Robustness assessment and improvement of a neural network for blood oxygen pressure estimation

Paolo Arcaini
National Institute of Informatics
Tokyo, Japan
arcaini@nii.ac.jp

Andrea Bombarda
University of Bergamo
Bergamo, Italy
andrea.bombarda@unibg.it

Silvia Bonfanti
University of Bergamo
Bergamo, Italy
silvia.bonfanti@unibg.it

Angelo Gargantini
University of Bergamo
Bergamo, Italy
angelo.gargantini@unibg.it

Daniele Gamba
AISent S.r.l.
Dalmine, Italy
daniele@aisent.io

Rita Pedercini
AISent S.r.l.
Dalmine, Italy
rita.pedercini@aisent.io

***Abstract*—Neural networks have been widely applied for performing tasks in critical domains, such as, for example, the medical domain; their robustness is, therefore, important to be guaranteed. In this paper, we propose a robustness definition for neural networks used for regression, by tackling some of the problems of existing robustness definitions. First of all, by following recent works done for classification problems, we propose to define the robustness of networks used for regression w.r.t. alterations of their input data that can happen in reality. Since different alteration levels are not always equally probable, the robustness definition is parameterized with the probability distribution of the alterations. The error done by this type of networks is quantifiable as the difference between the estimated value and the expected value; since not all the errors are equally critical, the robustness definition is also parameterized with a "tolerance" function that specifies how the error is tolerated. The current work has been motivated by the collaboration with the industrial partner that has implemented a medical sensor employing a Multilayer Perceptron for the estimation of the blood oxygen pressure. After having computed the robustness for the case study, we have successfully applied three techniques to improve the network robustness: data augmentation with recombined data, data augmentation with altered data, and incremental learning. All the techniques have proved to contribute to increasing the robustness, though in different ways.**

I. Introduction

Neural Networks (NNs) are increasingly adopted to perform different complex activities [1], such as recognition, control, decision making, etc. Often, they are employed in safety-critical domains [10], such as in the medical practice [13]. A desired property of an NN is its *robustness*, i.e., the ability of the network to correctly process unknown (i.e, not seen during training) inputs. NN robustness is usually defined and computed by using *adversarial* examples, i.e., inputs that are particularly challenging for the network under test, and that are created by exploiting the network internal structure [20]. However, different works [16], [17], [25] have remarked that, due to their origin, they may not reflect real inputs that could occur during the network usage. Our position is that a meaningful robustness measure should take into account the plausible alterations that are specific to the application domain. Therefore, in [2], we have proposed to define the robustness of an NN used for classification, by considering *real alterations* that may occur to input data. The definition is independent of the input type, and so it applies to different input types, such as digital images, audio, and text [4].

Since the previous definition of robustness is ed to *classifiers*, it is not suitable for networks used for regression analysis and *estimation*, like the network whose robustness we were asked to assess by our industrial partner. The error done by a classifier is not subject to a measure, and only false negatives and false positives are considered errors when computing robustness. For regression, instead, the error can be actually *measured*, and different levels of errors influence the network robustness level. For instance, a network that always guarantees a small percentage error (like 5%) may be considered more robust than a network with a greater percentage error (like 10%); indeed, a user may tolerate more a smaller error than a larger error. For this reason, one objective of this work is to introduce a suitable measure to assess and weigh the error done by regression networks.

Another limit of previous works is that different alteration levels (see Sect. III-A) are all equally considered in the robustness computation. However, after some initial meetings with the industrial partner, it was apparent that some alterations are more likely to occur for some levels than for others. For example, considering the signal acquisition, lower values of clock offset are more likely to occur than the higher ones.

Thus, to tackle the limits of the previous works, in this paper we propose a novel definition of robustness of NNs used for regression, that takes into account both the tolerance to different levels of errors and the occurrence probability of a given alteration level.

The analysis reported in this paper is applied to a Multilayer Perceptron (MLP), but it is applicable to all kinds of NNs

P. Arcaini is supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST, and Engineerable AI Techniques for Practical Applications of High-Quality Machine Learning-based Systems Project (Grant Number JPMJMI20B8), JST-Mirai.

used for regression and estimation. MLPs are feed-forward artificial neural networks generating a set of outputs from a set of inputs. They are the simplest example of Artificial Neural Network (ANN), where each neuron is fully connected to those in the layer below and has a non-linear activation function. They are networks composed of at least three layers: an input layer, one or more hidden layers, and an output layer. MLPs are widely used since they can be seen as universal function approximators [8] and can be used to create mathematical models for regression.

The current work has originated from the collaboration with our industrial partner AISent[1], which researches and implements AI solutions. The partner required us to analyze an MLP that is used for the estimation of the partial pressure of oxygen of the blood flowing in a sensor. The company wanted to gain confidence in the reliability of the model, in particular when it is subject to the alterations that can occur during its typical working condition [12]. We have investigated different formulas for robustness computation, in order to choose the most suited one for the considered application. Moreover, since the system is used in safety-critical procedures and it is expected to be as much robust as possible, we have examined three methods for improving the robustness of the MLP under analysis: two methods are based on retraining using datasets enriched with additional data (obtained in different ways), and another method uses a new network, which is trained on new data and applied in parallel with the original one.

The paper is structured as follows. Sect. II presents the case study on which we have conducted our analysis, while Sect. III introduces the novel definition of robustness based on the notions of tolerance and alteration level probability. Sect. IV presents the alterations we have considered, how the robustness definition has been implemented in practice and discusses the obtained results. Sect. V presents methods suitable for enhancing the robustness of a multilayer perceptron. Sect. VI presents a more critical discussion of the work done and reports the feedback provided by the industrial partner. Finally, Sect. VII reviews works related to our approach, and Sect. VIII concludes the paper.

II. Case Study

In medical practice, constantly assessing the right value of the partial pressure of oxygen (pO_2) of the blood is very critical, especially during surgery. One can derive the pO_2 level by observing the blood fluorescence. When exposed to a bright pulse, the blood responses with a fluorescence that can be described (or better "approximated") by a biexponential function defined as follows:

$$fluorescence(t) = A \cdot (e^{-B_1 t} - e^{-B_2 t}) \quad (1)$$

being A, B_1, and B_2 parameters that characterize the response, and t the time passed from the instant of the light pulse. The parameters A, B_1, and B_2 depend on the current level of pO_2 and on the blood temperature. An example of this curve (experimentally taken) is shown in Fig. 1. Using a spotlight

[1]https://aisent.io/en/

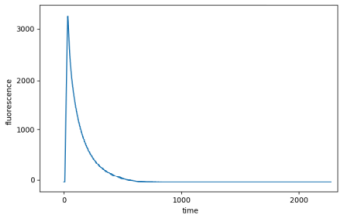


Fig. 1. Blood fluorescence in response to a spotlight pulse

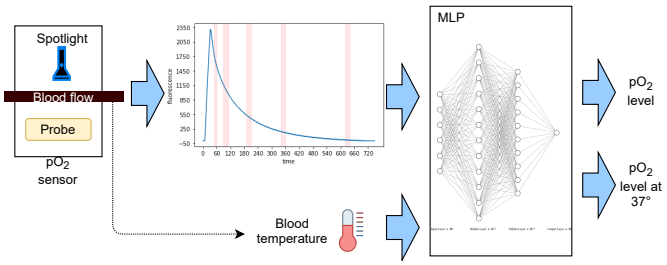


Fig. 2. Overview of the MLP-based sensor for pO_2 estimation

to illuminate the blood and a probe to measure the response to the bright pulse, one could try to estimate the parameters A, B_1, and B_2 and then produce an estimation of the pO_2. However, finding the best biexponential curve (i.e., finding the fittest parameter values), is very challenging and it proved to be unfeasible by the microcontroller that the industrial partner had chosen to use in the sensor.

