APU- ANALYTICAL PROPAGATION OF ERRORS AND UNCERTAINTIES IN DEEP LEARNING IN THE DOMAIN OF PHYSICAL SCIENCES

Gerson Eduardo de Mello

ᵃUniversidade do Vale do Rio dos Sinos - Campus São Leopoldo, Av. Unisinos, 950 - Cristo Rei, - RS, São Leopoldo , 93022-750, Brasil

ARTICLE INFO

ABSTRACT

In recent years, neural networks have gained popularity for making predictions across various applications. However, these predictions come with inherent uncertainties stemming from factors such as data acquisition, parameter choices, network architecture, and the training process. These uncertainties, both epistemic (related to the model) and random (data-derived), require careful quantification and propagation for a comprehensive understanding of the predicted outcomes. To address this, the study focuses on assessing and quantifying the uncertainties associated with predictions made by Multilayer Perceptron (MLP) Neural Networks in the domain of physical sciences. An example is precision agriculture, where the focus may be on the physical properties of the soil. In civil engineering, interest may be directed towards the physical characterization of construction materials. In the aerospace industry, wind tunnel tests are used to evaluate the physical properties of aircraft, for example. The approach consists of investigating and quantifying the uncertainties arising from data acquisition and propagating them through the trained model. Using the error propagation law, widely applied in physical models, the APU employs equations derived from the local approximation of this model through a Taylor series. Next, the propagated data uncertainty is combined with the uncertainties associated with the model's random and systematic errors, which arise from training and testing. This combination results in the prediction uncertainty, defining a range around the predicted value with an associated probability. Additionally, a validation method for the APU is proposed, providing the user with a way to evaluate its performance before applying it to future predictions. APU, with its modular architecture, can provide model, data, and prediction uncertainties for any pre-trained MLP Neural Network, offering a low computational cost compared to Bayesian methods.Furthermore, the method may be the only one thus far to propagate the measurement uncertainty originating from the calibration of the measuring instruments used in obtaining the dataset. To evaluate the effectiveness of APU, a comparison was conducted with the popular MC Dropout method proposed by Yarin Gal in 2016, aiming to enhance uncertainty estimation in neural networks. The UC Irvine Machine Learning Repository dataset served as a reference for evaluating and comparing the results. The analysis revealed that the APU method shows promise in real-world conditions, especially in the field of physical sciences, where low computational cost and immediate response are essential. This is significant in edge computing scenarios, real-time monitoring, addressing concerns such as latency, bandwidth, and data privacy. The study contributes to advancing the application of deep learning in the physical sciences. The source code is available at https://github.com/Geduardomello/Dissertation.

1. Introduction

The exponential increase in the quantity and diversity of experimental, theoretical, and simulation data is reshaping the traditional approach to information analysis and processing, highlighting data-driven science as a new fundamental paradigm [1]. This so-called "4th paradigm" seamlessly integrates and complements the three previous approaches: empirical observations of natural phenomena, theory based on equations, models, and generalizations, and high-complexity simulations that introduced the third pillar of science, giving it a computational dimension. While these techniques have been widely applied in complex fields such as biology, medicine, and astronomy, they have gained prominence more recently in the disciplines of physics, chemistry, and materials science. Thus, the intersection of theory, experimentation, and computation has given rise to a new exploratory facet centered on data. In summary, science has evolved into a fourth dimension—it has become "datified."

Within this 4th paradigm, i.e., data-driven science, deep learning, highlighted by [2], plays a crucial role in advancing the physical sciences. This approach enables the prediction of physical quantities from experimental data obtained through measurement instruments, leveraging the ability of neural networks to learn intricate relationships between measured and predicted quantities. However, when making predictions in the field of physical sciences, it becomes imperative to consider the uncertainties associated with these predictions, as experimental data may be affected by measurement errors, and the model fitting process may have inherent flaws, as discussed by [3].

Along with the advancement of data-driven science comes significant challenges. Currently, it is widely recognized that neural networks face difficulties in providing accurate estimates of the accuracy of their predictions, often

✉ gerson.mello@senairs.org.br (G.E.d. Mello)
ORCID(s):

dealing with issues of overconfidence or underconfidence, compromising their practical applicability, as observed by [4].

In this scenario, the search for models capable of assessing the key components of uncertainty for neural networks has become a fundamental challenge. This uncertainty can be categorized into three types: random uncertainty arising from data, epistemic uncertainty related to the architecture and parameters of the network, and prediction uncertainty, which accumulates the effects of the first two uncertainties (HE, 2022). So far, artificial neural networks have failed to bridge this gap and adequately assess the uncertainty of their predictions, hindering their practical adoption, especially in domains where incorrect results can lead to catastrophic consequences. Thus, it is crucial to assess the quality of each prediction, which is closely related to the magnitude of the associated uncertainty and the coverage probability. Only with this information, including the predicted result, the corresponding uncertainty, and the coverage probability, is it possible to analyze the relative risks of decisions based on these predictions.

For example,[5] presents satisfactory results in the use of neural networks for soil pH prediction, demonstrating the potential of this modeling in precision agriculture. However, no issues related to the uncertainty of this prediction were mentioned. In most soils, phosphorus reaches its maximum availability in the pH range of 6 to 7, as various sources agree [6, 7, 8]. The term "available" suggests the integration of supply from the soil with uptake by the plant. Outside this range, for example, at 5.5, availability drops by about 40% as . Without information on the uncertainty in pH estimation, it is difficult to assess the risks associated with decision-making based on pH measurement. Just an uncertainty of ±0.5 in pH would be enough to question a pH 6 result that, seemingly, would fall within specifications, but due to uncertainty could be as low as 5.5 pH, resulting in a 40% error in estimating soil phosphorus availability. Such errors would have a direct impact on productivity, with considerable losses as illustred Figure 1.

The main objective of this study is to develop a reliable algorithm capable of simultaneously determining random, systematic, and prediction uncertainties in MLP neural networks trained for regression problems in the field of physical sciences. From the determination of this uncertainty, the goal is to obtain a confidence interval around the predicted value that allows assessing the degree of reliability of the predicted result. Additionally, the aim is to achieve a performance that equals or surpasses that of classical and contemporary state-of-the-art algorithms, with an emphasis on the feature of requiring reduced computational resources. To evaluate the effectiveness of this approach, we conducted a comparison between APU and MC Dropout, a Bayesian approach that uses the Monte Carlo method to estimate epistemic uncertainty. Additionally, we validated the APU method, using JCGM 101 as a reference, a methodology widely employed in physical sciences to validate analytical methods, with a focus on JCGM 100 as a starting point.

The structure of the paper is as follows: in addition to the introduction in Section 1, we have Section 2, which explores the concept, causes, and types of uncertainty in the context of DNNs, providing a comprehensive view of model, data, and prediction uncertainty, while drawing parallels between error theory in physics and bias-variance theory in machine learning. Section 3 presents an overview of the proposal, detailing each of the necessary steps for implementation condensed into a proposed algorithm. At the end of this section, the algorithm validation methodology is described. Section 4 introduces the evaluation methodology, presents comparative results of the model, and includes an explanation and discussion on the interpretability of the approach. Section 5 presents the achieved results. Section 6 highlights the main contributions and conclusions drawn.

2. Background

2.1. Concept of uncertainty in neural networks

Few studies have formally addressed the meaning of the word "uncertainty" within the context of DNNs. Basically, the concept has been largely associated with measuring the lack of confidence in the predictions made by the algorithms, but other interesting points have been raised, such as uncertainty being perceived as a state of data or information, including fuzziness, disagreement, lack of specificity, ambiguity, doubt, hesitation, unpredictability, indeterminacy, and so on. Among the definitions of uncertainty, notable examples include the rational quantification of the proximity between predictions and observations ([9]), the lack of knowledge or measure of imprecision associated with predictions made by models ([10]), the lack of confidence or reliability in predictions made by artificial neural networks ([4, 11]), the measure of uncertainty associated with predictions made by deep regression models ([12]), and the perceived state of data or information, such as imprecision, disagreement, lack of specificity, and ambiguity ([13]). Additionally, it is highlighted that "uncertainty" is a polysemous word related to concepts like doubt, hesitation, unpredictability, or indeterminacy, where the most common meaning refers to the lack of confidence in predictions generated by a model ([14]).

2.2. Causes of uncertainties existing in the context of DNNs

Regarding the causes of uncertainty, we can summarize the variation or uncertainty in the system's output ([15, 16, 17, 18]), inherent errors in measurement or data acquisition systems ([15, 17, 10, 14]), lack of correspondence between training and test data sets ([11, 17]), errors in the architecture or specification of the model ([15, 17]), presence of noise or measurement error in the data ([15, 10, 19, 18, 14]), lack of knowledge about the system or model parameters ([17, 10, 19, 20, 14]), incompleteness or lack of knowledge about the system ([20, 14]), inadequate representation or lack of information about the data distribution ([11, 17, 19, 14]), and unpredictability of the model or outcome ([11, 20]).
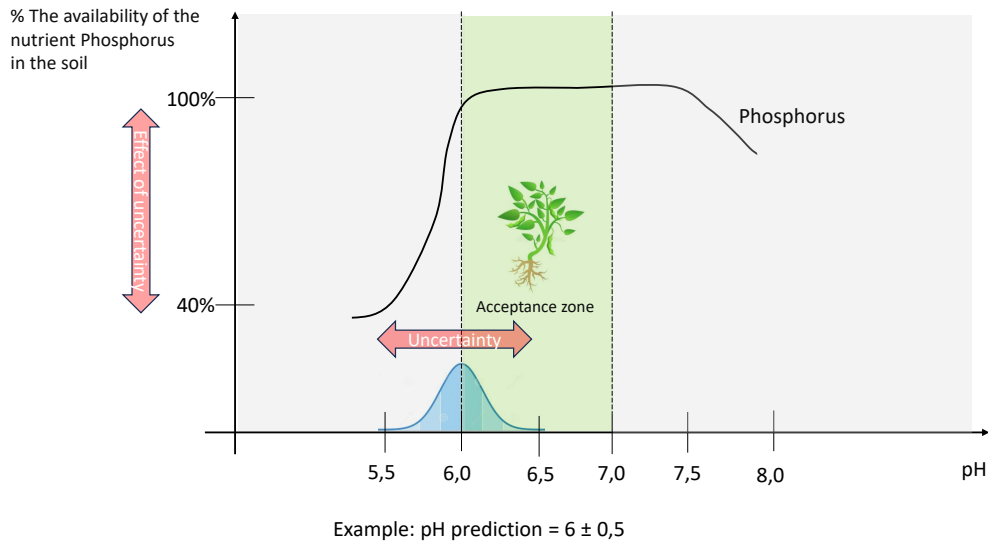
Figure 1: Application of predictions uncertainty in the Precision Agriculture

2.3. Types of uncertainties existing in the context of DNNs

In recent years, the literature has categorized uncertainties arising from neural networks into different classes. These approaches highlight the distinction between epistemic uncertainty and aleatoric uncertainty [11], as well as uncertainty related to the data, the network model used, and changes in data distribution [15].

Epistemic uncertainty arises when the test and training data are incompatible, resulting in a lack of training data in certain parts of the domain [21]. This uncertainty is associated with a lack of information about the model and can be caused by errors in the model structure, training procedures, and calibration. The structure of the neural network, including the number of parameters, directly affects its performance and the uncertainty of its predictions. Deeper networks, for example, may be overly confident in their softmax outputs [15]. Furthermore, regarding epistemic uncertainty, it can be said that it arises due to a lack of adequate knowledge. In this case, models can be defined to address different prediction-related issues based on the available knowledge. In problems involving extensive datasets, it can happen that there is a large amount of data but with little relevant information. In such situations, methods based on Artificial Intelligence can be used to identify efficient models that characterize the emerging features of the data [16]. It is important to note that the available data can often be incomplete, contain noise, discrepancies, or exhibit multiple modes of distribution.

