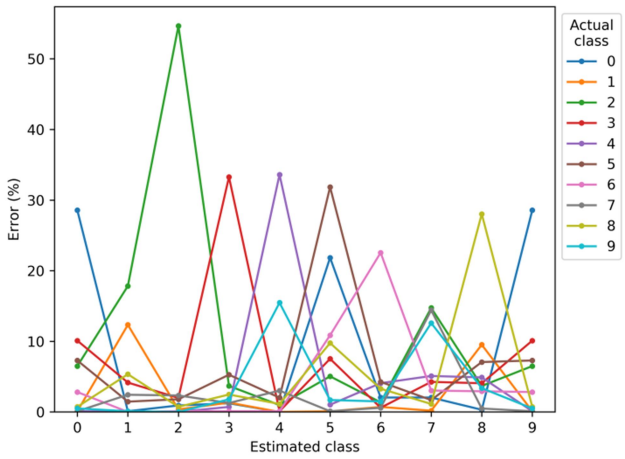

FIGURE 14. Linear representation of the unit error matrix corresponding to the classification of the MNIST testing dataset.

In the previous linear representation (Fig. 14) an abnormally high value is observed in each line, corresponding to the element e_{ii}, that is, the number of instances belonging to the i-th class erroneously classified as belonging to any other class.

To explain these peaks let us first remind that the C elements of the i-th row in the confusion matrix are mutually dependent having $C - 1$ degrees of freedom, that is, they obey the equation

$$\sum_{j=1}^{C} m_{ij} = m_{ii} + \sum_{j \neq i} m_{ij} = m_i. \tag{14}$$

Then, the number of hits (correct classifications) for the i-th class is

$$m_{ii} = m_i - \sum_{j \neq i} m_{ij}. \tag{15}$$

Recalling the definition of the elements of the error matrix, its diagonal elements can be written as

$$e_{ii} = m_i - m_{ii} = \sum_{j \neq i} m_{ij} = \sum_{j \neq i} e_{ij}. \tag{16}$$

The term e_{ij} counts the number of instances belonging to the i-th class, erroneously classified as belonging to the j-th class. Calling $\bar{e}_{ij}$ its mean value, $\forall j \neq i$, it can be written that $e_{ii} = (C-1) \cdot \bar{e}_{ij}$. Then, in the MNIST example (with $C = 10$), the value of e_{ii} will be 9 times higher than the mean of the remaining e_{ij}. This is the reason why a peak appears in the linear representation of Fig. 14. The distribution of the classification errors for each class is depicted in Fig. 15, where it is clearly shown that the value of e_{ii} (in green) is much higher (9 times) than the value of $\bar{e}_{ij}$ (in blue).

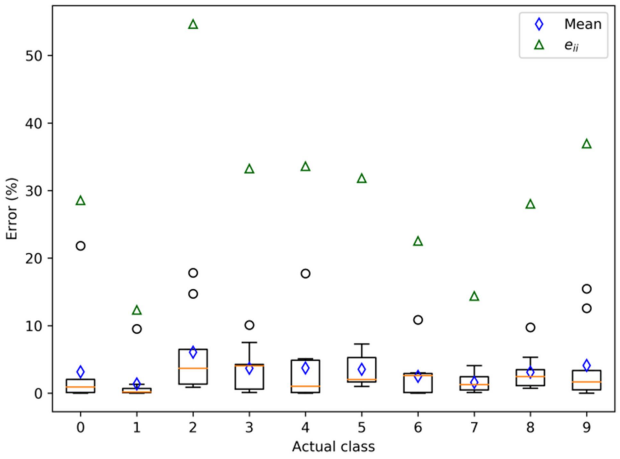

FIGURE 15. Boxplot containing the distribution of the values for $e_{ij}, \forall j \neq i$. The mean value $\bar{e}_{ij}$ (in blue) and the e_{ii} (in green) are also depicted. All the error values are expressed in %.

Considering the $C - 1$ degrees of freedom in the rows of the error matrix, any of them can be omitted without losing information. Then, removing the element e_{ii} is a convenient decision as it eliminates the peaks in the plot, as it is depicted in Fig. 16.

It must be noted that the horizontal axis does not indicate the estimated class but an index to this class once the redundant element has been removed, that is, the value corresponding to the same class. Then, for instance, in the green line (actual class 2), the index corresponds to the estimated classes 0, 1, 3, 4,…, 9, a sequence where the class 2 has been omitted. More formally, for the i-th actual class (the row in the matrix) and the j-th estimated class (column),

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

the index k of the estimated class is defined by the expression

$$\begin{cases} k = j, \forall j < i \\ k = j - 1, \forall j > i \end{cases} \quad (17)$$

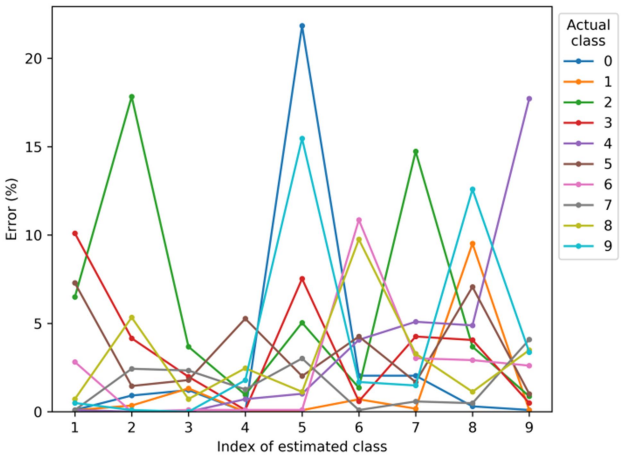

FIGURE 16. Linear representation without redundancies of the unit error matrix corresponding to the classification of the MNIST testing dataset.

B. STEP REPRESENTATION OF THE ERROR MATRIX

In the linear representation without redundancies of the unit error matrix (Fig. 16) let us focus on a particular class, for instance, class 2 as this is the class obtaining the worst classification results. The row of the matrix corresponding to this class can be represented as in Fig. 17.

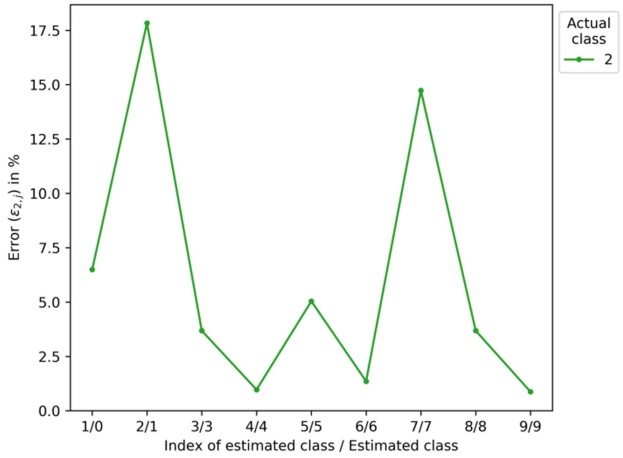

FIGURE 17. Linear representation without redundancies of the unit error matrix corresponding to the classification of the MNIST testing dataset (actual class 2).