For this reason, the company decided to deploy on the microcontroller an MLP trained for the estimation of pO_2. An overview of the MLP-based sensor is shown in Fig. 2.

The MLP model takes as input a limited number of samples of blood fluorescence obtained in response to the bright pulse and the blood temperature, to estimate the values of the pO_2 at two temperatures: at the current temperature and at 37 °C. In particular, the MLP uses the means values of the curve extracted in the intervals $[50, 60]$, $[90, 110]$, $[190, 210]$, $[340, 360]$ and $[620, 640]$. It is composed of 6 neurons in the input layer, 12 neurons in the first hidden layer, 10 neurons in the second hidden layer, and 2 neurons in the output layer, giving as output the estimation of the pO_2 value at the current temperature and the prediction of pO_2 value at 37 °C (see Fig. 2). All the neurons in the layers use sigmoid activation functions. To train, validate, and test the MLP, the company used a dataset composed of 21,650 curves similar to the one in Fig. 1. These samples come from 16 different types of probes and 178 different spotlights. For each input sample, the true values of the pO_2 at the two temperatures (i.e., current and 37 °C) were given by an analysis of the blood using a precision measurement instrument. As usual, 60% of the dataset has been used for the training phase, 20% for the validation phase, and 20% for the testing phase.

A. Research Objective

At the beginning of the project, several meetings were held with the industrial partner to define how to measure robustness,

i.e., the ability of the network to correctly evaluate slightly altered inputs.

As a first step, we have identified domain-specific alterations that can affect the network during its operation.

Regarding the robustness, since the industrial partner works on different domains (in addition to the medical domain), we realized that there is no single robustness definition that is suitable for all application domains. Indeed, the effect of the network error (i.e., the difference between the real value and the estimated value) is different in different applications. For example, in some application domains, any level of error is detrimental to the system, and so the robustness definition should reflect that the error must be kept as small as possible; considering the clock offset alteration of the case study, an estimation error of 4% is less acceptable compared to an error of 2%. In some other application domains, instead, the system operation may not be affected by errors up to a given level (e.g., the precision of the system embedding the network) and so the robustness definition should reflect this. In order to account for these different application domains, we decided to define the robustness in a "parameterized" way, by allowing to specify the type of *tolerance* to the error.

In a similar way, given an alteration type, it could be that some alteration levels are more probable than others. For example, if we consider data acquisition using electronic probes, certain levels of clock offset (e.g., $20\,ms$) are more probable than others (e.g., $40\,ms$). In some application domains, a designer may accept some errors when they are due to very rare alterations, while, in other domains, errors should be considered equally, independently of the probability of the alteration causing them. So, we chose to parameterize the robustness definition by including how the *probability* of an alteration level should be considered.

Sect. III introduces the robustness measure that considers the error tolerance and the alteration probability. After that, we have proposed three methods (see Sect. V) to improve the robustness to make the network more robust against the possible alterations.

III. Proposed Robustness measure

Differently from NN classifiers in which the output of the network is either correct or not, in NN used for regression tasks, the correctness of the network can be quantified with a continuous measure. The main approaches to evaluate the performance of neural networks for regression problems are the *Mean Squared Error* (MSE) and the *Mean Absolute Error* (MAE). The former represents the average of the squared difference between the target value and the value estimated by the model; since it squares the residuals, it penalizes even small errors, leading to over-estimation of how bad the model is. The latter is the absolute difference between the target value and the value estimated by the model. Given this definition, MAE is more robust to outliers and does not penalize the errors as extremely as MSE [24]. However, since MAE scale is the same as the data being measured, its value is absolute and it is difficult to easily understand the relative error. For this reason, in this paper, we will use the *Mean Absolute Percentage Error* (MAPE), the percentage equivalent of MAE. Considering an input set of size n, where y_i is the real value for the input i and $\hat{y}_i$ is the estimated value for the same input, the MAPE is defined as follows:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{\|y_i - \hat{y}_i\|}{y_i}$$

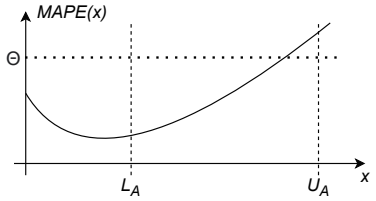


Fig. 3. *MAPE* for different alteration levels

NNs can be compared using the *MAPE* in nominal conditions ($MAPE_0$), which is computed using the test set. The MAPE metric, as warned in [14], is not applicable in problems where the real value for y_i is close or equal to zero, because it results in very large numbers. In our case study, pO_2 values are never close to zero, so *MAPE* is suitable.

A. Alterations

In a real scenario, data to be processed by an MLP can be altered w.r.t. their nominal shape. For instance, an MLP that has been trained on data describing how a physics measure changes in time, may be affected in its estimations by acquisition time variations, due to clock offsets or to a cut of the communication between the sensor and the processing unit. These are examples of *alterations*. A formal definition of alteration for all kinds of inputs is given in [3] and reported in the following.

Definition 1 (Alteration)**.** An *alteration* of type A of an input t is a transformation of t that mimics the possible effect on t when a problem during its acquisition or elaboration occurs in reality. We identify with $[L_A, U_A]$ the range of plausible alterations of type A.

Higher levels of alterations may lead to greater MAPE values in the estimation made by the MLP (see Fig. 3).

In some application domains, alterations can randomly occur at any level, while, for other types of applications, some alteration levels are more likely to occur than others. To reflect this characteristic of alterations, we define the concept of *alteration probability*.

Definition 2 (Alteration Probability)**.** Given an alteration of type A, we identify with p_A the *probability* of the alteration A, i.e., the probability distribution of the alteration level having as support the interval $[L_A, U_A]$.