Conversely, aleatoric uncertainty pertains to the inherent noise within the data. This noise can arise due to limitations in measurement systems, such as resolution and calibration errors. Additionally, it can result from the variability of random noise across the domain, a phenomenon referred to as heteroscedasticity [4]. Aleatoric uncertainty is the irreducible uncertainty present in the data, which is reflected in the predictions. This uncertainty is not a property of the model itself but rather an inherent property of the data distribution and, therefore, cannot be eliminated [16].

Additionally, distributional uncertainty is highlighted as an additional source of uncertainty. Real-world environments are highly variable and subject to constant changes, such as temperature, lighting, and object size. When significant changes occur in the data distribution compared to the training set, it affects the performance of the neural network [15].

However, it is important to note that some approaches may differ in terminology and emphasis placed on certain aspects of uncertainty. For example,[20] defines in his work that aleatoric uncertainty and epistemic uncertainty stem from the data. In other works, it is observed that some authors separate data uncertainty from model uncertainty, confusing them with the terms epistemic and aleatoric. As a contribution to this work, we can highlight some points that were the result of the synthesis made by analyzing the various works in this area:

1. When we want to explain a real-world phenomenon through data, the first aspect is to consider whether the collected variables are the most significant ones to explain the phenomenon. In other words, we understand that the problem is not necessarily the data, but the lack of it to explain the phenomenon. This lack of information to explain the phenomenon will impact the prediction error, thus being related to epistemic uncertainty.

2. Another issue is to know if the collected data is reliable, if the way it was collected resulted in measurement errors, bias, lack of representativeness regarding the phenomenon, etc. This uncertainty is related to the data.

3. It is difficult to establish an equivalence between randomness and data, epistemic and model. It does not seem that this is true because there is a mixture between the subject (model and data) and nature (random and epistemic).

4. In the field of physics, for example, the terms used to address uncertainty are random error and systematic error. In this context, uncertainty related to the model (epistemic) will be estimated by determining two components: systematic error and random error. For example, these errors, in the context of regression, are referred to as residuals. In a typical regression, it is expected that the mean of all residuals will approach zero, and their distribution will be approximately normal. This mean represents the systematic error of the model, also known as systematic bias, which refers to a type of error that is consistent or systematic in all measurements or observations. This error reflects the model's tendency, which would approximate the concept of epistemic. On the other hand, random error, also known as random variability, is associated with the precision of the model, as depicted in Figure 2. This error refers to a type of error that occurs unpredictably and irregularly, resulting in random variations in the results of different measurements or observations.

2.4. Error Theory in Physical Sciences

In the context of Physical Sciences, the term uncertainty encompasses both the idea of an estimate of a range of values within which the probable value is expected, as well as a measure of the possible error in the estimated value obtained by the result[22]. We can then associate uncertainty as an estimator of the effectiveness or quality of a result in the same way that the mean squared error measures the effectiveness of a prediction model. Particularly in this work, it will be useful to relate these terms, that is, Error Theory in Physical Sciences with the Bias-Variance Trade-off Theory in Machine Learning.

Error propagation is a fundamental concept in physics (and other sciences) that describes how uncertainties in measured or calculated values propagate through mathematical calculations to influence the uncertainty of the final results.

There are several methodologies for error propagation, depending on the complexity of the problem and the precision required. Two common methods are Analytical Method and Monte Carlo Method.

2.4.1. Analytical Method

The Analytical Method involves the use of differential calculus to determine how uncertainties in the initial values affect the final value. When a quantity G_2 functionally depends on another quantity G_1 through a generic rule $G_2 = f(G_1)$, the relative error of G_2 can be obtained through a Taylor series expansion. To simplify the notation, let's call G_2 as y and G_1 as x, so that $y = f(x)$. We then have, expanding $f(x)$ in a Taylor series around the point x_0:

$$f(x) = f(x_0) + \frac{df}{dx}(x - x_0) + \frac{1}{2!}\frac{d^2 f}{dx^2}(x - x_0)^2 + \frac{1}{3!}\frac{d^3 f}{dx^3}(x - x_0)^3 + \dots \quad (1)$$

Considering $x - x_0 = \Delta x$, then:

$$f(x_0 + \Delta x) = f(x_0) + \frac{df}{dx}\Delta x + \frac{1}{2!}\frac{d^2 f}{dx^2}\Delta x^2 + \frac{1}{3!}\frac{d^3 f}{dx^3}\Delta x^3 + \cdots \quad (2)$$

As we did previously, let's discard terms with exponents greater than 1 since they represent variations beyond the admitted precision. Therefore, it results in:

$$f(x_0 + \Delta x) \approx f(x_0) + \frac{df}{dx}\Delta x \quad (3)$$

Remembering that the variation in y is:

$$\Delta y = f(x_0 + \Delta x) - f(x_0) \quad (4)$$

then:

$$f(x_0 + \Delta x) - f(x_0) = \frac{df}{dx}\Delta x \quad (5)$$

and $\Delta y = \frac{df}{dx}\Delta x$, it is concluded that:

$$\Delta G_2 = \frac{df}{dG_1}\Delta G_1 \quad (6)$$

The Taylor series can be applied to a wide range of smooth functions, including polynomial, exponential, trigonometric, rational, and logarithmic functions. These functions, which can have a Taylor expansion, are mathematically defined as analytic functions. However, the ease of expansion and practical utility may vary, as some functions may require special considerations due to singularities or discontinuities. In summary, the Taylor series is a powerful tool, but its applicability depends on the specific nature of the function in question.

2.4.2. Monte Carlo Method

The Monte Carlo Method is statistical method involves simulating a large number of experiments (or calculations) using random distributions of the input variables with their respective uncertainties. The analysis of the results of these simulations provides an estimate of the uncertainty in the final value.

When dealing with error propagation, challenges arise from correlations between variables, which require modifying the standard formula to include covariance terms. Non-linear systems complicate error propagation, often necessitating numerical methods or simulations instead of linear approximations. Additionally, distinguishing between systematic errors, which can be corrected if identified, and random errors, which propagate as described, is crucial.

2.5. Systematic and Random Errors

In general, a measurement has imperfections that give rise to an error in the measurement result. Traditionally, an error is seen as having two components, namely, a random component and a systematic component. Error is an idealized concept and errors cannot be known exactly. The random error presumably originates from temporal or spatial, stochastic, or unpredictable variations in influential quantities. The effects of such variations, hereinafter referred to as random effects, are the cause of variations in repeated observations of the measurand. Although it is not possible to compensate for the random error of a measurement result, it can generally be reduced by increasing the number of observations; its expectation or expected value is zero.

The systematic error, like the random error, cannot be eliminated, but it too can often be reduced. If a systematic error originates from a recognized effect of an influential quantity in a measurement result, hereinafter referred to as a systematic effect, the effect can be quantified and, if significant with respect to the required accuracy of the measurement, a correction or correction factor can be applied to compensate for the effect. It is assumed that, after this correction, the expectation or expected value of the error caused by a systematic effect is zero. The relationship between errors can be seen in the figure 2.
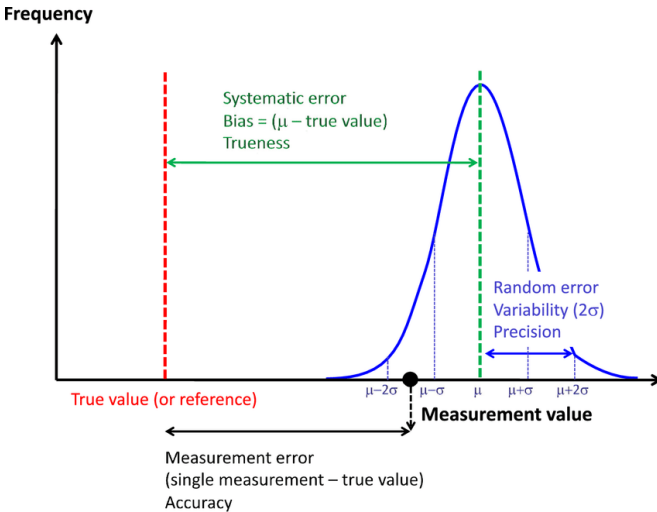

Figure 2: Equivalences between error theory in the context of physical sciences

2.6. Bias-Variance Trade-off Theory in Machine Learning

The regression problem is to construct a function $f(x)$ based on a "training set" $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ with the purpose of approximating y in future observations of x. This is sometimes called "generalization,". To explicitly show the dependence of f on the data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we will write $f(x; D)$ instead of simply $f(x)$. Given D, and given a particular x, a natural measure of the effectiveness of f as a predictor of y is

In this context, the effectiveness or quality of a model can be represented by the mean squared error $E[(y - f(x; D))^2 \mid x, D]$. This error can be decomposed into two parts:

$$E[(y - f(x; D))^2 \mid x, D] = \underbrace{E[(y - E[y \mid x])^2 \mid x, D]}_{(noise)} +$$

$$\underbrace{(f(x; D) - E[y \mid x])^2}_{(quadratic-distance)} \quad (7)$$

The first part on the right side of the equation represents the variance of y given x and does not depend on the data D or the estimator f, also known as "data noise". The second part on the right side measures the effectiveness of f as a predictor of y.

An important characteristic is that f depends on D, so a global evaluation of f takes into account the possible subsets D. Removing the noise (first part on the right side), we can write the expectation of the model in terms of an average of the possible subsets D:

$$E_D[(f(x; D) - E[y \mid x])^2] \quad (8)$$

where E_D represents the expectation with respect to the training set D, which is the average over all possible D.

For a particular training set D, $f(x; D)$ may be an excellent approximation of $E[y \mid x]$, thus being an almost optimal predictor of y. However, $f(x; D)$ may vary substantially for different realizations of D, or the average of $f(x; D)$ over all possible D may be far from the regression $E[y \mid x]$. These variations can contribute to large values in the mean squared error, making $f(x; D)$ an unreliable predictor of y.

These sources of estimation error are decomposed into bias and variance:

$$E_D[(f(x; D) - E[y \mid x])^2] = \underbrace{E_D[(f(x; D) - E_D[f(x; D)])^2]}_{(bias)}$$

$$+ \underbrace{(E_D[f(x; D)] - E[y \mid x])^2}_{(variance)} \quad (9)$$

model bias, that is, the average of the difference between the predicted value and the real (y) average value at a given x. Variance is the measure of variability between different results of f generated from different training sets D. It has practical significance within the context of the trade-off during model optimization. Once a model is defined for a given training set D, this term becomes zero. However, the bias remains, and along with the noise, it becomes a quantifier of the uncertainty or doubt regarding a predicted result for this defined model.

If, on average, $f(x; D)$ is different from $E[y \mid x]$, then $f(x; D)$ is considered biased. An unbiased estimator can still have a large mean squared error if the variance is high, as $f(x; D)$ can be highly sensitive to the data, leading to poor performance.[23]

There is often a trade-off between the contributions of bias and variance to the estimation error, which creates a

type of "uncertainty principle" [24]. Typically, variance is reduced through "smoothing", by combining, for example, the influences of samples that are close in the input space x. However, this introduces bias, as details of the regression function are lost; for example, sharp valleys and peaks are blurred.