Recalling that the row values in the error matrix are $e_{ij} = m_{ij}, \forall j \neq i$, the sum of this values is

$$\sum_{j \neq i} e_{ij} = e_{ii} = m_i - m_{ii}. \quad (18)$$

As $e_{ij} = \epsilon_{ij} m_i$, this equation can be rewritten as

$$\sum_{j \neq i} \epsilon_{ij} = \frac{1}{m_i} \sum_{j \neq i} e_{ij} = \frac{1}{m_i}(m_i - m_{ii}) = 1 - \frac{m_{ii}}{m_i}, \quad (19)$$

which is the sum of the values in Fig. 17.

The term m_{ii}/m_i is usually denominated the True Positive Rate of the i-th class (TPR_i), also known as Sensitivity or Recall. Its complementary, that is $1 - TPR_i$, it is called False Negative Rate (FNR_i) or Miss Rate. Then it can be said that the sum of values in Fig. 17 is

$$\sum_{j \neq i} \epsilon_{ij} = FNR_i. \quad (20)$$

To visualize this value as the area under the line in Fig. 17, it is better to transform the linear representation of the error matrix in a step representation, as it is depicted in Fig. 18. There, each non-redundant value of the error matrix for the i-th class is represented as a step of unit width. The linear equivalent representation is also drawn as a dashed line.

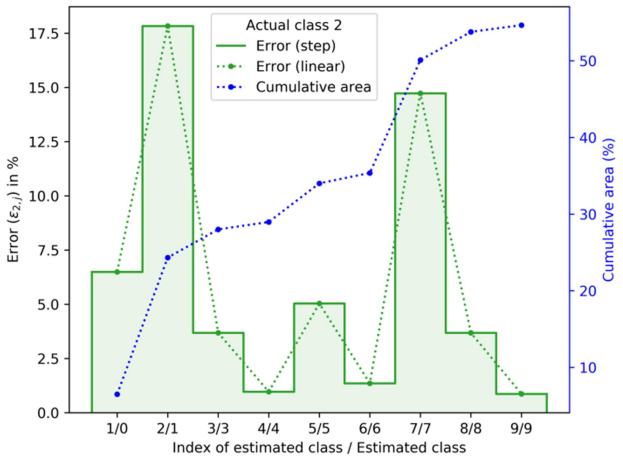

FIGURE 18. Step representation without redundancies of the unit error matrix corresponding to the classification of the MNIST testing dataset (actual class 2). The dashed green line is the equivalent linear representation. The dashed blue line is the cumulative area.

Considering the unit width of each step, the area under the step line is

$$A_i = \sum_{j \neq i} A_{ij} = \sum_{j \neq i} (1 \cdot \epsilon_{ij}) = \sum_{j \neq i} \epsilon_{ij} = FNR_i. \quad (21)$$

The cumulative values of these areas are also drawn in the graphic (dashed blue line).

C. POLAR REPRESENTATION OF THE ERROR MATRIX

The visualization of the error matrix row for class 2 (linearly represented in Fig. 17), can be redrawn in a radial shape. For this purpose, $C - 1$ radii are sketched, each one corresponding to a non-redundant element of the i-th row in the error matrix. The k-th non-redundant element is represented by a line at an angle (respect to the horizontal)

$$\varphi_k = \frac{2\pi k}{C - 1}. \quad (22)$$

Then the angular width corresponding to each class is

$$\Delta\varphi = \frac{2\pi}{C - 1}. \quad (23)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

The result of this plot is depicted in Fig. 19. It must be noted again that the labels in the outermost circle do not indicate the estimated class but the indices to the estimated class.

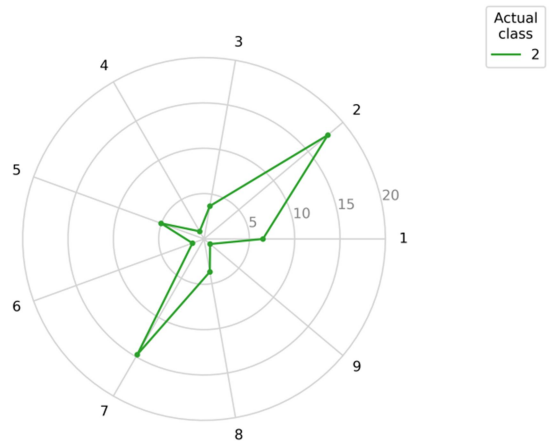

FIGURE 19. Radial representation without redundancies of the unit error matrix corresponding to the classification of the MNIST testing dataset (actual class 2).

To make the resulting area meaningful, it is better to transform the radial representation of the error matrix into a representation by arcs, as shown in Fig. 20. In this plot, which can be denominated the *sectorial* or *pie representation*, each non-redundant value of the error matrix for the i-th class is represented by a circular sector of constant angular width, $\Delta\varphi$. The dashed line represents the equivalent radial representation. The area inside the resulting plot is

$$A_i = \sum_{j\neq i} A_{ij} = \sum_{j\neq i} (\Delta\varphi \cdot \epsilon_{ij}) = \sum_{j\neq i} \left(\frac{2\pi}{C-1} \cdot \epsilon_{ij}\right) = \frac{2\pi}{C-1} \sum_{j\neq i} \epsilon_{ij} = \frac{2\pi}{C-1} FNR_i. \quad (24)$$

It can be seen that this area is proportional to the miss rate.

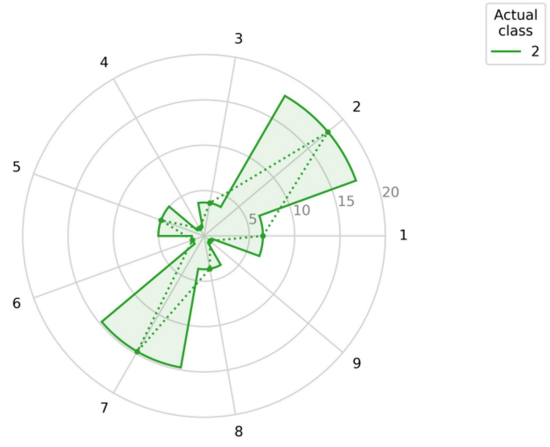

FIGURE 20. Sectorial representation without redundancies of the unit error matrix corresponding to the classification of the MNIST testing dataset (actual class 2). The dashed line is the equivalent radial representation.

D. CONFUSION STAR

In the radial (Fig. 19) and sectorial (Fig. 20) plots discussed in the previous subsection, the representation of a single row of the error matrix have been addressed. To extend this visualization to the whole matrix, one plot for each actual class can be drawn, using different colors to distinguish them. The resulting graphic is depicted in Fig. 21.

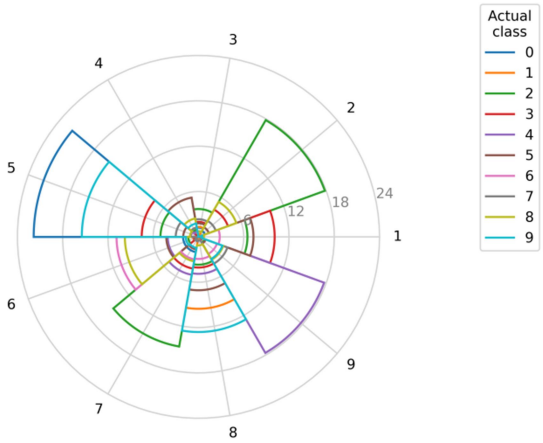

FIGURE 21. Sectorial representation without redundancies of the unit error matrix corresponding to the classification of the MNIST testing dataset. Each color represents the results for an actual class (row of the error matrix).