The most common examples of probability distributions for alterations are:

- *Uniform probability*: all the alteration levels are equally probable, as shown in Fig. 4(a) and formally defined as:

$$p_A(x) = \begin{cases} \dfrac{1}{U_A - L_A} & L_A \le x \le U_A \\ 0 & \text{otherwise} \end{cases}$$

- *Linear probability*: lower alteration levels are more probable than the higher ones, as shown in Fig. 4(b) and formally defined as:

$$p_A(x) = \begin{cases} \dfrac{2}{(U_A - L_A)^2} \cdot (U_A - x) & L_A \le x \le U_A \\ 0 & \text{otherwise} \end{cases}$$

However, other types of probability functions can be used as well, such as the truncated normal or half-normal distributions.

Note that the alteration probability can also be used to specify the "importance" the user wants to give to a level of alteration. For instance, a uniform probability is more likely to be used for systems that should be equally resilient to all the levels of an alteration within a given interval. On the other hand, a linear probability can be preferred when the system is not critical, and it is more important to perform better on lower alterations than on the higher ones.

B. Tolerance

Different *MAPE* values can be more or less acceptable, depending on the criticality of the task performed by the NN and by the precision required by the system requirements. Therefore, we define the *tolerance* to the NN error as follows:

Definition 3 (Tolerance)**.** Let Θ be a threshold representing the maximum *MAPE* value accepted by the system requirements, and $MAPE_A(x)$ the value of the error when an alteration A of level x is applied to the input data. We identify the desired *tolerance* to the error $MAPE_A(x)$ with a function $Tol_{MAPE_A}(x)$ such that:

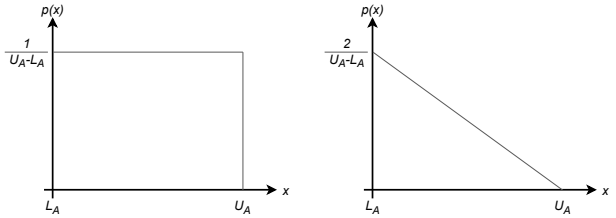
$$\begin{array}{ll} Tol_{MAPE_A}(x) = 1 & \text{for } MAPE_A(x) = 0 \\ 0 \le Tol_{MAPE_A}(x) \le 1 & \text{for } 0 < MAPE_A(x) \le \Theta \\ Tol_{MAPE_A}(x) = 0 & \text{for } MAPE_A(x) > \Theta \end{array}$$

Users can choose the desired type of tolerance function. In the following, two examples of tolerance functions are described:

- *Uniform tolerance*: all the different values of $MAPE_A(x)$ are considered in an equal way, as shown in Fig. 5(a) and formally defined as:

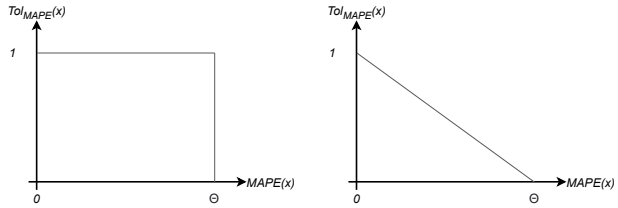
$$Tol_{MAPE_A}(x) = H(\Theta - MAPE_A(x)) \quad \text{where } H(k) = \begin{cases} 1, & k \ge 0 \\ 0, & k < 0 \end{cases}$$

The intuition is that, as long as the *MAPE* is below or equal to the threshold Θ, the tolerance is maximum, otherwise is 0.



(a) Uniform probability (b) Linear probability

Fig. 4. Examples of functions describing the probability p_A of an alteration level A



(a) Uniform tolerance (b) Linear tolerance

Fig. 5. Examples of functions describing the tolerance

- *Linear tolerance*: lower values of $MAPE_A(x)$ are more tolerated than the higher ones, as shown in Fig. 5(b) and formally defined as:

$$Tol_{MAPE_A}(x) = \frac{\max(\Theta - MAPE_A(x), 0)}{\Theta}$$

C. Robustness for an MLP

Applying alterations as defined in Def. 1 will change the accuracy of the MLP model; generally, it decreases by increasing the alteration level. A decreasing accuracy means that the *MAPE* for the estimation will increase, as previously shown in Fig. 3.

Given the tolerance function *Tol*, the *alterations* A, and their *probability* p_A, we can define the *robustness* as follows.

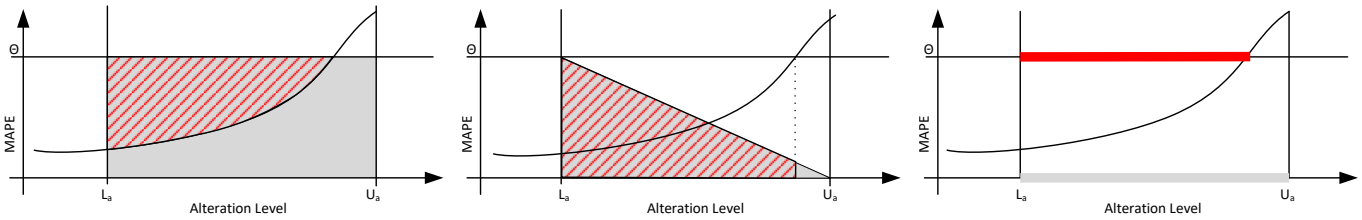
Definition 4 (Robustness)**.** Let M be an MLP model under evaluation. Let $MAPE_A(x)$ be the value of the error done by M when an alteration A of level x is applied to the input data, $p_A(x)$ the probability of the alteration, and $Tol_{MAPE_A}(x)$ the tolerance for *MAPE* values of the selected network. The *robustness* $rob_A(M) \in [0, 1]$ of MLP M w.r.t. alterations of type A in the range $[L_A, U_A]$ is formally defined as:

$$rob_A(M) = \int_{L_A}^{U_A} Tol_{MAPE_A}(x) \cdot p(x)\, dx \qquad (2)$$

Intuitively, robustness is the sum (integral) of all the errors the network commits when all the possible alterations are applied. Alterations are weighted by their probability and errors by the specified tolerance.

The robustness definition guarantees the following properties:

1) the robustness is always between 0 and 1, i.e., $0 \le rob_A(M) \le 1$;
2) if a network has always zero error, its robustness is 1, i.e., $rob_A(M) = 1$ if $\forall x \in [L_A, U_A], MAPE_A(x) = 0$.



(a) UL robustness (b) LU robustness (c) UU robustness

Fig. 6. Graphical representation of different types of robustness

Note that the condition is sufficient but not necessary, i.e., the robustness can also be 1 for systems in which the error is greater than 0 for some alteration intervals (e.g., if uniform tolerance is used and the error is never greater than Θ);

3) if a network has an error always greater than the specified threshold Θ, its robustness is 0: $rob_A(M) = 0$ if $\forall x \in [L_A, U_A], MAPE_A(x) > \Theta$.