The bias-variance decomposition is a way of analyzing the expected generalization error of the learning algorithm with respect to a specific problem as the sum of three terms: bias, variance, and an irreducible error resulting from noise in the problem itself [23, 25].

In general, the Mean Squared Error (MSE) encapsulates the main elements contributing to the discrepancy in predictions, suggesting that it can be a satisfactory measure of uncertainty. Nevertheless, relying solely on MSE as an indicator of uncertainty can lead to inaccuracies. The crucial issue lies in evaluating the adequacy and representativeness of the data used in training, as the complete capture of these three key terms may be compromised. For example, the way data is obtained can influence the lack of representativeness of error sources in MSE calculation. If data is collected at a single moment using measuring instruments, calibration uncertainty and temporal variation inherent in the instrument may not be adequately considered. Another situation arises when there is a limited amount of data, resulting in a lack of representativeness of the sample regarding the real characteristics of the problem, a phenomenon known in statistics as sampling bias. In summary, although MSE is a common metric for assessing the overall accuracy of predictions, its isolated use as a measure of uncertainty associated with each prediction may not accurately reflect the complexity and diversity of factors contributing to prediction errors.

2.7. Gaps in the literature

As noted by [26], current methods often struggle to effectively disentangle model uncertainty from data uncertainty, frequently confusing the two. Despite the popularity of Monte Carlo Dropout (MCDO) proposed by [27], Deep Ensembles (DE) by [28], and Hyper Deep Ensembles (HDE) introduced by [29], none of them can reliably capture all the essential aspects of model uncertainty. Additionally, obtaining epistemic uncertainty as part of a model prediction can be a fundamental safety requirement. However, the need for real-time inference from sensor data makes using MC Dropout a challenging proposition due to its high computational cost. [15] observes that there is still no comprehensive and structured evaluation of existing methods based on various real-world applications. While works like [30] have taken initial steps towards real-world evaluations, there is no standardized protocol for testing uncertainty quantification methods, as also pointed out by [10] and [12] in their work. Furthermore, there are no reference methods for uncertainty that can be used in these assessments. The author also highlights that a broad and structured comparison of existing methods for estimating uncertainty in real-world applications is still unavailable. Even the evaluation using real-world data is not common in current machine learning

research papers. Consequently, for a specific application, it remains uncertain which uncertainty estimation method works best and whether newer methods outperform older ones in real-world examples.

Current approaches for estimating uncertainty in neural networks require modifications to the network architecture and optimization process, tend to disregard prior data knowledge, and generally make oversimplified assumptions that underestimate uncertainty. Additionally, [31] mentions that traditional techniques for uncertainty estimation model network activations and weights through parametric probability distributions. However, these approaches are particularly challenging to train and are rarely applied in robotics scenarios.

There is another category of methods that estimate uncertainties through sampling. However, these methods often do not capture uncertainties in the data, resulting in overly confident predictions. Furthermore, sampling-based approaches tend to ignore any relationship between uncertainty in the data and uncertainty in the model, increasing the risk of underestimating uncertainties. Consequently, a sample input with a high level of noise should have a higher model uncertainty than the same sample with less noise. Monte Carlo sampling methods typically cannot adequately represent data uncertainty, especially due to factors like sensor noise.

In summary, the identified gaps in the literature related to uncertainty in machine learning models find answers in the contributions of this work, which offer an innovative, efficient, and well-founded method to address these complex issues. A summary is presented in Table 1.

3. Proposed of the Methodology

3.1. Overview

In this work, we address the challenge of uncertainty in prediction within the context of machine learning. To achieve this goal, we introduce the Analytical Propagation of Errors and Uncertainties (APU, for short) algorithm. The objective is to establish a robust mathematical foundation for estimating uncertainty in neural network predictions by combining uncertainties arising from the dataset and those originating from the chosen model. To achieve this goal, we carefully examine the nature of deep learning neural networks, specifically Multi-Layer Perceptrons (MLPs), identifying specific characteristics that have allowed us to establish a connection between the typical prediction model of a neural network and a mathematical measurement model, a concept extensively covered in established documents such as [22], which systematically delineates the propagation of uncertainties through measurement models. In fact, a neural network designed to predict a physically related variable can be considered a measurement model. Through this model, we were able to evaluate the specific characteristics of physical data, including uncertainties stemming from measurements.

Table 1
Relationship between gaps and proposed contributions

Gaps Identified in the Literature	Contributions
Current methods fail to achieve an adequate separation between model uncertainties and data uncertainties, often confusing them. [26].	Develop a method that can separately determine the uncertainty of the data, the model, and the combination of both.
Obtaining epistemic uncertainty as part of a model prediction can be a fundamental requirement in terms of functional safety. However, the need for real-time inference from sensor data makes the use of MC Dropout a challenging proposition due to its high computational cost [26].	Develop a computationally low-cost method capable of performing real-time (offline) inferences that can be applied in critical situations, such as a functional safety requirement.
According to [15], there is still no clear standardized protocol for testing uncertainty quantification methods, as also pointed out by [10] and [12] in their work. Additionally, there are no reference methods for uncertainty that can be used in these assessments.	Present the JCGM 101 method as a strong candidate for an uncertainty validation method within the domain of physical sciences.
Although deep neural networks are highly flexible and efficient, they do not directly incorporate domain-specific expertise, which is often available and can be described by mathematical or physical models. [15].	Physics-guided models offer several opportunities to integrate explicit knowledge and practical representations of uncertainty into a deep learning framework. In practice, it's essential to better assess the data used for training and testing, considering the uncertainties stemming from the measurement instruments used to obtain the data.
To address the issue of uncertainty in the data,[32]. proposed the inclusion of a "variance" variable in each output, which is trained using a maximum likelihood-based loss function, also known as heteroscedastic loss. Combining this approach with Monte Carlo sampling allows for the prediction of both model uncertainty and data uncertainty. However, it's essential to note that this method requires modifications to the network architecture due to the inclusion of variance and the use of heteroscedastic loss for training, which may not always be feasible.	Propose a methodology for estimating uncertainty in neural networks without requiring modifications to the network architecture and optimization process.
According to [10], it is noted that most of these studies do not provide relevant code and data, a problem that is also emphasized by [11].	Making the code available for future replications and analysis by the scientific community.

The instruments, which are generally overlooked when evaluating the reliability of neural network predictions. To estimate random uncertainty, instead of introducing arbitrary noise into the data, we propose estimating the measurement uncertainties of the instruments from which the data were acquired. This way, propagating them through the already trained network using the analytical method of error propagation theory. As for epistemic uncertainty, instead of creating probability distributions for the network weights (using a Bayesian method), we suggest an approach based on the model's performance. This involves decomposing prediction errors into a random component and a systematic component, while also considering the uncertainty associated with the reference value used to determine prediction errors (ground truth), since this directly affects the value of each determined prediction error.

Finally, to estimate predictive uncertainty, we propose a combination of random and epistemic uncertainties. One of the significant advantages of this approach is that it does not depend on computationally expensive simulations. Instead, it utilizes mathematical tools and an analytical approach. This is achieved by expanding the network's prediction model through a Taylor series, as per the analytical method of error propagation [22].

Another crucial point is that the method anticipates validation based on the [33], which, in summary, employs numerical methods and Monte Carlo simulation and is widely used to validate analytical error propagation methods for uncertainty determination in the realm of physical sciences. A comparison is planned, based on a maximum allowable tolerance, which is determined considering the number of significant digits for the expression of uncertainty. Additionally, we will address some possible steps if the difference between methods exceeds the initially stipulated tolerance.

3.2. APU

One of the characteristics of the APU method is its modularity, allowing its application to any pre-trained MLP neural network. It is understood that a trained neural network has preferably undergone the entire optimization process related to the bias-variance dilemma, using various techniques such as cross-validation and regularization to adjust the model parameters and achieve the best balance between bias and variance. In other words, a significant effort has already been made to systematically reduce the model's uncertainty, characterized by bias-variance. After this process, an optimized model is obtained, which will be used for new predictions. At this point, the APU method can be applied. By knowing the performance of the optimized model, the APU combines the different uncertainties related to the model and is possibly the only method so far capable of translating the uncertainty arising from data noise by examining or estimating the metrological characteristics of the sensors used for data acquisition and propagating them through the model, and
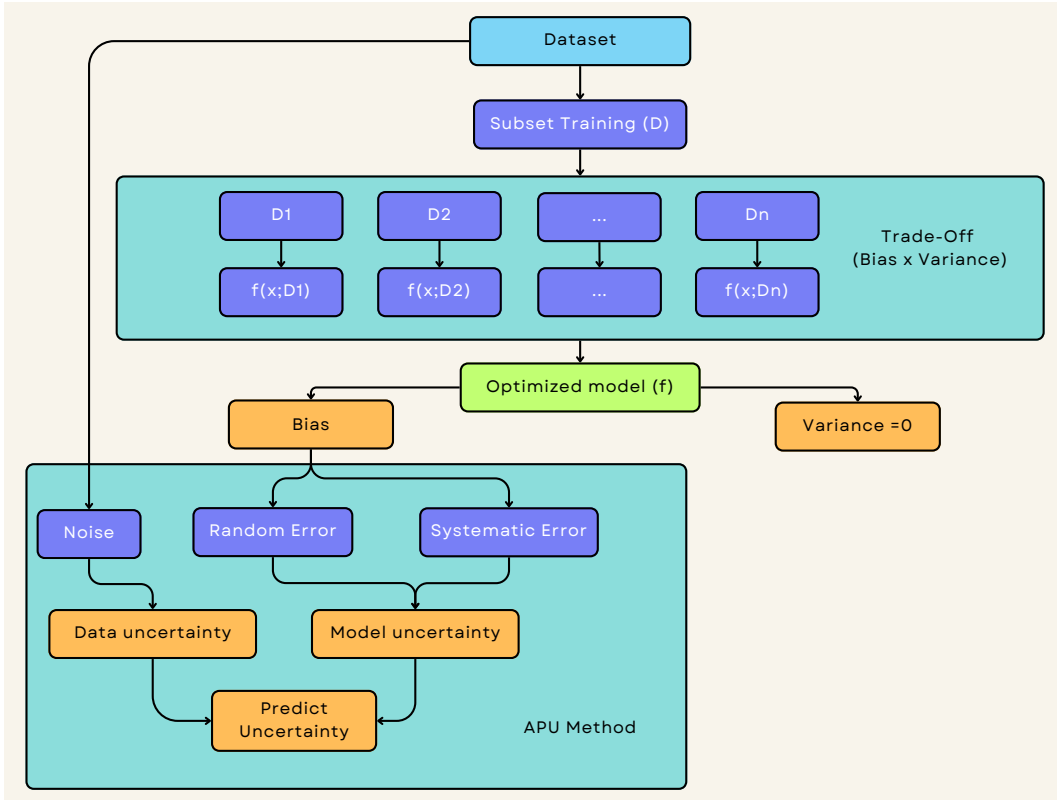
Figure 3: Validation flowchart of the APU method.

combining them with model uncertainties. A diagram can be seen in Figure 3.

The method will be presented sequentially, addressing specific fundamental issues at each stage. An algorithm consisting of 13 stages will be proposed, covering all the steps of the method. Essentially, the stages are as follows:

1. Acquisition of the mathematical model (F), defined in this context as the function generated by the result of training a neural network.
2. Determining the random error (RE) and systematic error (SE) of the network based on the prediction errors of all data (test and training).(Algorithm 1,2 and 3).
3. Determination of data uncertainty
   (a) Evaluation of the uncertainty matrix associated with each model input, through the analysis of the calibration of sensors that produced the data for training and testing.(Algorithm 4).
   (b) Determination of the matrix of partial derivatives of y with respect to each input x, using the numerical method of finite differences, given that the function cannot be explicitly known.(Algorithm 5).
   (c) Determination of the correlation matrix between input variables x using the data used for training and testing.(Algorithm 6).
   (d) Obtaining the contributions matrix, resulting from the combination of the previous matrices. (Algorithm 7).