Reading this plot is not an easy task as the C lines are overlapped. An alternative to improve its readability is to divide the circle in C regions, each one corresponding to an actual class (a row of the error matrix). Then, each region is again divided into $C-1$ sectors, one for each column once the redundant e_{ii} element is removed.

If the C regions have the same size a *balanced* representation is obtained where the angular separation between two radii is

$$\Delta\varphi = \frac{2\pi}{C \cdot (C-1)}. \quad (25)$$

The so obtained star-like result is depicted in Fig. 22.

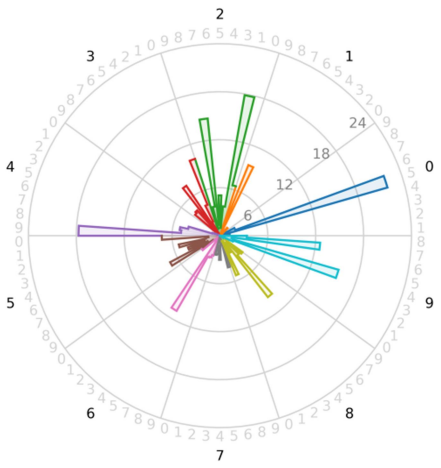

FIGURE 22. Balanced confusion star corresponding to the classification of the MNIST testing dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

This shape justifies naming this representation as the *confusion star*. It must be noted that the gray labels in the outermost circle do not indicate the estimated classes but the indices to these classes once the redundant elements e_{ii} have been removed. In [39] a similar although simpler polygonal solution is proposed with the name of *cobweb*.

E. CONFUSION GEAR

The confusion star has been defined based on the error matrix. An alternative election is to use the classification hits instead of the errors. So, the classification results of the instances belonging to the i-th class, summarized in the i-th row of the confusion matrix $\boldsymbol{CM_i}$, are now transformed in the vector $\boldsymbol{HM_i} \equiv [w_{i1} \quad w_{i2} \quad \dots \quad w_{iC}]$, whose elements are defined as $w_{ii} = m_{ii}$ and $w_{ij} = m_i - m_{ij}, \forall j \neq i$. For a perfect classification $w_{ij} = m_i, \forall j$. The matrix $\boldsymbol{HM} = \{w_{ij}\}$ is called the *hit matrix* of the classifier.

The i-th row of this matrix can also be formulated in terms of the ratio over the total number of instances belonging to the i-th class, $\boldsymbol{HM_i} = [\psi_{i1}m_i \quad \psi_{i2}m_i \quad \dots \quad \psi_{iC}m_i]$, where the ratio $\psi_{ij} = w_{ij}/m_i$. The matrix $\boldsymbol{\Psi} = \{\psi_{ij}\}$ is called the *unit hit matrix*.

To represent this matrix, a procedure similar to that used in the representation of the error matrix is followed: the circle is divided into C regions (one for class) and then each region is again divided into $C-1$ sectors, one for each column once the redundant w_{ii} element is removed. If the C regions have the same size, a *balanced* representation of the hit matrix is obtained as in Fig. 23. The resemblance of this graph to a gear is used to refer to it as the *confusion gear*.

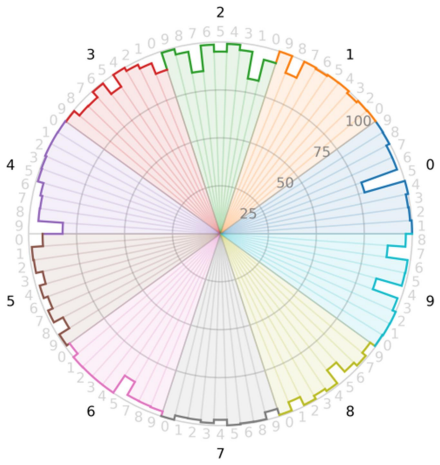

FIGURE 23. Balanced confusion gear corresponding to the classification of the MNIST testing dataset.

Recalling that the row values in the hit matrix are $w_{ij} = m_i - m_{ij}, \forall j \neq i$, the sum of these values is

$$\sum_{j \neq i} w_{ij} = \sum_{j \neq i} m_i - m_{ij} = (C-1)m_i - \sum_{j \neq i} m_{ij}. \quad (26)$$

Considering that

$$\sum_{j=1}^{C} m_{ij} = m_{ii} + \sum_{j \neq i} m_{ij} = m_i, \quad (27)$$

then

$$\sum_{j \neq i} m_{ij} = m_i - m_{ii}, \quad (28)$$

and substituting this result in (26), it is obtained that

$$\sum_{j \neq i} w_{ij} = (C-2)m_i + m_{ii}. \quad (29)$$

As $\psi_{ij} = w_{ij}/m_i$, this equation can be rewritten as

$$\sum_{j \neq i} \psi_{ij} = \frac{1}{m_i} \sum_{j \neq i} w_{ij} = (C-2) + \frac{m_{ii}}{m_i}. \quad (30)$$

Recalling that the term m_{ii}/m_i is the True Positive Rate of the i-th class (TPR_i), (30) can finally be expressed as

$$\sum_{j \neq i} \psi_{ij} = C - 2 + TPR_i. \quad (31)$$

V. DISCUSSION

A. IMBALANCED CONFUSION STAR AND GEAR

To obtain the balanced confusion star (Fig. 22) and gear (Fig. 23), the circle was divided into C equal-sized regions. A different *imbalanced* approach is also possible using regions whose sizes are proportional to the number of instances belonging to each class. The region corresponding to the i-th class spans an angle of

$$\beta_i = 2\pi \frac{m_i}{m}, \quad (32)$$

and the angular separation between two radii is

$$\Delta\varphi_i = \frac{2\pi m_i}{m(C-1)}. \quad (33)$$

As the classes in the MNIST dataset are barely imbalanced, the reduced Abalone dataset is used in this case. The balanced confusion star is depicted in Fig. 24, while the corresponding imbalanced version is shown in Fig. 25.

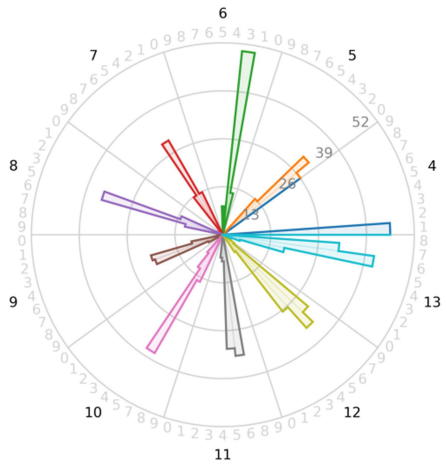

FIGURE 24. Balanced confusion star corresponding to the classification of the reduced Abalone dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

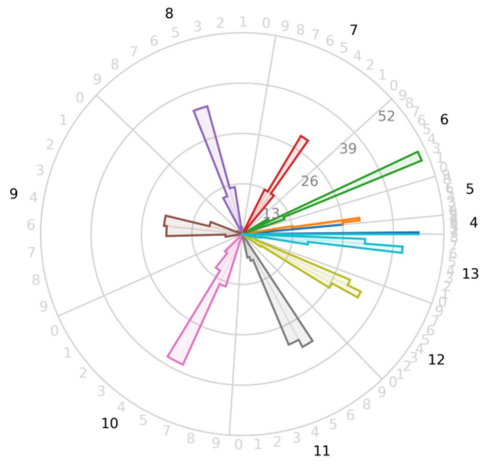

FIGURE 25. Imbalanced confusion star corresponding to the classification of the reduced Abalone dataset.