The robustness depends on the type of tolerance and probability chosen. For example, we can define the following subtypes of robustness:

a) UL robustness: When uniform probability (any alteration level is equally likely) and linear tolerance (lower errors are preferable) are chosen, we obtain the following definition of robustness:

$$rob_A^{UL}(M) = \frac{\int_{L_A}^{U_A} \max\left(\Theta - MAPE_A(x), 0\right)\, dx}{\Theta \cdot (U_A - L_A)}$$

This definition of robustness computes the ratio between the striped red area and the gray one in Fig. 6(a). In this way, it can be used for systems where, for higher alteration values, large *MAPE* values are acceptable, while, for lower alteration values, the smaller the *MAPE* value the better.

b) LU robustness: It is obtained when linear probability and uniform tolerance are used:

$$rob_A^{LU}(M) = \frac{\int_{L_A}^{U_A} H(\Theta - MAPE_A(x)) \cdot (U_A - x)\, dx}{\frac{1}{2} \cdot (U_A - L_A)^2} = \\ = \frac{\int_{x \in [L_A, U_A] | MAPE(x) < \Theta} \left(\Theta \cdot \frac{U_A - x}{U_A - L_A}\right) dx}{\frac{1}{2} \cdot \Theta \cdot (U_A - L_A)}$$

This definition of robustness computes the ratio between the area of the striped red region and the area of the gray triangle in Fig. 6(b). The definition is suitable for systems where it is crucial to respect the threshold Θ along all the alteration interval $[L_A, U_A]$, in particular for low (and more likely) levels of alteration.

c) UU robustness: It is obtained when uniform probability and uniform tolerance are used:

$$rob_A^{UU}(M) = \frac{\int_{L_A}^{U_A} H(\Theta - MAPE_A(x))\, dx}{\Theta \cdot (U_A - L_A)}$$

With this definition, we compute the ratio between the lengths of the red and the gray lines in Fig. 6(c). UU robustness is suitable for systems where it is crucial to respect the threshold Θ along all the alteration interval $[L_A, U_A]$, independently from the probability of the alteration.

d) LL robustness: It is obtained when linear probability and linear tolerance are used:

$$rob_A^{LL}(M) = \frac{\int_{L_A}^{U_A} \max(\Theta - MAPE_A(x), 0) \cdot (U_A - x)\, dx}{\frac{1}{2} \cdot \Theta \cdot (U_A - L_A)^2}$$

Higher alteration levels, which impact more on the input data, may be the ones that lead to higher values of *MAPE*. However, in some systems, these alteration levels can be less probable than the lower ones, and a user may be less worried about some high error in very rare cases. Thus, LL robustness is suitable for these kinds of systems, when the user does not want to penalize too much high error values for the highest alteration levels. Note that it is difficult to provide a graphical interpretation of LL robustness as done for the other types of robustness.

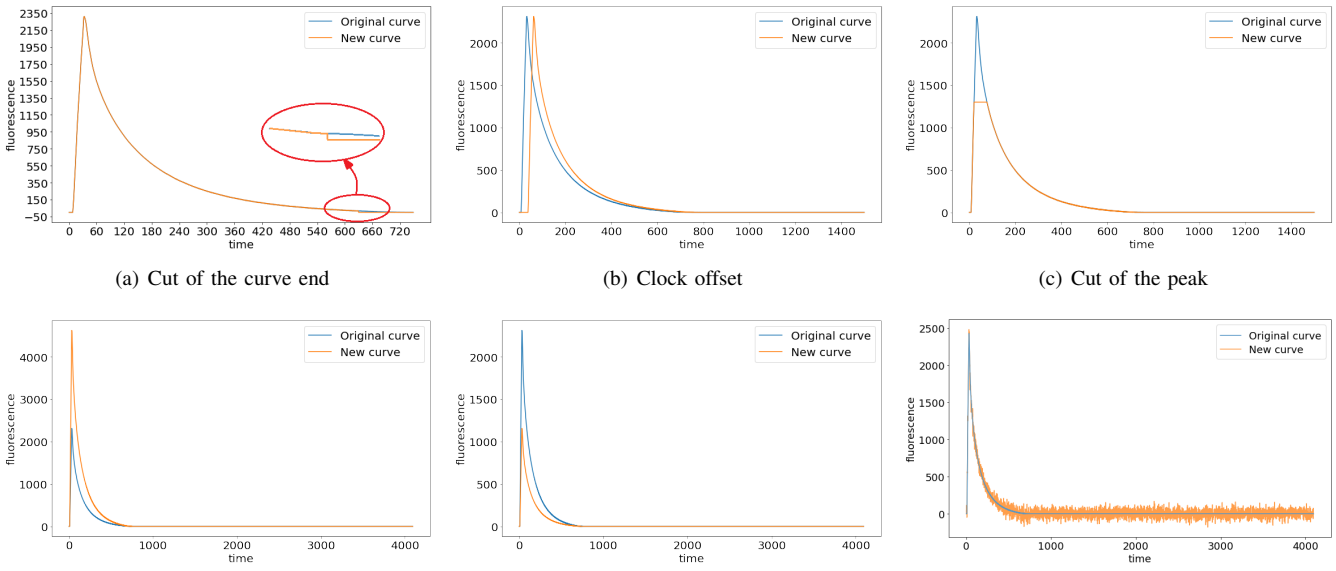
IV. Robustness Analysis

The case study considered in this paper (see Sect. II) is classified as a safety-critical system since the neural network estimates the blood oxygen pressure, a critical blood parameter. The industrial partner, after a careful study, has set the maximum accepted MAPE Θ to 10%, a value that physician experts considered safe. In the following, we first describe in Sect. IV-A the most plausible alterations in signal acquisition that may affect the operation of the sensor; then, we describe in Sect. IV-B how we actually measure the robustness, and finally, in Sect. IV-C, we present the robustness results of the case study.

A. Alterations

Table I shows the parameter values used for each alteration, which have been chosen using the knowledge achieved through the domain analysis. The value $\times n$ represents the number of alteration levels applied and uniformly sampled for each alteration type in the range $[L_A, U_A]$. For each alteration, one extreme of the interval consistently leaves the curve unaltered (e.g., clock offset with 0 does not alter the curve).

Cut of the curve end: it consists in "cutting" the end of the curve obtained in response to the spotlight pulse (see



(a) Cut of the curve end (b) Clock offset (c) Cut of the peak

(d) Amplification (e) Attenuation (f) Gaussian noise

Fig. 7. Typical alteration examples for the pO_2 estimation case study

TABLE I
ALTERATIONS VALUES USED FOR ROBUSTNESS ANALYSIS

ID	Alteration	L_A	U_A	$\times$ n
CC	Cut of the curve end	620 *ms*	640 *ms*	21
CO	Clock offset	0 *ms*	30 *ms*	31
CP	Cut of the peak	1300 *RFU*	4000 *RFU*	271
AM	Amplification (scale)	100 %	200 %	11
AT	Attenuation (scale)	50 %	100 %	6
GN	Gaussian noise	0	50	51

Fig. 7(a)). This alteration mimics a real situation in which a disruption, failure, or anomalous system behavior leads to a problem in which the final part of the curve is lost during acquisition. In our analyses, we have considered only cuts in the last range, i.e., in $[620, 640]$. The alteration is implemented by choosing a t in that range and setting to zero the response after t.