(e) From the resulting matrix from the previous item, determining the random uncertainty (Algorithm 8).

(f) Determination a set of test vectors and validating the uncertainty of the data through comparison with the Monte Carlo simulation method (JCGM 101:2008). This validation is based on a tolerance criterion based on the number of significant digits expected for the problem.We will address this stage in greater detail in section 3.3.

4. Determination of epistemic uncertainty, considering the random error.
5. Determination of prediction uncertainty by combining random and epistemic uncertainties.(Algorithm 9).
6. Determination of expanded uncertainty, incorporating systematic error (SE) when applicable, and multiplying by the k_p factor for a specific coverage probability.(Algorithm 10).
7. Determination of expanded uncertainty, incorporating systematic error (SE) when applicable, and multiplying by the kp factor for a specific coverage probability.(Algorithm 10).
8. Validation of the expanded uncertainty result through the coverage interval test. Establish a significance level, for example, 95%, and assess the percentage of true values from the test group vectors that fall within the interval stipulated by the uncertainty.The validation flow of the method is illustrated in Figure 5.
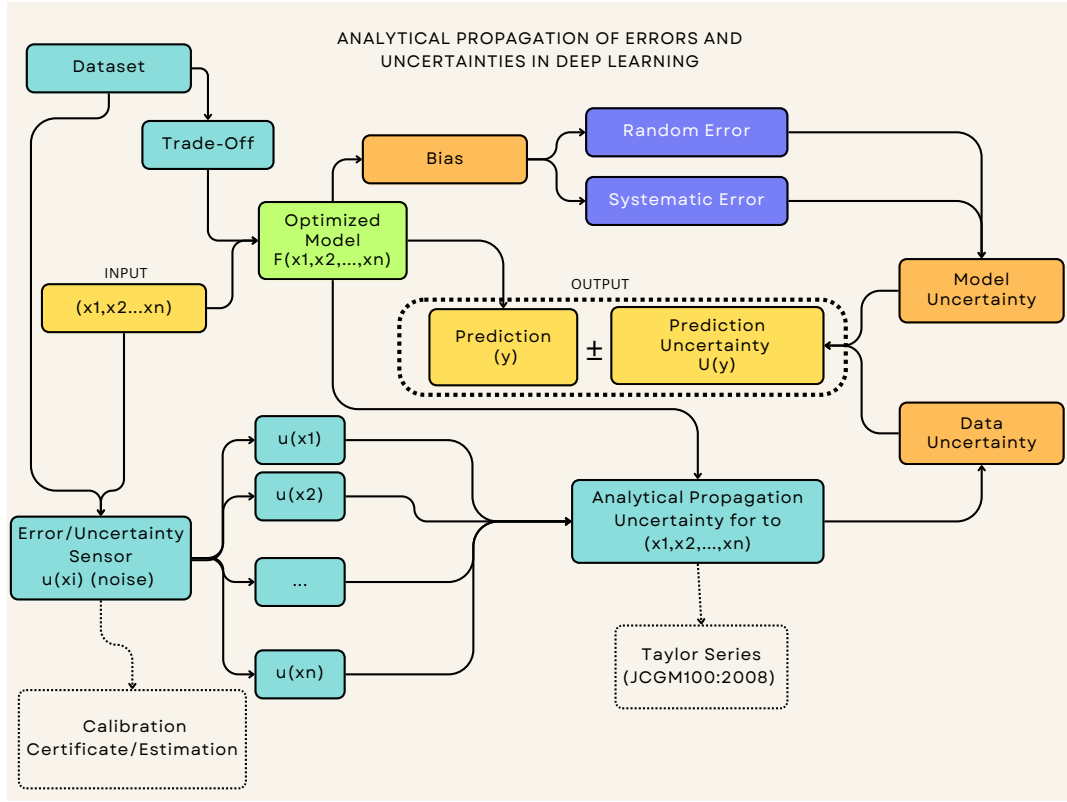
Figure 4: Description of the inputs and outputs in the APU algorithm.

Details of the methodology are presented throughout section 3. The general scheme can be seen in figure 4.

3.2.1. Measurement function

The APU method is based on the analytical theory of error propagation, using a local approximation by Taylor series, as mentioned in 2.4.1. Therefore, it is necessary to demonstrate that a trained MLP neural network can be viewed as an analytical function. Without loss of generality, every trained MLP neural network can be mathematically represented by a function F, defined as a composition of activation functions, as briefly described in equation 10. It is easy to demonstrate that the composition of n analytical functions is also analytical (Appendix A). Therefore, it follows that F is analytical and can be locally expressed by a Taylor series, provided that its activation functions are analytical.

$$F(x) = \varphi_L(W_L \cdot \varphi_{L-1}(W_{L-1} \cdot \ldots \varphi_1(W_1 \cdot x + b_1) \ldots + b_{L-1}) + b_L) \tag{10}$$

where L are the layers, W are the weights, b are the biases, and φ are the activation functions in each layer.

It is also possible to consider non-analytical activation functions by replacing them with polynomials (which are analytical). According to [34], every activation function can be approximated by a polynomial function, which allows neural networks to be compared to polynomial regressions. Transcendental activation functions are often computationally approximated by polynomials (Taylor series or Padé approximations). This was recommended in [35] for general use and was utilized in neural network computations in firmware in [36]. For generic functions, the Stone-Weierstrass Theorem [37] asserts that every continuous function can be approximated by a polynomial. For further details, refer to [34].

3.2.2. Data Uncertainty

So far, in the literature, data uncertainty has been mainly addressed in terms of "noise," also known as random uncertainty. To estimate this uncertainty, it is common to add noise to the data before training and testing, so that this uncertainty can propagate through the network during training. However, this research proposes a more accurate approach to estimate such noises, which are often scaled and added arbitrarily. This is done by evaluating the instruments and conditions during data collection.

It is natural that field measurements are affected by environmental conditions, errors, and uncertainties of the measuring instruments. When more reliable data is desired, it is crucial that the instruments be calibrated by competent laboratories, where the errors and uncertainties of the instrument are identified under their best conditions of use. These characteristics are usually evaluated and documented in a calibration certificate, which not only provides the error and uncertainty over the entire range of the instrument but
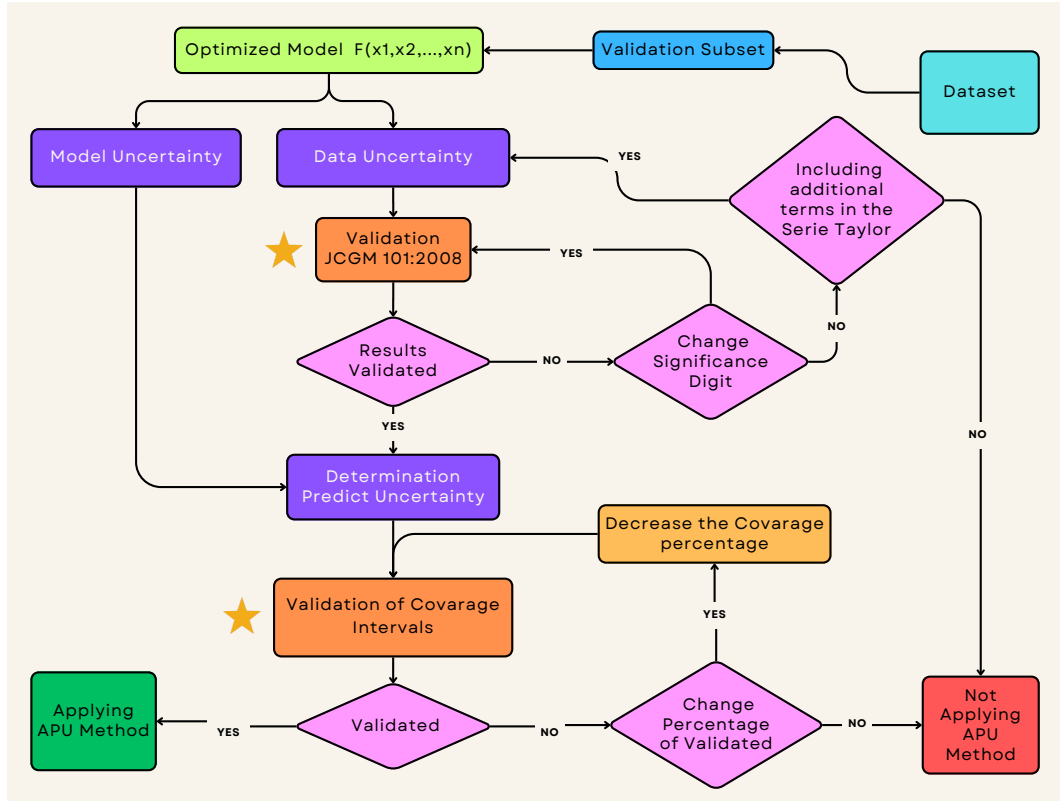
Figure 5: Validation flowchart of the APU method.

also presents the probability distribution of the uncertainty associated with a confidence level. This information can be easily obtained through critical analysis of the calibration certificate. More details regarding information contained in a calibration certificate can be found in Appendix A.

In an application where the input vector of a model has dimension k, i.e., $\mathbf{v} = (x_1, x_2, \ldots, x_k)$, an uncertainty associated with each component of this vector is determined in advance through analysis of calibration certificates, instrument testing, or estimates based on prior knowledge of measurements. These uncertainties are subsequently consolidated into a matrix referred to here as the uncertainty matrix $M1$. For example, consider the model represented by the function $y = F(x_1, x_2, \ldots, x_k)$. The matrix $M1$ will then be defined as follows:

$$M1 = \begin{bmatrix} u^2(x_1) & u(x_1).u(x_2) & u(x_1).u(x_3) & \ldots & u(x_1).u(x_k) \\ 0 & u^2(x_2) & u(x_2).u(x_3) & \ldots & u(x_2).u(x_k) \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & u^2(x_k) \end{bmatrix}$$

I- Evaluation of Partial Derivatives

Since it is not feasible to obtain and work explicitly with the function $F(x) = y$ generated during the training of a Neural Network, it is necessary to resort to numerical methods. These methods are applied when we have only a set of points belonging to the function, without knowing its analytical form. To calculate first and second-order partial

derivatives more accurately, we employ approaches based on the concept of "perturbation". Among these approaches, precise mathematical methods found in numerical calculus stand out, such as the finite difference method or more advanced techniques like interpolation with Lagrange polynomials. In the APU method, we will employ the technique of progressive finite differences (equation 9). Further details can be found in [38].

$$\frac{\partial y}{\partial x_1} = \frac{f(x_1 + h, x_2, \ldots x_n) - f(x_1, x_2, \ldots, x_n)}{h} \quad (11)$$

with "h" sufficiently close to zero.