In the imbalanced star it can be noted that, for example, the region corresponding to class 9 (with 138 instances) is remarkably wider than that corresponding to class 4 (with 11 instances).

B. AREAS OF THE CONFUSION STAR AND GEAR

Intuitively it can be seen that the area enclosed by the confusion star is a metric of the classifier's performance: the larger the area, the worse the classifier. The opposite statement can be affirmed for the confusion gear: the larger the area, the better the classifier. So, analyzing these areas can be useful as their use as alternative classification metrics.

Let us consider a balanced confusion star where the enclosed area is the sum of the area of each sector A_{ij}, that is,

$$A = \sum_{i=1}^{C}\left(\sum_{j\neq i} A_{ij}\right) = \sum_{i=1}^{C}\left(\sum_{j\neq i}(\Delta\varphi \cdot \epsilon_{ij})\right). \tag{34}$$

Recalling (25)

$$A = \sum_{i=1}^{C}\left(\sum_{j\neq i}\left(\frac{2\pi}{C\cdot(C-1)}\cdot \epsilon_{ij}\right)\right). \tag{35}$$

$$A = \sum_{i=1}^{C}\left(\frac{2\pi}{C\cdot(C-1)}\sum_{j\neq i}\epsilon_{ij}\right). \tag{36}$$

Considering (20)

$$A = \frac{2\pi}{(C-1)}\cdot\frac{1}{C}\sum_{i=1}^{C} FNR_i = \frac{2\pi}{C-1}FNR. \tag{37}$$

The ratio of this area to the total area of the circle is called the Internal Area Ratio (IAR) and is defined as

$$IAR \equiv \frac{A}{2\pi} = \frac{FNR}{C-1}, \tag{38}$$

that is, a value proportional to the multiclass miss rate (FNR). For binary classification ($C = 2$), $IAR = FNR$.

Focusing on the area outside the confusion star, an analogous External Area Ratio (EAR) can be defined as

$$EAR \equiv 1 - IAR = 1 - \frac{FNR}{C-1} = \frac{C-1-FNR}{C-1}. \tag{39}$$

Recalling that $FNR = 1 - TPR$ it can be written that

$$EAR = \frac{C-1-(1-TPR)}{C-1} = \frac{C-2+TPR}{C-1}. \tag{40}$$

Considering now the imbalanced confusion star, the enclosed area is

$$A = \sum_{i=1}^{C}\left(\sum_{j\neq i} A_{ij}\right) = \sum_{i=1}^{C}\left(\sum_{j\neq i}(\Delta\varphi_i \cdot \epsilon_{ij})\right). \tag{41}$$

Recalling (33)

$$A = \sum_{i=1}^{C}\left(\sum_{j\neq i}\left(\frac{2\pi m_i}{m(C-1)}\cdot \epsilon_{ij}\right)\right). \tag{42}$$

$$A = \sum_{i=1}^{C}\left(\frac{2\pi}{m(C-1)}\sum_{j\neq i} m_i\epsilon_{ij}\right). \tag{43}$$

Since $e_{ij} = m_i\epsilon_{ij}$, this equation can be rewritten as

$$A = \frac{2\pi}{m(C-1)}\sum_{i=1}^{C}\sum_{j\neq i} e_{ij}. \tag{44}$$

Recalling (18)

$$A = \frac{2\pi}{C-1}\cdot\frac{1}{m}\sum_{i=1}^{C}(m_i - m_{ii}). \tag{45}$$

$$A = \frac{2\pi}{C-1}\left(\frac{1}{m}\sum_{i=1}^{C} m_i - \frac{1}{m}\sum_{i=1}^{C} m_{ii}\right). \tag{46}$$

Since $\sum_{i=1}^{C} m_i = m$,

$$A = \frac{2\pi}{C-1}\left(1 - \frac{1}{m}\sum_{i=1}^{C} m_{ii}\right). \tag{47}$$

Two of the most common classification performance metrics are the accuracy, defined as

$$ACC \equiv \frac{1}{m}\sum_{i=1}^{C} m_{ii}, \tag{48}$$

and the error rate $ER \equiv 1 - ACC$. Substituting these expressions in (47) yields

$$A = \frac{2\pi}{C-1}(1 - ACC) = \frac{2\pi}{C-1}ER. \tag{49}$$

The Internal Area Ratio (IAR) is then

$$IAR \equiv \frac{A}{2\pi} = \frac{ER}{C-1}, \tag{50}$$

that is, a value proportional to the multiclass error rate (ER). For binary classification ($C = 2$), $IAR = ER$.

TABLE III
SUMMARY OF AREAS: INTERNAL AND EXTERNAL AREA RATIOS FOR THE CONFUSION STAR AND GEAR.

	Star		Gear	
	Balanced	***Imbalanced***	***Balanced***	***Imbalanced***
IAR	$\frac{FNR}{C-1}$	$\frac{ER}{C-1}$	$\frac{C-2+TPR}{C-1}$	$\frac{C-2+ACC}{C-1}$
EAR	$\frac{C-2+TPR}{C-1}$	$\frac{C-2+ACC}{C-1}$	$\frac{FNR}{C-1}$	$\frac{ER}{C-1}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

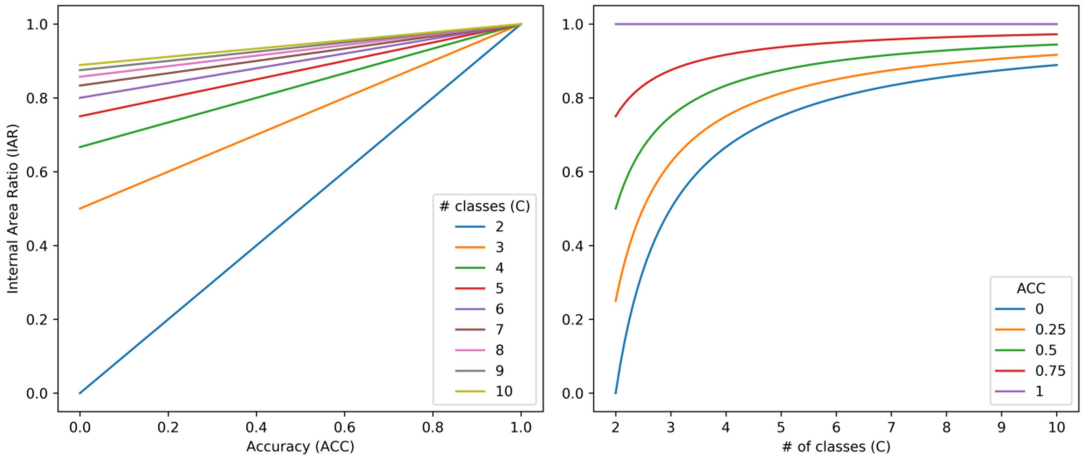

FIGURE 26. Relationship between the Internal Area Ratio (IAR) and the accuracy (ACC) for the imbalanced confusion gear: linear dependency for a single dataset (left) and nonlinear relation for different datasets (variable number of classes, right).