Clock offset: it represents the difference of time calculation in different systems (see Fig. 7(b)). It can happen when the microprocessor of the acquiring system is not correctly set up, or there is a delay in signal generation. We have used a maximum offset of 30 *ms*, a value that guarantees a delay in the clock greater than the maximum width of the intervals considered as input.

Cut of the peak: it aims at representing a cut in the peak of the curve, which simulates saturation events (see Fig. 7(c)). This is a common phenomenon in electronics, where a signal can not exceed a specific range of values, due to problems in the acquisition chain or voltage drops. In these cases, high values of the curve are set to a threshold. The signals analyzed in our case study have a maximum amplitude of 4000 *RFU* (Relative Fluorescence Units). So, in order to cover only the relevant values, we have applied a cut starting from 1300 *RFU* to 4000 *RFU*.

Amplification: it simulates the effect of different probes and spotlights on the measurement of the same blood sample (see Fig. 7(d)). In fact, from domain analyses, it has been demonstrated that changing the probes or spotlights slightly amplifies the response curve, even if the real pO_2 value remains the same. We have tested amplifications up to 200% of the original amplitude.

Attenuation: it represents the opposite of the amplification, i.e., the signal is attenuated by using different probes and/or spotlights (see Fig. 7(e)). The two alteration types have been evaluated separately since we wanted to highlight potential differences between the two. We have used attenuation values up to 50% of the original amplitude.

Gaussian noise: it simulates the noises that are common for electronic signals (see Fig. 7(f)). In our case study, we generate Gaussian noise with a standard deviation into the range between 0 (i.e., the absence of noise) and 50 (i.e., the maximum value leading to an acceptable and plausible signal-to-noise ratio).

B. Measuring Robustness

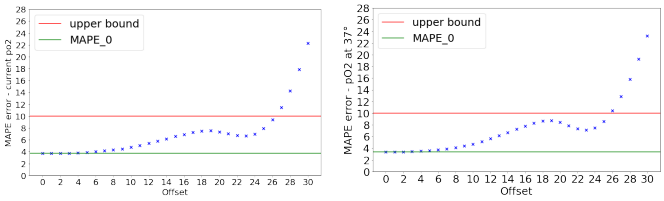
In this section, we present the robustness analysis algorithm. The pseudocode of the algorithm used for this purpose is shown in Alg. 1[2]. In order to analyze the robustness of an MLP, we need to consider *curves_raw* and *targets*, i.e., the curves of fluorescence obtained in response to the bright pulse and the true values for the pO_2 (both at the current temperature and at 37 °C). The analysis is performed over a model M, for an *alteration* with values uniformly distributed in an interval *a_levels*, each one with a probability described by the

[2]The source code that computes the robustness is published at https://bit.ly/3ilwNOX, while we can not distribute models and data for IP protection.

Algorithm 1 Algorithm for robustness analysis

Require: *curves_raw*, the set of fluorescence curves obtained in response to the bright pulse
Require: *targets*, the true values of pO_2
Require: M, the model trained to estimate pO_2 values
Require: *alteration*, the applied alteration
Require: *a_levels*, the list of n levels uniformly distributed in the range of the chosen alteration, i.e., $[L_A, U_A]$
Require: *prob*, the probability distribution of the alteration to be applied
Require: *Tol*, the desired tolerance function
Require: *intervals*, the list of intervals on which the mean values have to be computed (i.e., [50, 60], [90, 110], [190, 210], [340, 360], [620, 640])
Ensure: *rob_res*, the computed robustness value

▷ For each level of alteration
1: **for all** $l \in a\_levels$ **do**
  ▷ Apply the alteration level to all the input curves
2: *alt_curves* ← *alteration*.`apply`(*curves_raw*, *l*)
3: **for all** $c \in alt\_curves$ **do**
    ▷ Compute the mean values in the intervals
4: *meanValues* ← `compMeanValues`(*c*, *intervals*)
    ▷ Compute estimations for altered data
5: *pred*.*add*(*M*.`estimate`(*meanValues*))
6: **end for**
  ▷ Compute errors
7: *MAPE*[*l*] = `compute_mape`(*pred*, *targets*)
8: **end for**
9: *rob_res* ← ROBUSTNESS(*MAPE*, *prob*, *Tol*, *a_levels*)
10: **return** *rob_res*



(a) pO_2 at the current temperature (b) pO_2 at 37 °C

Fig. 8. *MAPE* variation during robustness analysis for the *Clock offset* alteration

probability distribution *prob*. Moreover, the tolerance function *Tol* has to be specified. The algorithm describes the procedure used to compute robustness. For each alteration level l (line 1), it performs the following instructions. First, starting from *curves_raw*, new altered fluorescence curves are generated (line 2), by applying the defined level l of the *alteration*. Then, for each generated curve, the algorithm extracts the mean values in the five intervals of interest (line 4), which are used as input for computing the pO_2 estimation (line 5). The results are then used to compute the *MAPE* (line 7) for the defined level l of the *alteration*. Finally, having all the partial *MAPE* values, the function ROBUSTNESS (line 9) performs the robustness computation by solving the integral as per Def. 4.

C. Robustness Results

Using the alterations presented in Sect. IV-A, and applying the procedure explained in Alg. 1, we have computed the robustness of the MLP under analysis. As an example, we report in Fig. 8 the data of the MAPE obtained during robustness analysis applying *Clock offset* alteration from 0 *ms* to a maximum offset of 30 *ms*. We can see that the MAPE remains under the fixed threshold (Θ = 10%) until an offset of 26 *ms*, both for the pO_2 at the current temperature and the pO_2 at 37 °C.