Thus, for each input variable x, the derivative of y with respect to x is calculated. These derivatives are subsequently consolidated into a matrix referred to here as the partial derivatives matrix M2:

$$M2 = \begin{bmatrix} (\frac{\partial y}{\partial x_1})^2 & 2.\frac{\partial y}{\partial x_1}.\frac{\partial y}{\partial x_2} & 2.\frac{\partial y}{\partial x_1}.\frac{\partial y}{\partial x_3} & \ldots & 2.\frac{\partial y}{\partial x_1}.\frac{\partial y}{\partial x_k} \\ 0 & (\frac{\partial y}{\partial x_2})^2 & 2.\frac{\partial y}{\partial x_2}.\frac{\partial y}{\partial x_3} & \ldots & 2.\frac{\partial y}{\partial x_2}.\frac{\partial y}{\partial x_k} \\ 0 & 0 & (\frac{\partial y}{\partial x_3})^2 & \ldots & 2.\frac{\partial y}{\partial x_3}.\frac{\partial y}{\partial x_k} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & (\frac{\partial y}{\partial x_k})^2 \end{bmatrix}$$

II-Evaluation of Input data Covariance

The presence and importance of correlations between input quantities cannot be ignored. The associated covariances must be experimentally evaluated whenever possible.

It is necessary to rely on intuition based on previous experiences and general knowledge when estimating the degree of correlation between input quantities, which may result, for example, from the effect of common influences such as ambient temperature, barometric pressure, and humidity, when applied in the physical sciences.

It is common practice to insert "noise" into the training data of the neural network so that this effect can be propagated in the learning process. However, this noise is usually inserted independently, ignoring the fact that input variables may be strongly correlated. This practice can introduce somewhat artificial noise as it ignores these possible correlations. It is possible to estimate such correlations through the statistical obtaining of these coefficients, as per Equation 10, and covariance, as per Equation 11.

$$\rho_{ij} = \frac{\sum_{i=1}^{n}(x_i - \bar{x}_i)(x_j - \bar{x}_j)}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2 \sum_{i=1}^{n}(x_j - \bar{x}_j)^2}} \quad (12)$$

$$cov(x_i, x_j) = u(x_i).u(x_j).\rho_{ij} \quad (13)$$

These correlations indicate that as variations occur in one variable, these variations will also occur in some other correlated variable, and they will be simultaneously propagated through the neural network. These uncertainties may either amplify or cancel out, depending on the magnitude and sign of the partial derivative and the correlation coefficient of these variables.

After determining all correlation coefficients among the variables x, a matrix of correlation coefficients M3 is established:

$$M3 = \begin{bmatrix} \rho_{(x_1,x_1)} & \rho_{(x_1,x_2)} & \rho_{(x_1,x_3)} & \rho_{(x_1,x_3)} & \rho_{(x_1,x_k)} \\ 0 & \rho_{(x_2,x_2)} & \rho_{(x_2,x_3)} & \cdots & \rho_{(x_2,x_k)} \\ 0 & 0 & \rho_{(x_3,x_3)} & \cdots & \rho_{(x_3,x_k)} \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & \cdots & \rho_{(x_k,x_k)} \end{bmatrix}$$

III-Evaluation of Data Uncertainty (u_D^2)

For each input vector $\mathbf{v}_i = (x_1, x_2, \ldots, x_n)$, the uncertainty of the data is determined by combining the uncertainties of each component x_i of the vector $\mathbf{v}_i$. This combination is determined by multiplying the elements corresponding to the rows and columns of matrices $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$, forming the matrix $\mathbf{M}_4$ according to the expression:

$$\begin{bmatrix} (\frac{\partial y}{\partial x_1})^2.u^2(x_1).\rho_{(x_1,x_1)} & 2.\frac{\partial y}{\partial x_1}.\frac{\partial y}{\partial x_2}.u(x_1).u(x_2) & \cdots & 2.\frac{\partial y}{\partial x_1}.\frac{\partial y}{\partial x_k}.u(x_1).u(x_k)\rho_{(x_1,x_k)} \\ 0 & (\frac{\partial y}{\partial x_2})^2.u^2(x_2).\rho_{(x_2,x_2)} & \cdots & 2.\frac{\partial y}{\partial x_2}.\frac{\partial y}{\partial x_k}.u(x_2).u(x_k)\rho_{(x_2,x_k)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & (\frac{\partial y}{\partial x_k})^2.u^2(x_k).\rho_{(x_k,x_k)} \end{bmatrix}$$

Replacing $u(x_i).u(x_j).\rho_{ij}$ with $cov(x_i, x_j)$ according to equation 11, and summing all terms in M4, we have:

$$u_D^2(v_0) = \sum_{i=1}^{k}[\frac{\partial y}{\partial x_i}(v_0).u(x_i)]^2 +$$
$$2.\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\frac{\partial y}{\partial x_i}(v_0).\frac{\partial y}{\partial x_j}(v_0).cov(x_i,x_j) \quad (14)$$

The data uncertainty (u_D^2) is determined for input vector (v_0) by adding to the previous expression the uncertainty associated with obtaining the y data, that is, $u^2(y)$ that represents the uncertainty of the data regarding the target value. Such results also carry an uncertainty as they were obtained in a similar manner to the x_i, that is, through measurement instruments. The final expression is then determined as equation 13:

$$u_D^2(v_0) = \sum_{i=1}^{k}[\frac{\partial y}{\partial x_i}(v_0).u(x_i)]^2 +$$
$$2.\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}\frac{\partial y}{\partial x_i}(v_0).\frac{\partial y}{\partial x_j}(v_0).cov(x_i,x_j) +$$
$$u^2(y) \quad (15)$$

The equation 13 is based on the local approximation by Taylor Series of the function F and is widely used in physics, specifically in the field of metrology. Further details can be found in JCGM 100:2008.

3.2.3. Model Uncertainty (u_M)

The uncertainty related to the model (epistemic) will be estimated by determining two components: systematic error (SE) and random error (RE). In other words, in the APU method, our approach aims to estimate the model's uncertainty by evaluating the prediction errors obtained from all the data used in training and testing. These errors, in the context of regression, are referred to as residuals. In a typical regression, it is expected that the mean of all residuals approaches zero, and their distribution is approximately normal. This calculated mean (Equation 14) represents the systematic error of the model, also known as systematic bias, which refers to a type of error consistent or systematic in all measurements or observations. This consistency is associated with the degree of approximation of the error distribution to the normal curve, so it is important to check this distribution to better assess the performance of the trained model. This error reflects the trend of the model, as illustrated in Figure 3 and described in Algorithms 1 and 2.

$$SE = \frac{1}{n}\sum_{i=1}^{n}\Delta_i = E[F(x_i) - y_i] \quad (16)$$

where $F(x_i)$ represents the predictions generated by the trained model, y_i represents the actual observed value, and n represents the number of dataset instances.

On the other hand, random error, also known as aleatory variability, is associated with the precision of the model, as represented in Figure 3. This error refers to a type of error that occurs unpredictably and irregularly, resulting in random variations in the results of different measurements or observations. This error can be determined by the standard deviation calculated around the systematic error, as shown in Equation 153 (Algorithm 3).

$$RE^2 = \frac{1}{n} \sum_{i=1}^{n} ((F(x_i) - y_i) - SE])^2 \qquad (17)$$

where y_i represent the predictions generated by the model and Es represent the Systematic error. Figure 4 illustrates the relationship between random and systematic errors Next, we will formally present the calculation of epistemic uncertainty according to APU:

The model uncertainty is determined independently for each input vector according to Equation 16.

$$u_M = RE \qquad (18)$$

3.2.4. Prediction Uncertainty (u_P)

The combined prediction uncertainty $uc_P(v_0)$ is composed of the combination of the random uncertainty and the model uncertainty, as given by Equation 15:

$$u_P^2(v_0) = u_D^2(v_0) + u_M^2 \qquad (19)$$

With the expression shown in equation 16, it is possible to determine the expanded uncertainty (U) for each input vector v with the predicted value $y(v)$. This is achieved by taking the square root of u_p^2, multiplying it by the coverage factor $k_p = 2$. The factor k_p is the coverage factor related to the desired probability p. It is known that, for an approximately normal distribution and a probability of 95%, 2 standard deviations are required on each side of the mean., and finally adding the epistemic uncertainty (SE) according to Equation 18.

$$U_P^2(v_0) = k_p \cdot \sqrt{u_P^2(v_0)} + SE \qquad (20)$$

3.2.5. Necessary conditions for the application of the technique

Some conditions are necessary for the application of the proposed mathematical method:

i) The function F generated during model training is analytic;

ii) The x_i involved in significant higher-order terms of a Taylor series approximation for $f(x)$ are independent;

iii) The probability density functions (PDFs) assigned to the x_i involved in higher-order terms of a Taylor series approximation for $f(x)$ are Gaussian;

iv) Higher-order terms that are not included in the Taylor series approximation for $f(x)$ are negligible.

Condition i) is discussed in 3.2.1. As for conditions ii), iii), and iv), these depend on prior knowledge of the physical phenomena in question, the need to expand the series for more terms, and the measuring instruments used. Since these conditions cannot always be met, the validation process described in 3.3 is crucial.

3.3. Validation of the APU method

The APU method includes two validation stages in its implementation. The first test is related to data uncertainty, while the second is related to predictive uncertainty, which incorporates model uncertainty. These procedures are essential for practical application in specific critical scenarios. From this validation, it can be concluded whether the APU method is suitable for the physical problem in question. For this, a validation group must be selected, usually the same used in the training stage, where the goal is to optimize the model by minimizing bias and variance according to the trade-off theory mentioned in section 2. Next, we will discuss each of these validation stages in more detail. A validation flowchart can be seen in Figure 5.

3.3.1. Validation of the data uncertainty through comparison with the JCGM 101

It is always expected that the APU uncertainty methodology will function satisfactorily in many situations. However, determining whether all the conditions necessary for its proper functioning are valid is not a simple task. In fact, the degree of difficulty involved in this determination is generally considerably greater than what is required to apply the Monte Carlo method (JCGM 101), as long as appropriate software is available. Therefore, since these conditions cannot be readily tested, all cases should be validated [22]. Since the domain of validity of the Monte Carlo Method is broader than that of the APU uncertainty methodology, it is recommended that both be applied, followed by a comparison of the results. If this comparison is favorable, the APU uncertainty methodology can be used in this case and in sufficiently similar future problems. Otherwise, the use of the Monte Carlo Method or another appropriate approach should be considered.

The objective of the comparison procedure is to determine, within a stipulated numerical tolerance, whether there is agreement between the coverage intervals obtained by the APU and JCGM 101 uncertainty methodologies. This numerical tolerance is assessed in terms of the limits of the coverage intervals and corresponds to the tolerance obtained when expressing the standard uncertainty $u(y)$ with what can be considered an appropriate number of significant decimal places. The procedure is as follows:

- Establish a numerical tolerance associated with $u(y)$ as half of the last significant digit of the uncertainty;

- Compare the coverage intervals obtained by the APU and 101 uncertainty methodologies to determine if the

required number of correct decimal places has been obtained in the coverage interval provided by the APU uncertainty methodology.

$$d_{inf} = |y - U_p - y_{inf}| \qquad (21)$$
$$d_{sup} = |y + U_p - y_{sup}| \qquad (22)$$

That is, the absolute differences between the respective limits of the two coverage intervals. So, if both d_{inf} and d_{sup} are not greater than δ, the comparison is favorable, and the APU uncertainty methodology will have been validated in this case.

Denoting n_{dig} as the number of significant decimal digits in any numerical value q, the numerical tolerance δ associated with q is obtained as follows:

- express q in the form $c.10^l$, where c is an integer with n_{dig} decimal digits and l is an integer;

- assume

$$\delta = \frac{1}{2}.10^l \qquad (23)$$

I-Propagation of Input Variable PDFs Through the Neural Network

The quality of the probability distributions of the output variables is directly related to the level of knowledge about the data. For each of the input vectors of the model (vi) belonging to the test data, it is recommended to use a minimum of $M = 10^6$ vectors, distributed according to the Probability Density Function (PDF) around the values of each component of the vector.