Regarding now the confusion gear, similar expressions can be derived for its internal and external areas. All these results are summarized in TABLE III. From the previous results it can be seen that the areas in the confusion star and gear, are directly related to classical performance metrics. For example, the imbalanced confusion gear has an internal area linearly proportional to the accuracy (ACC), while the external area is linearly proportional to the error rate. So these areas can be considered a visual representation of the classification performance.

However, the relation among areas and classical metrics has to be carefully considered. While this relation is linear for a certain dataset (a constant number of classes C), it becomes nonlinear if the classification performance is analyzed through different datasets. The relationship between the IAR and the ACC for the imbalanced confusion gear is depicted in Fig. 26, both for a single dataset (left) and for different datasets (right).

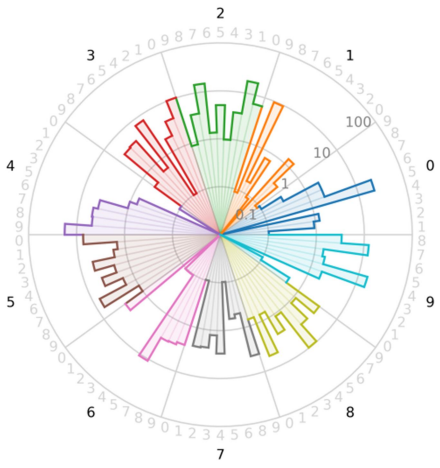

FIGURE 27. Balanced confusion star corresponding to the classification of the MNIST testing dataset (logarithmic scale). The radii values 0.1, 1, 10 and 100, correspond to percentage of errors in logarithmic scale.

C. LOGARITHMIC CONFUSION STAR

Both the balanced (Fig. 22) and the imbalanced (Fig. 24) confusion stars do not properly visualize the values of the error matrix when they are very small. To overcome this problem and to accommodate in a single graphic very different error values, the length of the radii are made proportional to the logarithm of the errors. The result obtained using this procedure is depicted in Fig. 27.

In this graphic the center of the circle does not correspond to a null error but to an arbitrarily chosen small value (0.01 in the graphic).

In general, hit matrices do not have very small values (usually greater than 50%), so the use of the logarithmic scale is not required.

D. CONFUSION STARS FOR MANY CLASSES

As the number of classes increases, any graphic representation of the confusion matrix becomes less clear. For instance, the colored grid corresponding to the classification of the CIFAR-100 dataset is shown in Fig. 28. In that graphic is very difficult to identify in which classes the classifier is underperforming and should be improved.

If the classifier performance is visualized using the confusion star, the result is depicted in Fig. 29. In this plot is easier to identify that, for example, the classifier is having problems to correctly identify instances of class 47 and 52. Therefore, although the confusion star becomes less clear as the number classes increases, it is a better representation than the classical colored grid.

E. SEQUENCE OF CONFUSION STARS

Following the evolution of a certain feature or metric is a common task in science and engineering [40]. In the field of classification algorithms there are some applications where it is convenient to visualize the performance, not of a single classifier, but of a sequence of classifiers, comparing their results depending on the value of a certain parameter or hyperparameter. Even some tools has been proposed to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

visualize the evolution of the classification process either at the instance level [36] or the confusion matrix level [41].

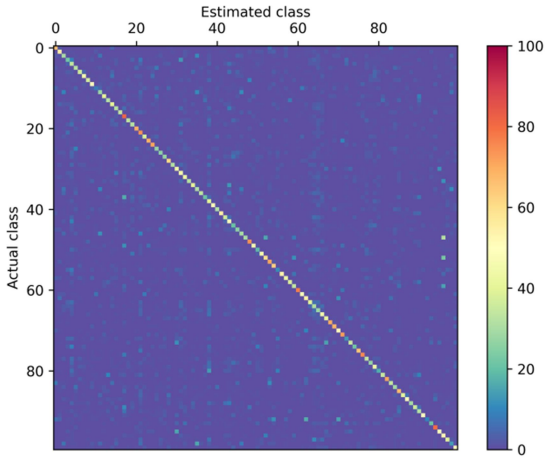

FIGURE 28. Colored grid corresponding to the classification of the CIFAR-100 dataset.

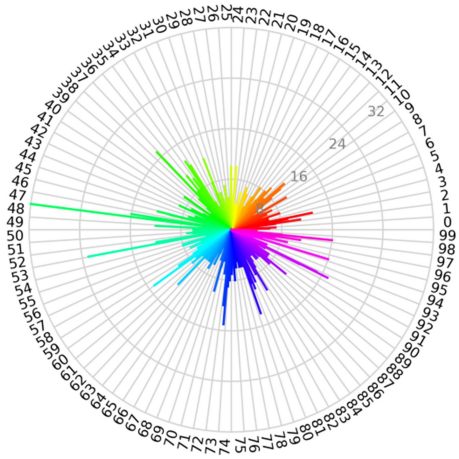

FIGURE 29. Confusion star corresponding to the classification of the CIFAR-100 dataset.

Also the confusion stars and gears can be used for this purpose. To show how it can be done let us consider again the MNIST dataset and the same neural network classifier with a single hidden layer and a sigmoid as the activation function. For this analysis the number of neurons in the hidden layer is increased from 8 to 128 and the number of training epochs rises from 5 to 100. The objective of these improvements is to obtain a wider range of classification performances.

To determine the impact of the number of training instances on the classification performance, a variable number of instances to train the network are used, observing the accuracy of the classification in each case. The result is usually known as the *learning curve*, depicted in Fig. 30.

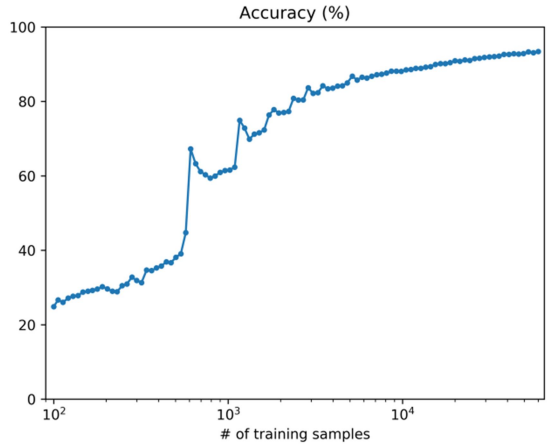

FIGURE 30. Learning curve of the MNIST dataset using a neural network with a single 128-neurons hidden layer.