TABLE II
ROBUSTNESS W.R.T. ALTERATIONS FOR THE ORIGINAL NETWORK

Alteration	UL Rob [%] pO_2	UL Rob [%] pO_2 37 °C	LU Rob [%] pO_2	LU Rob [%] pO_2 37 °C	UU Rob [%] pO_2	UU Rob [%] pO_2 37 °C	LL Rob [%] pO_2	LL Rob [%] pO_2 37 °C
CC	5.16	6.70	26.53	26.53	14.29	14.29	9.85	12.65
CO	37.19	34.52	98.34	97.40	87.10	83.87	48.65	47.59
CP	48.84	50.96	97.10	96.97	82.96	82.59	59.61	62.52
AM	28.73	27.16	96.69	92.56	81.81	72.73	41.02	41.09
AT	33.52	32.31	88.89	88.89	66.67	66.67	47.71	47.54
GN	32.89	32.37	98.12	96.16	86.27	80.39	45.67	46.10
Avg Rob	31.06	30.67	84.28	83.09	69.85	66.76	42.09	42.92

Table II reports the obtained robustness results (for UL, LU, UU, and LL presented in Sect. III) for all the alterations introduced in Sect. IV-A, both for the pO_2 at the current temperature and pO_2 at 37 °C. From an analysis of the results, we have obtained the following observations:

Robustness comparison among different formulas: The selected probability distribution and tolerance function affect the robustness value. Indeed, if we consider the same alteration type, we observe that the robustness values of the different robustness formulas vary greatly. For example, choosing a linear tolerance function (i.e., UL and LL) the robustness is much lower than that obtained with the uniform tolerance (i.e., LU and UU).

> The different ways of considering the alteration levels and the MAPE error do indeed lead to different robustness levels, showing that some formulas are more demanding than others.

Effect of different robustness formulas on network analysis: In the previous observation, we have noticed that there is indeed a difference between the robustness values computed by the different formulas. We now observe that such difference does not simply lead to a modification of the magnitude of the robustness value across different formulas, but also affects the comparison we can do between different alteration types.[3] For example, considering alteration *cut of the peak* and the robustness of pO_2, we notice that it has the highest UL and LL robustness values, while the robustness values for LU and UU are both the third ones in their rankings. On the other hand, considering the robustness of pO_2 for alteration *Gaussian noise*, we observe that UL and LL robustness values are both ranked fourth in their rankings, while LU and UU robustness values are both ranked second. Therefore, while UL and LL consider the network to be more robust against *cut of the peak* rather than against *Gaussian noise*, LU and UU consider the opposite. This clearly shows that different formulas are really considering different aspects related to the network error (i.e., the amount of the error and the probability of the alteration

[3]This would also apply to the comparison of the robustness values of different networks.

causing it), and this leads to different judgments. This is a positive aspect of our parameterized robustness definition.

> The different robustness formulas are indeed different, as they lead to different conclusions when analyzing the robustness values of different alterations.

Agreement on best and worse cases: While in the previous points we have observed that the different robustness formulas can consider differently some alterations (when the amount of error and the probability affect the ranking), when the error is always very high or very low, the different formulas are consistent. For example, the *cut of the curve end* is considered by all formulas as the alteration with the minimum robustness value: indeed, values near the end of the curve represent important information for the network estimation, and cutting them greatly affects the MAPE, and so robustness values of the different formulas are very low.

> The different robustness formulas tend to agree when the MAPE error is consistently very high or very low.

Because of the criticality of the case study system, as suggested by our industrial partner, we have selected the *UU robustness*, i.e., the uniform probability distribution for the alteration and the uniform tolerance function. The choice of uniform probability is motivated by the fact that, in medical practice, the sensor should be robust against any alteration level, regardless of the probability of the alteration (since even a very rare alteration level can lead to terrible consequences). On the other hand, the choice of uniform tolerance is motivated by the industrial partner, in accordance with the physicians who consider any error below the selected threshold Θ safe for medical practice.

V. Improving Robustness

Our industrial partner, after having discussed with us about the robustness analysis we have performed and presented in Sect. IV-C, asked us to find a way (if existing) to improve the robustness of their MLP model, with only minimal changes in the system architecture. For this goal, we have examined three different methods: (A) data augmentation with recombined data, (B) data augmentation with altered data, and (C) incremental learning. The first two methods consist in enriching the training dataset by adding artificial or altered data, in order to retrain the original MLP (without changing its structure); the third method, instead, consists in adding another MLP aside the original one and leaving the original one as it is.

A. Data Augmentation with Recombined Data (DA-RD)

Data Augmentation (DA) is a wide subject [23], including a suite of techniques that increase the size and quality of the training dataset.

The first suitable approach to improve the robustness is the one using a modified version of some classical data augmentation methods for classification tasks, i.e., the *data augmentation with recombined data* (DA-RD). In particular, the idea of this technique is to create new input data (*virtual*) by recombining existing ones (*real*) [11]. This solution can be easily applied to our case study because, in our data set, there exist many curves that, although they represent the same labeled pO_2 true value, differ in shape, mainly because of differences in temperature, and types of probes and/or spotlights. For this reason, we have determined new samples in two different ways: by averaging two curves with out-of-range estimation errors (i.e., higher than 10%), and by averaging a curve with a high estimation error with one with a low error. In both cases, the curves which have been averaged are with the same pO_2 target values. The intent is to capture new intermediate curves with a known true value of pO_2. In our case study, we had 1141 samples with high estimation error, so we have generated 546 new samples with the first method and 1141 with the second one, obtaining a new dataset composed of 1687 additional samples with which we have retrained the MLP under analysis.

TABLE III
$MAPE_0$ OF THE ORIGINAL NETWORK AND THE RETRAINED ONES

Model	# Training curves	$MAPE_0$ [%] pO_2	pO_2 37 °C	Avg UU robustness pO_2	pO_2 37 °C
Original	12,990	3.70	3.35	69.85	66.76
DA-RD	14,677	3.08	3.08	64.92	65.36
DA-AD	19,485	3.11	3.06	78.05	76.13
IL	6,495	3.32	3.20	78.89	76.25

This new retrain has led to enhanced performance, in terms of $MAPE_0$, compared with the one of the network trained with original data, as shown in Table III. In particular, the results show that using the DA-RD we have reduced the values of $MAPE_0$, both for the pO_2 and pO_2 at 37 °C.

In terms of robustness, this technique has, unfortunately, worsened the robustness on average (see Table III), but with very different results depending on the specific alteration applied. Table IV shows the results in terms of robustness for the estimation provided by the new networks for each alteration. The columns ΔRob show the difference w.r.t. the robustness of the original network.

The results show that by using the DA-RD technique, we only have been able to slightly increase (or maintain equal) the performance for the majority of alterations, while for the amplification and Gaussian noise alteration, we have significantly degraded the robustness of the model. This has proven not to be a very good solution, but it was not a surprise, since by using this technique we do not add any new data that could mimic possible unexpected alterations.