II-Cumulative Distribution Function and Coverage Interval for the Output Quantity

A coverage interval for each vector is determined. After defining the TI (Tolerance Interval) function, which sets the size of the 95% probability range intervals, it becomes possible to identify the minimum range by determining the minimum and maximum values (y_{inf} and y_{sup}) through the inverse calculation of the cumulative function of the probability distribution. This enables the identification of the values of d_{inf} and d_{sup}, allowing for their evaluation in accordance with the tolerance established in equation 18.

3.3.2. Validation of the Coverage Interval Validation

The expanded predictive uncertainty is a coverage interval constructed around the network's prediction value, which is expected to include the target value in 95% of cases. This construction is based on multiplying the predictive uncertainty by a factor k_P related to the required coverage level. The higher the coverage probability, the greater this factor will be. The purpose of validation at this stage is to assess the accuracy of this constructed interval, as it aims to achieve 95% effectiveness when tested. In this manner, a simple test is conducted for the entire validation group. For each predicted value in this group, the coverage interval

is determined around this prediction, and it is evaluated whether the target value falls within this interval. As previously mentioned, a success rate of at least 95% is expected.

It is important to note that the choice of coverage percentage is associated with the criticality established by the specific application and is determined by the user. These intervals are essential for associating the risks involved in using these predicted data in real situations. If the method does not achieve the desired probability, it can be assessed whether it is possible to lower this probability. If so, the entire validation process can be conducted again under a new probability k_P. Otherwise, the use of this method is discarded. Details can be seen in the flowchart in Figure 5.

3.4. Algorithm Configuration

To better exemplify the proposed methodology, we present all the steps divided into 10 algorithms, with the ultimate goal of providing a 95% coverage interval for $F(v_i)$ given a v_i.

Algorithm 1 Obtaining the Prediction Error e_i

1: $m \leftarrow$ number of elements in the list (train +test)
2: $list \leftarrow$ empty list
3: **for** $i = 1$ to m **do**
4:     $element \leftarrow (F(v_i) - y_i) = e_i)$
5:     To add $element$ à $list$
6: **end for**
7: **Out:** $list(e_i, i = 1...m)$

Algorithm 2 Obtaining Systematic Error (SE)

1: $m \leftarrow$ number of elements in the list (train +test)
2: $element \leftarrow \frac{\sum_{i=1}^{m}(e_i)}{m} = SE$
3:
4: **Output:** SE

Algorithm 3 Obtaining Model Uncertainty (u_M)

1: $m \leftarrow$ number of elements in the list (train +test)
2: $element \leftarrow \sqrt{\frac{1}{n}\sum_{i=1}^{n}((F(x_i) - y_i) - SE)^2} = RE$
3:
4: **Output:** $RE = u_M$

---

4. Experiment Evaluation

4.1. Methodology

To evaluate the effectiveness of the method, training will be conducted for a neural network on each of the selected datasets. The first dataset will be "Airfoil Noise," and the other will be "Concrete Compressive Strength," both containing physical variables and made available by the UCI Repository.The proposed approach involves the analysis of a random set of labeled test data. The method will be applied to each element of this set, and a confidence interval around

Algorithm 4 Obtaining Partial Derivatives c_{ki}

1: $h \leftarrow$ set value
2: $v_i \leftarrow$ value to be predicted
3: $y_i \leftarrow$ prediction result
4: $c_k \leftarrow$ determine partial derivative for all x_{ji} para todo j=1...k
5: Determine $F(x_i j)$ para todo j=1...k
6: $F(x_{1i}) \leftarrow F_\theta((x_{1i} + h, x_{2i}, ...x_{ki})$
7: $F(x_{2i}) \leftarrow F_\theta((x_{1i}, x_{2i} + h, ...x_{ki})$
8: ...
9: $F(x_{ki}) \leftarrow F_\theta((x_{1i}, x_{2i}, ...x_{ki} + h)$
10: Determine os coeficientes de sensibilidade c_{ki}
11: $c_{1i} \leftarrow \frac{(F(x_{1i}) - y_i)}{h}$
12: $c_{2i} \leftarrow \frac{(F(x_{2i} - y_i)}{h}$
13: ...
14: $c_k(i) \leftarrow \frac{(F(x_{ki}) - y_i)}{h}$
15: **Output:** Matrix C_{1xk} : (c$_{1i}$,c$_{2i}$,...,c$_{ki}$)

Algorithm 5 Constructing a New Sensitivity Coefficient Matrix (M1)

1: Original matrix: C_{1xk}
2: New matrix: M_{kxk}
3: **for** $i \leftarrow 1$ to number of rows in C_{1xk} **do**
4:     **for** $j \leftarrow 1$ to number of columns in C_{1xk} **do**
5:         $M_{kxk}[i][j] \leftarrow c_{1i} \cdot c_{1j}$ from C_{1xk}
6:     **end for**
7: **end for**
8: Print matrix M_{kxk}

Algorithm 6 Obtaining covariance matrix V_{kxk}

1: $upi \leftarrow$ set values
2: $upx_i \leftarrow$ determine standard uncertainties for i=1...k
3: **Output:** Matrix I_{1xk}: (up$_1$,up$_2$,...,up$_k$)
4: Constructing matrix V_{kxk}
5: Original matrix: I_{1xk}
6: New matrix: V_{kxk}
7: **for** $i \leftarrow 1$ to number of rows in I_{1xk} **do**
8:     **for** $j \leftarrow 1$ to number of columns in CI_{1xk} **do**
9:         $V_{kxk}[i][j] \leftarrow up_{1i} \cdot up_{1j}$ from I_{1xk}
10:     **end for**
11: **end for**
12: Print matrix V_{kxk}

Algorithm 7 Construction of Upper Triangular Matrix with 1s T_{kxk} (M2)

1: Matrix of size k
2: Matrix T_{kxk} upper triangular with all elements = 1
3: **for** $i \leftarrow 1$ to k **do**
4:     **for** $j \leftarrow i$ to k **do**
5:         $T[i][j] \leftarrow 1$
6:     **end for**
7: **end for**
8: Print matrix T_{kxk}

Algorithm 8 Constructing Uncertainty Matrix U_{kxk} (M3)

1: Original matrix 1: M_{kxk}
2: Original matrix 2: V_{kxk}
3: Original matrix 3: T_{kxk}
4: New matrix: U_{kxk}
5: **for** $i \leftarrow 1$ to number of rows in $M_{kxk}, V_{kxk}, T_{kxk}$ **do**
6:     **for** $j \leftarrow 1$ to number of columns in $M_{kxk}, V_{kxk}, T_{kxk}$ **do**
7:         **if** $i = j$ **then**
8:             $U_{kxk}[i][j] \leftarrow m_{ij} \cdot v_{ij} \cdot t_{ij}$ from M_{kxk}, V_{kxk}, and T_{kxk} respectively
9:         **else**
10:
11:             $U_{kxk}[i][j] \leftarrow 2.m_{ij} \cdot v_{ij} \cdot t_{ij}$ from M_{kxk}, V_{kxk}, and T_{kxk} respectively
12:         **end if**
13:     **end for**
14: **end for**
15: Print matrix U_{kxk}

Algorithm 9 Calculating Data Uncertainty $u_D(v_i)$

1: Declare matrix size $k \times k$
2: Declare matrix: U_{kxk}
3: Declare variable $sum \leftarrow 0$
4: **for** $i \leftarrow 1$ to n **do**
5:     **for** $j \leftarrow 1$ to m **do**
6:         Read value $M[i][j]$
7:         $sum \leftarrow sum + M[i][j]$
8:     **end for**
9: **end for**
10: Print $sum = u_D(v_i)$

Algorithm 10 Obtaining Expanded Uncertainty (Up)

1: $up(y) \leftarrow$ set values
2: $element \leftarrow U_p(v_i) = \pm(2.\sqrt{u_D^2(v_i) + u_M^2 + u(y)^2} + Es)$
3:
4: **Output:** $(\pm U_p)$

---

the prediction will be determined. With a 95% probability that the true information (label) is within this interval, it is expected that 95% of the intervals calculated by the APU contain the true value (ground truth). If this is achieved, we consider that the method has fulfilled its purpose. Validation through Monte Carlo simulation (JCGM 101) and comparison with the popular MCDropout will be crucial to assess whether the uncertainty estimated by the APU was not overestimated, a risk that could distort the evaluation of the method's performance.

4.2. Benchmarks

For comparison validate model-related uncertainty, the MC Dropout method by Gal and Ghahramani (2016) will be used. Bayesian approaches are known for providing a robust theoretical framework for model-related uncertainty estimation, incorporating prior information and updating it

Table 2
Measurement instrument uncertainties estimation*.

Symbol	Variable	Unit	Instrument	Calibration Uncertainty	Distribution	Divider	Standard Uncertainty $u(x_i)$
x1	Frequency	Hz	Sound level meter	±3.366	t-Student	kp=2	±1.683
x2	Angle	°	Goniometer	±0.04	t-Student	kp=2	±0.02
x3	Length	m	Pachymeter	±0.0002	t-Student	kp=2	±0.0001
x4	Velocity	m/s²	Speed measurement system based on pressure difference (pitot tubes)	±0.3762	t-Student	kp=2	±0.1881
x5	Thickness	m	Estimate determined by calculation	±0.003	t-Student	kp=2	±0.00145
y	Pressure	dB	Decibel meter	±0.15	t-Student	kp=2	±0.075 ($u(y)$)

*For further details, please refer to JCGM 100:2008.

based on observed data. The Monte Carlo Dropout method is a technique that uses Dropout during the inference phase to obtain multiple samples from the model and then uses these samples to estimate the posterior distribution of the model's parameters.

5. Results

To estimate the random uncertainties, it was necessary to investigate the information available about the dataset to assess the conditions under which the data was obtained. By examining [39],for example, an article that details the experiment that generated the dataset, it is possible to find details about data acquisition, some uncertainties associated with the measured or calculated variables, or at least estimate the related uncertainty based on the measurement system. For more details, please refer to the mentioned document.

It is natural that field measurements are affected by environmental conditions, errors, and uncertainties of the measuring instruments. When more reliable data is desired, it is crucial that the instruments be calibrated by competent laboratories, where the errors and uncertainties of the instrument are identified under the best conditions of use. These characteristics are usually evaluated and documented in a calibration certificate, which not only provides the error and uncertainty over the entire range of the instrument but also presents the probability distribution of the uncertainty associated with a confidence level. This information can be easily obtained through critical analysis of the calibration certificate. More details about the information contained in a calibration certificate can be found in Appendix A. For the purpose of this article, typical instruments used in wind tunnels were investigated, and the uncertainties were estimated based on this investigation. Table 2 summarizes this information.

As mentioned earlier, the method was applied to all vectors in the randomly selected test group. For each vector, uncertainties related to input variables were estimated and propagated through the neural network. The results can be seen illustratively in Figure 6. A summary table with all the results can be found in the appendix (Table 6).

5.1. Validation of the random uncertainty through comparison with the JCGM 101

For each of the input vectors of the model (v_i) belonging to the test data, M $=10^6$ vectors were simulated, distributed according to the PDF around the values of each component of the vector, as shown in Table 13. A normal distribution was considered for all input variables, centered at the values of each component (x_i) of the vector (v_i), with the standard deviation being the uncertainty $u(x_i)$ determined for each component (x_i) according to Table 2.