This representation properly summarizes the performance of a classifier in a single metric, the accuracy in this example. However, it is possible to exploit the descriptive power of the confusion star for a better and more detailed insight of the evolution of the classification performance. Indeed, each dot in the learning curve has a corresponding confusion matrix that can be properly visualized as a confusion star.

Let us consider, for example, the significant increasing in the accuracy occurring around 500 training instances. While the learning curve does not detail what this improvement is due to or how it is distributed in each of the classes, an analysis of the confusion stars in accuracy, before and after the jump, can shed more light on the question. In Fig. 31, the confusion stars corresponding to a point with 502 samples (before the jump, accuracy of 38%) and another point with 610 samples (after the jump, accuracy of 67%) are shown. Quite important improvements (smaller errors) can be observed in, for example, the 0 classified as a 2, the 2 classified as a 1, and so on. In other words, the representation of the confusion matrix not only informs us of the *overall improvement* of the classifier, but also of *how this improvement is distributed*.

A similar representation can also be obtained using the confusion gear.

The application of the confusion stars to compare two points of the learning curve can be extended to a sequence of points, drawing a grid of stars as it is shown in Fig. 32. In that graphic, which resembles the concept of as small multiple [42], can be seen that, for example, the problems classifying instances of classes 4, 8 and 9 that shows the classifiers trained with up to 1000 training instances, are mostly solved once the 3000 instances barrier is overcome. From this point on, a smooth and continuous improvement of the classification results is obtained. The same information can be obtained analyzing the corresponding confusion gears.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

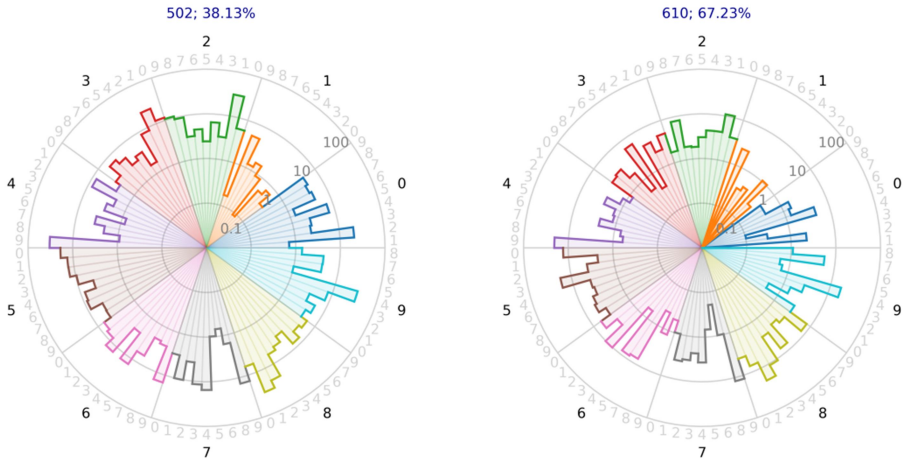

FIGURE 31. **Confusion stars (in logarithmic scale) corresponding to a pair of points before and after the first jump in the learning curve: 502 instances (left; accuracy of 38.13%) and 610 instances (right; accuracy of 67.23%).**

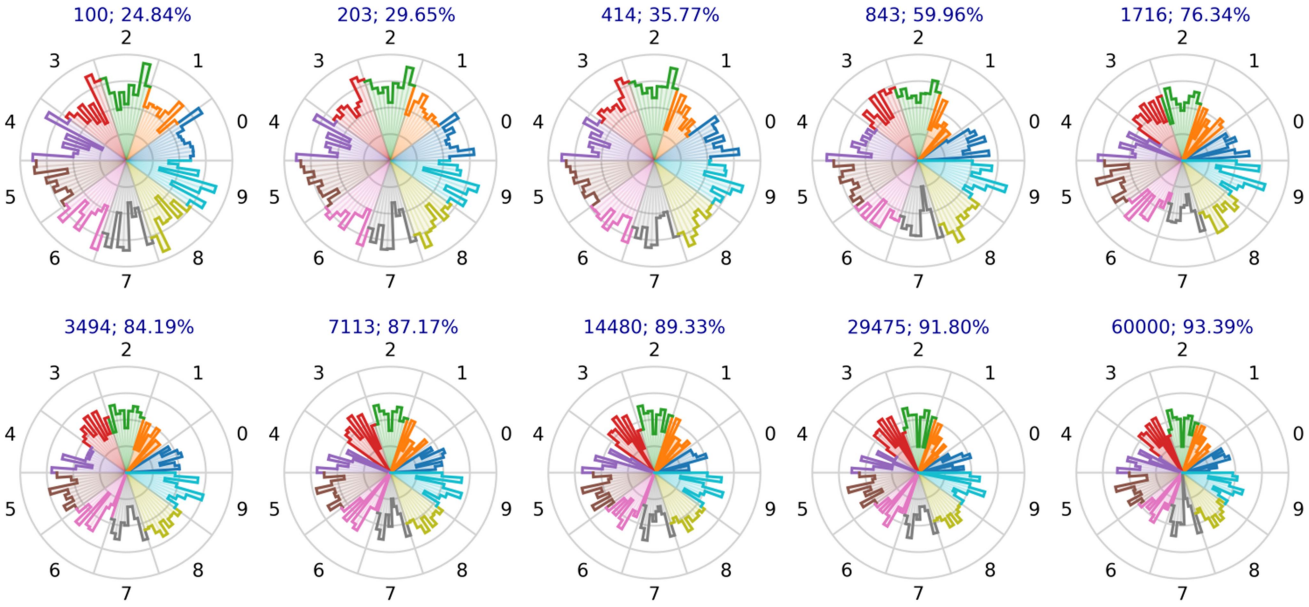

FIGURE 32. **Sequence of logarithmic confusion stars corresponding to several point of the learning curve.**

Representing a sequence of confusion matrices by a grid of stars has an obvious limitation of space: the more matrices to be represented, the smaller is the size of each star. To tackle this problem, the sequence of confusion matrices can be represented generating a movie where each frame corresponds to a single confusion matrix. An example of this video can be seen in the online version of the paper (see also appendix). In Fig. 33, an example of a frame of the movie is shown.

F. SUMMARY OF VISUALIZATION METHODS

Through the paper, up to 13 methods for visualizing classification performance have been described. Some of them focus on classification scores of single instances while others are interested on how the classifiers behave for the instances of certain classes. On the other hand, some visualization methods are designed primarily for two classes (binary classification) while others can represent multiple classes.