B. Data Augmentation with Altered Data (DA-AD)

Our robustness definition considers the error done by the network on altered data. *DA with altered data* (DA-AD) is a retraining approach that directly aims at increasing robustness: it explicitly targets the alterations and consists in adding, during the training phase, also data altered with the alterations listed in Sect. III-A. To reduce the training time and to keep

TABLE IV
ROBUSTNESS W.R.T. ALTERATIONS FOR THE NETWORKS RETRAINED WITH THE THREE APPROACHES

	Original		DA-RD				DA-AD				Incremental learning			
	Rob [%]		Rob [%]		Δ Rob [%]		Rob [%]		Δ Rob [%]		Rob [%]		Δ Rob [%]	
Alteration	pO_2	pO_2 37 °C	pO_2	pO_2 37 °C	pO_2	pO_2 37 °C	pO_2	pO_2 37 °C	pO_2	pO_2 37 °C	pO_2	pO_2 37 °C	pO_2	pO_2 37 °C
CC	14.29	14.29	19.05	19.05	4.76	4.76	28.57	28.57	14.28	14.28	14.29	19.05	0.00	4.76
CO	87.10	83.87	87.10	83.87	0.00	0.00	87.10	87.10	0.00	3.23	87.10	87.10	0.00	3.23
CP	82.96	82.59	83.70	83.70	0.74	1.11	83.33	84.07	0.37	1.48	83.70	83.70	0.74	1.11
AM	81.81	72.73	54.54	54.54	-27.27	-18.19	90.90	72.73	9.09	0.00	100.00	100.00	18.19	27.27
AT	66.67	66.67	66.67	66.67	0.00	0.00	100.00	100.00	33.33	33.33	100.00	83.33	33.33	16.66
GN	86.27	80.39	78.43	84.31	-7.84	3.92	78.43	84.31	-7.84	3.92	88.24	84.31	1.97	3.92

the network focused on the right values, we have added to the training set only the alterations causing a variation of the MAPE up to 5%, obtaining a dataset 1.5 times greater than the original one.

This new retrain has led to enhanced performance, in terms of $MAPE_0$, compared to the one of the MLP trained with original data as shown in Table III: the results show that using data augmentation has reduced the values of $MAPE_0$, both for the pO_2 at the current temperature and at 37 °C, w.r.t. the original model. The comparison between DA-RD and DA-AD highlights only minimal differences between the two.

Table III shows that the average robustness has significantly increased on average (more than 13%). Table IV shows the robustness results for the different alteration types. Unlike DA-RD, with DA-AD, we have been able to consistently increase the robustness of the analyzed MLP, even more than 33% for a single alteration. However, the retrain requires more time, since it is performed using nearly $7,000$ additional input curves. The only alteration for which the robustness has decreased is the Gaussian noise during the estimation of the pO_2 at the current temperature. However, the noise is, by definition, randomly distributed and, so, increasing the robustness w.r.t. this kind of alteration is very difficult.

C. Incremental Learning (IL)

Data augmentation contributes to enhancing the robustness performance of a network but requires the retraining of the original network with a larger set of inputs. Thus, in order to reduce the learning time and increase the network generalization at the same time, we have implemented a different approach, known as *incremental learning* (IL). It allows improving the model performances without retraining the whole network. In the literature, incremental learning is performed whenever new samples are available, by adjusting what has been learned according to them. This method was designed to work as an online technique, but we have adapted it (as suggested by [19]) to be used as an offline approach.

Our approach adds knowledge to an existing MLP without modifying it in order to improve the performance of the network. For this reason, the system structure has been modified. The new estimator is composed of two networks, as shown in Fig. 9, one representing the original network and one the new and non-previously trained network. During the retraining phase, only the new network is trained using the new altered dataset; the outputs of the two networks are used in the loss function computation and to update the weights in the new network. In this way, we have obtained a new network that is able to classify new inputs, using the support information provided by the original one. After the retraining phase, the inference of an input vector uses both networks: the final estimation is obtained by averaging the estimations of both models (the original and the new one).

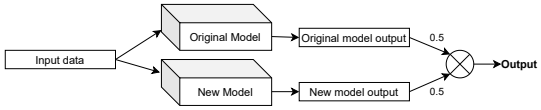


Fig. 9. Incremental learning technique

This approach has the advantage of avoiding the retrain of the whole system, but only a part of it, and of using a limited dataset composed of only altered input data. This led to a lower training time than the one required for the DA-AD technique, despite having a total dataset of the same size. Moreover, the original network is not modified; this is an advantage, as sometimes modifications are not possible if the model is read-only or available only as a black box.

Table III shows that the average robustness has significantly increased on average (more than 13%) w.r.t. to the original network. This value is slightly higher than the one obtained with DA-AD. Regarding $MAPE_0$, instead, it is slightly higher than the one obtained with DA-AD, but still better than the one of the original network. Table IV shows the results of robustness for each alteration type. We have been able to increase the robustness of the analyzed MLP up to 33%, and no robustness is decreased with any alteration (differently from DA-AD, for which we had some decreases).

These results show that IL is able to combine the advantages of the original network, i.e., focusing only on relevant input features, and the ones of the data augmentation, i.e., guaranteeing higher robustness w.r.t. the standard-trained network. At the end, the industrial partner has decided to choose the IL technique. Indeed, even if it is not the best choice in nominal conditions (i.e., $MAPE_0$), it still contributes to decreasing the $MAPE_0$ and is the best-performing method in terms of robustness. Moreover, its application requires less training time and does not change an important part of the network structure.

VI. Discussion

We here report some considerations of the industrial partner, we discuss how the proposed robustness measure can be integrated in an industrial development process, and how it can be used to provide confidence on the correctness of a DNN-based component used in safety-critical domains.

In our proposed solution, the user needs to select the probability of the alterations of interest. Nevertheless, in some application scenarios, the probabilities may be unknown and this can make it difficult to apply the formula for robustness computation. In these cases, interpreting the probability as a function that describes the "importance" of each alteration level is a viable solution. Indeed in our approach, the probability can act as alteration weight in the robustness computation.

From the experiments with the three robustness improvement techniques (see Sect. V), we observe that, for most alterations, the robustness is not 100% also after retraining. This shows that obtaining optimal robustness by simply retraining may not be possible. In this case, by looking at the final robustness results, the industrial partner has understood which are the most critical alterations that can still affect the network, and planned to adopt countermeasures from an electronic point of view. For example, the impact of Gaussian noise may be reduced by improving the cables shielding, or by amplifying the acquired signal so that the noise is less relevant.

Our industrial partner plans to obtain the certification of the new sensor. Note that, nowadays, many medical devices based on AI algorithms are certified by competent authorities [6], but a lack of clarity on the approval of AI/ML-based medical devices and algorithms characterizes the certification process. FDA approves AI-based medical systems in three cases: (i) AI algorithms have shown to be at least as safe and effective as another similar legally marketed; (ii) critical algorithms with high impact on humans are premarket approved, then the FDA determines if the device's safety and effectiveness is supported by satisfactory scientific evidence; (iii) novel medical devices which offer adequate safety and effectiveness are approved after performing a risk-based assessment. In our case, the sensor can be certified by showing that on the extensive data set, it proves to be at least as reliable as other devices already in the market. Although no robustness assessment is currently formally required, the industrial partner may include our results as part of a risk-assessment documentation because they allow to evaluate how the model resists to input perturbations.