At the time of data generation, all correlations between input variables were considered. A multivariate distribution was used when generating the data.

All values were propagated through the neural network, resulting in the output distribution. It is essential to emphasize that the Monte Carlo method provides an estimate only of random uncertainty, i.e., that associated with the data. Therefore, in order to validate the proposed method, a comparison will be made between the analytically calculated random uncertainty according to [22] and the Monte Carlo simulation method [33]. All the elements drawn for the test group were selected for this validation.

5.2. Validation of the coverage interval

The quality of the probability distributions of the output variables is directly related to the level of knowledge about the data. It is known that in this case, the data were acquired from typical wind tunnel measurement instruments. The distributions adopted as shown in Table 3.

5.3. Discussion

Analyzing Figure 6, it is evident that the most significant uncertainty is related to the model, represented by the green area. It can be observed that the data uncertainty (red area) is much smaller compared to that of the model. This is plausible, considering that the uncertainties related to the data were estimated based on high-precision instruments, typical for wind tunnel contexts.

This separate observation of uncertainties allows for a critical analysis of the results, enabling an evaluation, firstly, of whether the magnitude of uncertainty meets a specific purpose. If corrective measures are necessary to reduce it, the most appropriate approach would be to improve the
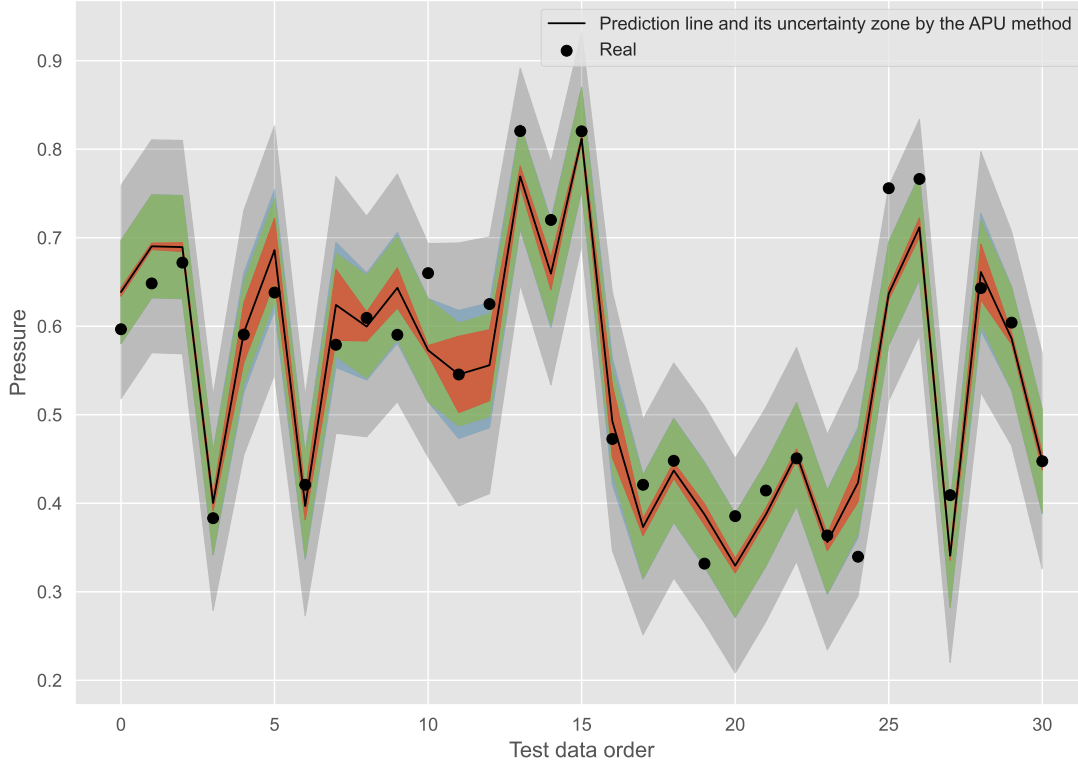
Figure 6: The figure illustrates the overlap of four regions around the predicted value by the model represented by the line on the graph. These regions were obtained through the **APU Method** and represent, respectively, the random uncertainty, indicated in red, the epistemic uncertainty, indicated in green, and the predictive uncertainty, indicated in blue, which essentially equals the epistemic uncertainty, making it difficult to visualize. The predictive uncertainty is a combination of the two former uncertainties. The indicated in gray represents the expanded predictive uncertainty for a coverage interval of 95.45%. This means that this calculated interval should encompass 95.45% of the actual points represented by the black dots, approximately 30 out of the 31 tested points. In practice, it can be observed that the gray area met expectations, covering 100% of the actual points.

model's performance, as the greatest source of uncertainty comes from the model itself.

The expanded predictive uncertainty (gray area) theoretically corresponds to 95% of the data. For a set of 31 elements, only 1 could fall outside this interval. By observing Figure 7, it is verified that all points are well distributed within the uncertainty zone, corresponding to a 95% confidence interval. This result reflects excellent performance, indicating that the uncertainty as a whole was neither underestimated nor overestimated. The analysis of the coverage intervals generated by the APU method can be observed in Table 3. The 95% coverage interval around the predicted value practically covered 100% of the test data, indicating that the constructed interval is indeed consistent with its intended purpose.

In the validation of data uncertainty through the [33], the values of y_{inf}, y_{sup}, d_{inf}, and d_{sup} mentioned in the previous section were obtained, making it possible to assess compliance with the established tolerance using the 31 vectors

from the test group. The details of these assessments are presented in Tables 4. Entries highlighted in red in Table 4 indicate instances that exceeded the established tolerance limit, which was based on 2 significant digits. If we reconsider the tolerance (δ) and agree to express the uncertainty with only 1 significant digit, the results are validated at 100% . This demonstrates that the analytical method is perfectly aligned with the simulations performed by the JCGM 101 method for each test vector.

5.4. MC Dropout

After constructing the neural network according to the characteristics previously described, T=1000 forward passes through the network are performed, thus collecting T predictions y for each sample. Therefore, for each vector belonging to the test set, we will have a prediction for the mean and a prediction for the variance, which will be our measure of uncertainty. The epistemic uncertainty was then determined

Table 3

Validation of the coverage interval.u_D represents the data uncertainty that varies according to the input (v). This is directly related to the magnitude of the partial derivatives calculated by the APU method for propagation. u_M represents the model uncertainty, based on the distribution of errors generated by the model. This uncertainty is fixed for any input value v. u_P represents the predictive uncertainty, which is the combination of u_D and u_M. Meanwhile, U is the predictive uncertainty expanded by the factor k_p to achieve a coverage level of 95%.I represents the 95% coverage interval constructed from U.

v	$Real(Y_i)$	Predict (y_i)	u_D	u_M	u_P	$U_{95\%}$	$I = y_i \pm U_{95\%}$	$Y_i \in I$?
0	0.597	0.570	0.0034	0.0670	0.0671	0.1374	[0.43 , 0.71]	yes
1	0.648	0.683	0.0049	0.0670	0.0672	0.1376	[0.55 , 0.82]	yes
2	0.672	0.685	0.0064	0.0670	0.0673	0.1378	[0.55 , 0.82]	yes
3	0.383	0.419	0.0160	0.0670	0.0689	0.1410	[0.28 , 0.56]	yes
4	0.591	0.642	0.0180	0.0670	0.0694	0.1420	[0.50 , 0.78]	yes
5	0.638	0.682	0.0256	0.0670	0.0717	0.1466	[0.54 , 0.83]	yes
6	0.421	0.439	0.0236	0.0670	0.0710	0.1453	[0.29 , 0.58]	yes
7	0.579	0.592	0.0181	0.0670	0.0694	0.1420	[0.45 , 0.73]	yes
8	0.609	0.662	0.0154	0.0670	0.0687	0.1407	[0.52 , 0.80]	yes
9	0.590	0.686	0.0020	0.0670	0.0670	0.1373	[0.55 , 0.82]	yes
10	0.660	0.594	0.0065	0.0670	0.0673	0.1378	[0.46 , 0.73]	yes
11	0.545	0.553	0.0048	0.0670	0.0672	0.1376	[0.41 , 0.69]	yes
12	0.625	0.553	0.0061	0.0670	0.0673	0.1378	[0.41 , 0.69]	yes
13	0.820	0.752	0.0066	0.0670	0.0673	0.1379	[0.61 , 0.89]	yes
14	0.720	0.660	0.0062	0.0670	0.0673	0.1378	[0.52 , 0.80]	yes
15	0.820	0.757	0.0021	0.0670	0.0670	0.1373	[0.62 , 0.89]	yes
16	0.473	0.537	0.0048	0.0670	0.0672	0.1376	[0.40 , 0.67]	yes
17	0.421	0.379	0.0045	0.0670	0.0671	0.1375	[0.24 , 0.52]	yes
18	0.448	0.495	0.0182	0.0670	0.0694	0.1421	[0.35 , 0.64]	yes
19	0.332	0.432	0.0050	0.0670	0.0672	0.1376	[0.29 , 0.57]	yes
20	0.385	0.375	0.0133	0.0670	0.0683	0.1398	[0.23 , 0.51]	yes
21	0.414	0.429	0.0070	0.0670	0.0674	0.1379	[0.29 , 0.57]	yes
22	0.451	0.450	0.0154	0.0670	0.0687	0.1407	[0.31 , 0.59]	yes
23	0.364	0.390	0.0091	0.0670	0.0676	0.1384	[0.25 , 0.53]	yes
24	0.340	0.455	0.0260	0.0670	0.0719	0.1469	[0.31 , 0.60]	yes
25	0.756	0.635	0.0181	0.0670	0.0694	0.1420	[0.49 , 0.78]	yes
26	0.766	0.684	0.0034	0.0670	0.0671	0.1374	[0.55 , 0.82]	yes
27	0.409	0.399	0.0120	0.0670	0.0681	0.1393	[0.26 , 0.54]	yes
28	0.643	0.616	0.0181	0.0670	0.0694	0.1420	[0.47 , 0.76]	yes
29	0.604	0.606	0.0105	0.0670	0.0678	0.1389	[0.47 , 0.75]	yes
30	0.447	0.491	0.0340	0.0670	0.0751	0.1535	[0.34 , 0.64]	yes

by the standard deviation of the T=1000 forward-pass for each test vector, and the results can be seen in Figure 8.

5.4.1. Discussion

As mentioned in the previous sections, the APU method addresses the determination of random and epistemic uncertainties separately. These uncertainties are combined to determine the predictive uncertainty. In contrast, the MC dropout method only addresses the epistemic uncertainty.

In Figure 7, we can observe a comparison between the epistemic uncertainties obtained by APU and MC dropout methods. It is essential to note that these estimates of epistemic uncertainty were obtained through different approaches. The APU method was based on the prediction errors of the trained model. In a conservative way, it projects a range of uncertainty considering how much the network can err, based on the knowledge obtained during training and testing.

Still, the results obtained were smaller than those obtained by MC dropout, whose method for obtaining uncertainty is Bayesian. It cannot be stated that the results from the APU were underestimated when compared to MC dropout because the predictive uncertainty, mainly formed by the epistemic uncertainty, showed an excellent fit to the 95% of the test data.