Visualization methods can be featured by how they represent the different classes (actual or estimated) and the classification performance. Some of them use color to convey the required information while others use geometric elements for this purpose: X and/or Y axis position in rectangular plots, radial and/or angular position in polar plots, length and/or width of graphical elements, etc.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3137630, IEEE Access

TABLE IV. SUMMARY OF VISUALIZATION METHODS

Visualization method	Fig.	Element	Multiple classes	Representation of…		Comments
				Classes	Classif. performance	
Linear scores	1	Instance	Yes	X-axis position	Y-axis position	For a single instance or for a very reduced number of them
Polar scores	2	Instance	Yes	Color	Polar position	For many instances (detailed information)
Box-plot scores	3	Instance	Yes	Position in a matrix of graphics (actual class). X-axis position (estimated class)	Box-plot with the statistical distribution	For many instances (statistical information)
Colored grid	4, 5	Class	Yes	Cell position	Color	Straightforward color-based representation of a confusion matrix
Stacked bar	6	Class	Yes	X-Y position	Stack length	Length-based representation of a confusion matrix
Binary colored matrices	7	Class	No	Position in a matrix of graphics (actual class).	Color	Classification results of each class (color-based representation)
Binary stacked bars	8	Class	No	Position in a matrix of graphics (actual class).	Stack length	Classification results of each class (length-based representation)
Binary ROC curves	9	Class	No	Color	Form of a curve	Classification results of each class with different decision thresholds
Chord diagram	10	Class	Yes	Color and angular position	Width of a chord	Polar width-based representation of a confusion matrix
Sankey diagram	11	Class	Yes	Color and linear position	Width of a ribbon	Linear width-based representation of a confusion matrix
Multidimensional scaling	12	Class	Yes	Color	Distance among classes	Similarity among classes as they are seen by the classifier
Confusion star	22	Class	Yes	Color and angular position	Radial position	Polar length-based representation of a confusion matrix (focus on errors)
Confusion gear	23	Class	Yes	Color and angular position	Radial position	Polar length-based representation of a confusion matrix (focus on hits)

A summary of the visualization methods described in the paper is shown in TABLE IV.

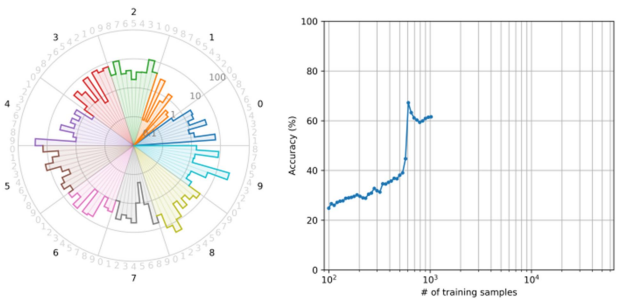

FIGURE 33. Frame of the movie generated to visualize the learning process: learning curve drawn up to 1000 training instances (right), and its corresponding confusion star in logarithmic scale (left).

VI. CONCLUSION

This paper has reviewed several methods to visualize classification results at different levels of detail: from those centered on how a particular instance or set of instances are classified, to those that summarize the classification performance in a single metric.

A particular interest has been devoted to classification results which are summarized in the form of a confusion matrix, presenting the main procedures to visualize it from the straightforward row-column matrix representation, with colors indicating the value of each matrix cell, to more complex and sophisticated graphics.

From this analysis, a new way of representing the information conveyed by confusion matrices is proposed in the form of a confusion star (focusing on the errors) or a confusion gear (centered on the hits). The new visualization tool can be employed to represent the original and possibly imbalanced confusion matrix, or the balanced unit version of that matrix.

The new tool successfully represents multiclass classification results in the form of a radial plot. The traditional way to represent confusion matrix uses *colors* (and eventually texts) to indicate the number of instances belonging to an actual class that are classified to an estimated class. Instead, confusion stars and gears use *shapes* to convey that information. Changing colors by shapes significantly improves the readability of the proposed graphics.

An additional property of the confusion stars and gears is that *the enclosed area provides information about the overall classification performance*. The relation of these areas to standard classification metrics has also been derived.

Finally, it has also been shown that the new graphic tools can usefully be employed to visualize the performances of a sequence of classifiers.

APPENDIX

Supplementary materials can be found in the on line version of the paper or they and can also be downloaded from https://github.com/amalialuque/confusionstar. They contain:

1) Three Excel files with the confusion matrices described in Section II.A.
2) An Excel file with the sequence of confusion matrices described in Section V.E.
3) A video file (in Graphics Interchange Format, GIF, format) visualizing the learning process described in Section V.E.
4) A Jupyter notebook, providing an implementation of the functions required to plot a confusion matrix as a confusion star (or confusion gears); and to generate a video file visualizing a sequence of confusion matrices in the form of confusion stars (or confusion gears).

Additionally, the algorithm that converts a confusion matrix into a confusion star plot can be found as supplementary material to the paper.

REFERENCES

[1] Fan-Yin Tzeng and Kwan-Liu Ma, "Opening the Black Box - Data Driven Visualization of Neural Networks," in *VIS 05. IEEE Visualization.*, 2005, pp. 383–390, doi: 10.1109/visual.2005.1532820.

[2] K. Wongsuphasawat *et al.*, "Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 1–12, 2018, doi: 10.1109/TVCG.2017.2744878.

[3] Y. Ming *et al.*, "Understanding Hidden Memories of Recurrent Neural Networks," in *2017 IEEE Conference on Visual Analytics Science and Technology, VAST 2017 - Proceedings*, 2018, pp. 13–24, doi: 10.1109/VAST.2017.8585721.

[4] A. W. Harley, "An interactive node-link visualization of convolutional neural networks," in *International Symposium on Visual Computing, ISVC 2015*, 2015, vol. 9474, pp. 867–877, doi: 10.1007/978-3-319-27857-5_77.

[5] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 618–626, 2017, doi: 10.1109/ICCV.2017.74.

[6] J. Wang, L. Gou, H. Yang, and H. W. Shen, "GANViz: A Visual Analytics Approach to Understand the Adversarial Game," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 6, pp. 1905–1917, 2018, doi: 10.1109/TVCG.2018.2816223.

[7] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg, "Embedding projector: Interactive visualization and interpretation of embeddings," in *Proc. Neural Inf. Process. Syst. Workshop Interpretable ML Complex Syst*, 2016.

[8] D. Chen, R. K. E. Bellamy, P. K. Malkin, and T. Erickson, "Diagnostic visualization for non-expert machine learning practitioners: A design study," in *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 2016, pp. 87–95, doi: 10.1109/VLHCC.2016.7739669.

[9] A. Bauerle, C. Van Onzenoodt, and T. Ropinski, "Net2Vis-A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 6, pp. 2980–2991, 2021, doi: 10.1109/TVCG.2021.3057483.

[10] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg, "Direct-Manipulation Visualization of Deep Networks," in *Proc. ICML Workshop Vis. Deep Learn.*, 2016.

[11] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, 2014, vol. 8689 LNCS, no. PART 1, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.

[12] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Proc. ICML Deep Learn. Workshop*, 2015.

[13] D. Cashman, G. Patterson, A. Mosca, N. Watts, S. Robinson, and R. Chang, "RNNbow: Visualizing Learning Via Backpropagation Gradients in RNNs," *IEEE Comput. Graph. Appl.*, vol. 38, no. 6, pp. 39–50, 2018, doi: 10.1109/MCG.2018.2878902.

[14] B. Alsallakh, A. Hanbury, H. Hauser, S. Miksch, and A. Rauber, "Visual methods for analyzing probabilistic classification data," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1703–1712, 2014, doi: 10.1109/TVCG.2014.2346660.

[15] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 667–676, 2018, doi: 10.1109/TVCG.2017.2744158.

[16] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. F. Lelieveldt, E. Eisemann, and A. Vilanova, "DeepEyes: Progressive Visual Analytics for Designing Deep Neural Networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 98–108, 2018, doi: 10.1109/TVCG.2017.2744358.