After our case study, the industrial partner is planning to introduce an automatic robustness assessment phase in their current pipeline when developing ML-based solutions. To this end, we are working on extending ROBY [4] in order to support also NN-based estimators.

We have applied the proposed approach based on MLP to the estimation of pO_2, nevertheless, our method is applicable to other models also in different application domains.

VII. Related Work

Nowadays, the testing of NN-based systems [18] is performed by analyzing different properties of the networks, and robustness is one of them [26]. The robustness of NNs has been studied in a lot of different ways since they are increasingly used in safety-critical domains. Most of the researches are focused on robustness for NN classifiers. For example, in [2], we have investigated on the robustness of neural networks images classifiers used for medical analysis, while in [4] we have extended the analysis to classifiers for all types of data, by introducing ROBY, a tool for robustness analysis. Nevertheless, in recent years, some researches have been conducted also for other kinds of NNs. For example, Xiange et al. [21] study the robustness of an MLP used in nuclear power plants for the diagnosis of loss of coolant accidents in nuclear reactors. The authors propose a robustness measure based on the maximum value assumed by the mean square error of the estimation. However, most of the robustness analyses are focused on *adversarial* robustness [22], [5], i.e., the robustness of the network w.r.t. adversarial examples. Carlini and Wagner [7] demonstrate that some of the most recent techniques used to increase the robustness w.r.t. adversarial examples can be easily fooled with other adversarial generation techniques. Thus, when creating an NN, especially for safety-critical domains, it is of paramount importance to evaluate and increase the robustness for plausible alterations as well, depending on the application field.

In our experiments, among all the alterations, we have added Gaussian noise to assess the robustness of the model. This approach has been also proposed by Dey et al. [9], who add three regularizing terms to the loss function of an MLP, in order to obtain a higher robustness w.r.t. multiplicative and/or additive noise. Other methods try to improve the robustness of an MLP by changing the training algorithm. For instance, Kerlirzin and Vallet [15] propose a modified back-propagation algorithm leading to enhanced robustness versus the destruction of neurons. In our case, instead, the industrial partner asked to maintain the same training algorithm.

VIII. Conclusion

In this paper, we have proposed a new robustness estimation measurement that includes two important concepts in its definition: alteration probability and tolerance. The former indicates the probability of an alteration level, while the latter indicates the acceptance of certain amounts of error, specified as Mean Absolute Percentage Error (*MAPE*). These functions are chosen by domain experts and they significantly affect robustness results. We have applied the robustness definition to an industrial case study to estimate the value of blood oxygen pressure, and we have compared the robustness results obtained with different alteration probability distributions and tolerance functions. Moreover, we have proposed three methods to increase the robustness without significant changes to the system architecture: data augmentation with recombined data, data augmentation with altered data, and incremental learning. The industrial partner, based on the obtained results, has chosen incremental learning because it increases the robustness and decreases the nominal *MAPE* (i.e., $MAPE_0$).

REFERENCES

[1] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[2] P. Arcaini, A. Bombarda, S. Bonfanti, and A. Gargantini. Dealing with robustness of convolutional neural networks for image classification. In *2020 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 7–14, 2020.

[3] P. Arcaini, A. Bombarda, S. Bonfanti, and A. Gargantini. Efficient computation of robustness of convolutional neural networks. In *2021 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 31–38, 2021.

[4] P. Arcaini, A. Bombarda, S. Bonfanti, and A. Gargantini. ROBY: a tool for robustness analysis of neural network classifiers. In *14th IEEE Conference on Software Testing, Verification and Validation, ICST 2021, Porto de Galinhas, Brazil, April 12-16, 2021*, pages 442–447. IEEE, 2021.

[5] E. R. Balda, A. Behboodi, and R. Mathar. Perturbation analysis of learning algorithms: Generation of adversarial examples from classification to regression. *IEEE Transactions on Signal Processing*, 67(23):6078–6091, 2019.

[6] S. Benjamens, P. Dhunnoo, and B. Meskó. The state of artificial intelligence-based FDA-approved medical devices and algorithms: an online database. *npj Digital Medicine*, 3(1), sep 2020.

[7] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, Los Alamitos, CA, USA, may 2017. IEEE Computer Society.

[8] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[9] P. Dey, K. Nag, T. Pal, and N. R. Pal. Regularizing multilayer perceptron for robustness. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(8):1255–1266, aug 2018.

[10] S. Dong, P. Wang, and K. Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021.

[11] F. Dubost, G. Bortsova, H. Adams, M. A. Ikram, W. Niessen, M. Vernooij, and M. de Bruijne. Hydranet: Data augmentation for regression neural networks. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 438–446, Cham, 2019. Springer International Publishing.

[12] P. G. Frankl, R. G. Hamlet, B. Littlewood, and L. Strigini. Evaluating testing methods by delivered reliability. *IEEE Trans. Softw. Eng.*, 24(8):586–601, Aug. 1998.

[13] Q. Fu, F. Yang, J. Zhao, X. Yang, T. Xiang, G. Huai, J. Zhang, L. Wei, S. Deng, and H. Yang. Bioinformatical identification of key pathways and genes in human hepatocellular carcinoma after CSN5 depletion. *Cellular Signalling*, 49:79–86, sep 2018.

[14] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[15] P. Kerlirzin and F. Vallet. Robustness in multilayer perceptrons. *Neural Comput.*, 5(3):473–482, May 1993.

[16] R. Mangal, A. V. Nori, and A. Orso. Robustness of neural networks: A probabilistic and practical approach. In *Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results*, ICSE-NIER '19, pages 93–96, Piscataway, NJ, USA, 2019. IEEE Press.

[17] D. Marijan, A. Gotlieb, and M. K. Ahuja. Challenges of testing machine learning based systems. In *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 101–102, April 2019.

[18] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella. Testing machine learning based systems: a systematic mapping. *Empirical Software Engineering*, 25(6):5193–5254, sep 2020.

[19] K. Shmelkov, C. Schmid, and K. Alahari. Incremental learning of object detectors without catastrophic forgetting. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017.

[20] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[21] X. Tian, V. Becerra, N. Bausch, T. Santhosh, and G. Vinod. A study on the robustness of neural network models for predicting the break size in LOCA. *Progress in Nuclear Energy*, 109:12–28, nov 2018.

[22] M. Uličný, J. Lundström, and S. Byttner. Robustness of deep convolutional neural networks for image recognition. In *Intelligent Computing Systems*, pages 16–30. Springer International Publishing, Cham, 2016.

[23] D. A. van Dyk and X.-L. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, mar 2001.

[24] C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1):79–82, 2005.

[25] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, J. Zhao, B. Li, J. Yin, and S. See. DeepHunter: A coverage-guided fuzz testing framework for deep neural networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2019, pages 146–157, New York, NY, USA, 2019. ACM.

[26] J. M. Zhang, M. Harman, L. Ma, and Y. Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, pages 1–1, 2020.