[17] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 8, pp. 2674–2693, 2019, doi: 10.1109/TVCG.2018.2843369.

[18] S. Liu, X. Wang, M. Liu, and J. Zhu, "Towards better analysis of machine learning models: A visual analytics perspective," *Vis. Informatics*, vol. 1, no. 1, pp. 48–56, 2017, doi: 10.1016/j.visinf.2017.01.006.

[19] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. P. Chau, "ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 88–97, 2018, doi: 10.1109/TVCG.2017.2744718.

[20] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 61–70, 2017, doi: 10.1109/TVCG.2016.2598828.

[21] A. P. Norton and Y. Qi, "Adversarial-Playground: A visualization suite showing how adversarial examples fool deep learning," in *2017 IEEE Symposium on Visualization for Cyber Security, VizSec 2017*, 2017, pp. 1–4, doi: 10.1109/VIZSEC.2017.8062202.

[22] R. E. Curtis, A. Yuen, L. Song, A. Goyal, and E. P. Xing, "TVNViewer: An interactive visualization tool for exploring networks that change over time or space," *Bioinformatics*, vol. 27, no. 13, pp. 1880–1881, 2011, doi: 10.1093/bioinformatics/btr273.

[23] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan, "EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers," in *Conference on Human Factors in Computing Systems - Proceedings*, 2009, pp. 1283–1292, doi: 10.1145/1518701.1518895.

[24] D. Dua and C. Graff, "UCI Machine Learning Repository," 2019. [Online]. Available: https://archive.ics.uci.edu/ml. [Accessed: 31-May-2021].

[25] J. Lever, M. Krzywinski, and N. Altman, "Classification evaluation," *Nat. Methods*, vol. 13, no. 8, pp. 603–604, 2016, doi: 10.1038/nmeth.3945.

[26] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit.*, vol. 91, pp. 216–231, 2019, doi: 10.1016/j.patcog.2019.02.023.
[27] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, 2012, doi: 10.1109/MSP.2012.2211477.
[28] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, 2020, doi: 10.1109/TPAMI.2019.2913372.
[29] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images.(2009)," *Cs.Toronto.Edu*, pp. 1–58, 2009.
[30] P. Madhusudhan, "CIFAR100 small image classification keras dataset," 2018. [Online]. Available: https://phani1995.github.io/artificial/neural/network/2018/10/31/CIFAR100_small_image_classification_keras_dataset.html. [Accessed: 18-Nov-2021].
[31] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-Aware Minimization for Efficiently Improving Generalization," in *International Conference on Learning Representations*, 2020.
[32] W. J. Nash, T. L. Sellers, S. R. Talbot, A. J. Cawthorn, and W. B. Ford, "The Population Biology of Abalone (Haliotis species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait," *Sea Fish. Div. Tech. Report, 48*, 1994.
[33] C. Seifert and E. Lex, "A novel visualization approach for data-mining-related classification," in *Proceedings of the International Conference on Information Visualisation*, 2009, pp. 490–495, doi: 10.1109/IV.2009.45.
[34] E. Beauxis-Aussalet and L. Hardman, "Visualization of Confusion Matrix for Non-Expert Users," in *EEE Conference on Visual Analytics Science and Technology (VAST)-Poster Proceedings*, 2014.
[35] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.
[36] M. Pühringer, A. Hinterreiter, and M. Streit, "InstanceFlow: Visualizing the Evolution of Classifier Confusion on the Instance Level," in *Proceedings - 2020 IEEE Visualization Conference, VIS 2020*, 2020, pp. 291–295, doi: 10.1109/VIS47514.2020.00065.
[37] R. Susmaga, "Confusion Matrix Visualization," in *Intelligent Information Processing and Web Mining*, Springer Berlin Heidelberg, 2004, pp. 107–116.
[38] Y. Zhou, T. Wischgoll, L. M. Blaha, R. Smith, and R. J. Vickery, "Visualizing confusion matrices for multidimensional signal detection correlational methods," in *Visualization and Data Analysis 2014*, vol. 9017, no. 3, p. 901709, doi: 10.1117/12.2042610.
[39] B. Diri and S. Albayrak, "Visualization and analysis of classifiers performance in multi-class medical data," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 628–634, 2008, doi: 10.1016/j.eswa.2006.10.016.
[40] S. Dutta, T. L. Turton, and J. P. Ahrens, "A Confidence-Guided Technique for Tracking Time-Varying Features," *Comput. Sci. Eng.*, vol. 23, no. 2, pp. 84–92, 2021, doi: 10.1109/MCSE.2020.3047979.
[41] A. Hinterreiter *et al.*, "ConfusionFlow: A model-agnostic visualization for temporal analysis of classifier confusion," *IEEE Trans. Vis. Comput. Graph.*, pp. 1–1, 2020, doi: 10.1109/tvcg.2020.3012063.
[42] E. Tufte, N. Goeler, and R. Benson, *Envisioning information (Vol. 2)*. Cheshire, CT: Graphics press, 1990.

AMALIA LUQUE (M'21) received her Industrial Engineering degree in 2007, Master´s in Automation, Robotics and Telematics in 2010 and an Industrial Engineering Doctoral Degree in 2014. She has been involved in teaching related to Project Engineering at the University of Seville since 2015, where she is currently a Professor. Her main areas of research are control, business intelligence, data mining, feature extraction and artificial intelligence.

MIRKO MAZZOLENI (M'18) was born in Ponte San Pietro, Italy, in 1989. He received the M.Sc. degree (summa cum laude) in computer engineering and the Ph.D. degree in engineering and applied sciences (control systems) from the University of Bergamo, Italy, in 2014 and 2018, respectively. He is currently an Assistant Professor with the University of Bergamo. His main research interests include system identification, machine learning, and fault diagnosis.

ALEJANDRO CARRASCO (M'07) received his Computer Engineering degree in 1998 and his Ph.D. in Computer Engineering in 2003 from the University of Seville (Spain). Since 1997, he has worked for several companies in the area of software engineering and computer networks, has founded a new technology-based company (NTBF) and he has actively participated and directed several R&D projects. Currently Dr. Carrasco is a Lecturer and Researcher at the Department of Electronic Technology at the University of Seville, and his research activities are focused on data mining, distributed services and artificial intelligence applied to industrial computing and cybersecurity.

ANTONIO FERRAMOSCA was born in Maglie (LE), Italy, in 1982. He received the Graduate degree in computer science engineering from the University of Pavia, Pavia, Italy, in 2006, and the Ph.D. degree in engineering, with full marks and honors (summa cum laude), from the University of Seville, Seville, Spain, in 2011. He is currently Assistant Professor with the Department of Management, Information and Production Engineering, University of Bergamo, Italy. He was a Visiting Researcher with the University of Wisconsin-Madison, Madison, WI, USA, in 2009, with the Institute of Technological Development for the Chemical Industry, Santa Fe, Argentina, in 2010, and with the University of Seville in 2015. He is author and co-author of more than 90 publications including book chapters, journal papers, conference papers, and industrial deliverables. His research interests include model predictive control, nonlinear systems, impulsive systems, distributed control, stability, control of biological systems, and robust control.