5.5. Final Discussion

Regarding uncertainties associated with a single decision, it can be stated that, especially in critical mission and safety applications, the precise assessment of measures can be of paramount importance. Therefore, evaluation approaches of this kind are highly desirable. In this need for precise uncertainty assessment, the APU method becomes useful in the sense that this technique approximates the black-box function at a specific point in the domain. Since this technique requires minimal computational resources and can be analytically and nearly instantaneously estimated, it can be leveraged when instant estimation of the uncertainty

Table 4

Validation of the test group points regarding the adherence of the proposed method. Tolerance = 0.005 for uncertainty whith 2 significant digits

vi	y	Up	y_{inf}	y_{sup}	Tolerance:0.005		Tolerance:0.05	
					d_{inf}	d_{sup}	d_{inf}	d_{sup}
0	0.586	0.008	0.581	0.59	0.004	0.004	0.004	0.004
1	0.68	0.024	0.653	0.69	0.003	0.014	0.003	0.014
2	0.686	0.013	0.684	0.698	0.011	0.001	0.011	0.001
3	0.425	0.034	0.378	0.464	0.014	0.006	0.014	0.006
4	0.63	0.045	0.569	0.692	0.015	0.017	0.015	0.017
5	0.685	0.06	0.653	0.74	0.028	0.005	0.028	0.005
6	0.439	0.058	0.375	0.508	0.006	0.011	0.006	0.011
7	0.603	0.039	0.569	0.637	0.005	0.005	0.005	0.005
8	0.655	0.013	0.642	0.665	0.001	0.004	0.001	0.004
9	0.683	0.025	0.666	0.689	0.007	0.019	0.007	0.019
10	0.603	0.022	0.586	0.622	0.005	0.002	0.005	0.002
11	0.563	0.015	0.55	0.577	0.002	0.001	0.002	0.001
12	0.558	0.018	0.543	0.573	0.003	0.003	0.003	0.003
13	0.775	0.01	0.739	0.778	0.026	0.007	0.026	0.007
14	0.694	0.021	0.672	0.7	0.001	0.015	0.001	0.015
15	0.774	0.03	0.748	0.781	0.004	0.023	0.004	0.023
16	0.549	0.015	0.543	0.563	0.008	0.002	0.008	0.002
17	0.384	0.012	0.37	0.396	0.003	0	0.003	0
18	0.495	0.04	0.444	0.526	0.012	0.009	0.012	0.009
19	0.429	0.013	0.419	0.44	0.003	0.002	0.003	0.002
20	0.354	0.029	0.328	0.378	0.003	0.004	0.003	0.004
21	0.405	0.019	0.388	0.422	0.002	0.002	0.002	0.002
22	0.456	0.055	0.422	0.491	0.021	0.019	0.021	0.019
23	0.387	0.027	0.365	0.409	0.006	0.005	0.006	0.005
24	0.442	0.056	0.417	0.489	0.031	0.009	0.031	0.009
25	0.638	0.039	0.601	0.662	0.002	0.015	0.002	0.015
26	0.673	0.007	0.667	0.675	0	0.005	0	0.005
27	0.368	0.024	0.345	0.394	0.001	0.001	0.001	0.001
28	0.631	0.039	0.593	0.665	0.001	0.005	0.001	0.005
29	0.61	0.012	0.601	0.622	0.003	0.001	0.003	0.001
30	0.486	0.056	0.44	0.524	0.011	0.018	0.011	0.018

associated with a single decision is essential, such as in autonomous driving.

Regarding the lack of knowledge about genuine sources of uncertainty in a specific domain, this is related to the fact that any type of neural network training is based on data sampling within a universal set that can be highly heterogeneous. Studies can be directed towards possible mitigations of this problem through data sampling techniques that might be more representative. Methodologies for developing a training dataset through a systematic and problem-based procedure can assist in reducing the effect of this uncertainty ignorance. Another significant aspect that can be addressed is in-depth research into a specific domain, thereby providing better insights into the choice of training data for supervised neural networks. In this work, attempts were made to estimate real uncertainties within the chosen domain, which is fundamentally related to the quality of measurements collected by measurement systems. Thus, evaluating the calibration certificates of instruments and understanding their behavior throughout their usage range provides valuable insights into the accuracy of uncertainties within the domain.

On the other hand, [17] states that there are several limitations to uncertainty combination approaches. According to the author, BNN or ensemble models require multiple forward passes for prediction, introducing extra computational and storage overheads. Efficiency is a concern. Secondly, the mere combination of data and model uncertainty lacks theoretical assurance, necessitating post-hoc calibration in the model. The APU method is an important alternative as it addresses uncertainties independently, doesn't introduce any computational overhead, and is well-founded within the physical sciences.

It's important to emphasize some key characteristics of the APU system, such as being agnostic to the network's architecture and task, not requiring changes in the optimization process, and the possibility of application to pre-formed architectures. These are desirable characteristics according to [31] for any structure aiming to estimate measurement uncertainty in deep learning. The APU method meets these considerations and becomes, in general, an excellent alternative for estimating uncertainty in deep learning.

Finally, with regard to technology transfer, some points are crucial. Firstly, the method does not interfere with the
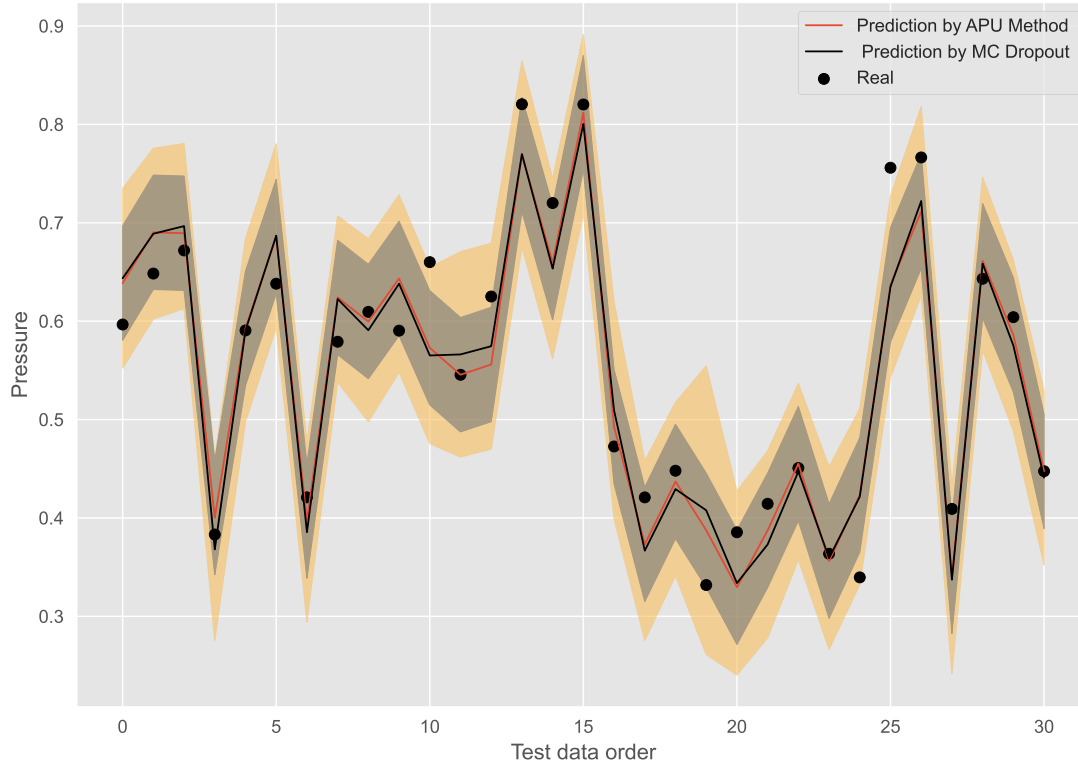
Figure 7: The area delimited in yellow represents the model uncertainty obtained by the **MC Dropout** method. The area delimited in brown represents the model uncertainty obtained by the **APU** Method.

training process or the architecture of the neural network. The method comes into play after these stages. For its implementation, it is necessary that relevant information about the dataset be available or rely on the specific knowledge of the domain in question. One such piece of information concerns the data acquisition, including the measurement instruments used, and if possible, the corresponding calibration certificates.

It is noteworthy that the metrological reliability of the dataset is crucial. In other words, it is essential to consider any bias caused by the instruments when estimating the uncertainty of the data. In the worst case, an estimate of this uncertainty can be obtained by evaluating the type of instrument used. It is crucial that this uncertainty represents the entire range of possible values for the variable.

6. Conclusion

The APU method emerges as a highly viable option for estimating uncertainties in deep learning applications. It allows the assessment of the impact of uncertainty associated with both data and the model, proving particularly effective in evaluating the reliability of predictions. Through APU, it

becomes possible to determine a comprehensive probabilistic range, providing an assessment of the risks associated with considering an outcome predicted by a neural network.

Unlike other proposed approaches, APU is capable of estimating all these uncertainties with very low computational cost, as it is based on a mathematical model of uncertainty propagation. Its computational efficiency enables the prompt determination of uncertainty, making it an attractive alternative, especially in edge computing, where computational efficiency is crucial. Additionally, being grounded in JCGM 100, the APU method benefits from the availability of a validation method through JCGM 101, which is a supplement to JCGM 100.

There are some significant obstacles in this approach. Firstly, it is important to note that this approach is applied restrictively to the field of physical sciences, in regression models, limiting its applicability in other fields. Additionally, there may be a possible lack of knowledge about how the data was obtained, which can affect the reliability of the results. Another obstacle is the potential lack of traceability of the measuring instruments used in creating the dataset, which may introduce difficulties in estimating instrument-related uncertainties. Finally, since the method relies on the

neural network's performance, any change that affects the network's performance, such as a shift in the data distribution, will also impact the accuracy of the APU method until the network undergoes retraining.

As next steps, it would be interesting to explore expanding the application of this method to other types of prediction algorithms beyond MLP neural networks, such as polynomial and logistic regression algorithms, decision trees, among others.

References

[1] T. Hey, S. Tansley, K. Tolle, J. Gray, The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research, 2009. URL https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery/

[2] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, Physics reports 810 (2019) 1–124.

[3] O. A. Helene, V. R. Vanin, Tratamento estatístico de dados em física experimental, Editora Blucher, 1991.

[4] Weytjenset al., Learning uncertainty with artificial neural networks for predictive process monitoring, Applied Soft Computing 125 (2022) 109134.

[5] L. Zhong, X. Guo, Z. Xu, M. Ding, Soil properties: Their prediction and feature extraction from the lucas spectral library using deep convolutional neural networks, Geoderma 402 (2021) 115366.

[6] H. Lambers, W. C. Plaxton, Phosphorus: back to the roots, Annual plant reviews volume 48: phosphorus metabolism in plants 48 (2015) 1–22.

[7] G. W. Leeper, N. C. Uren, et al., Soil science: an introduction., Melbourne University Press, 1993.

[8] N. Barrow, The effects of ph on phosphate uptake from the soil, Plant and soil 410 (2017) 401–410.

[9] R. Ghanem, D. Higdon, H. Owhadi, Handbook of uncertainty quantification, Springer, 2017.

[10] Zhou et al, A survey on epistemic (model) uncertainty in supervised learning: Recent advances and applications, Neurocomputing 489 (2022) 449–465.

[11] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al., A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Information Fusion 76 (2021) 243–297.

[12] Schmähling et al., A framework for benchmarking uncertainty in deep regression, Applied Intelligence (2022) 1–14.

[13] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: International conference on machine learning, PMLR, 2017, pp. 1321–1330.

[14] M. JOSÉ, O. PUJOL, J. VITRIÀ, A survey on uncertainty estimation in deep learning classification systems from a bayesian perspective (2022).

[15] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al., A survey of uncertainty in deep neural networks, arXiv preprint arXiv:2107.03342 (2021).

[16] H. D. Kabir, A. Khosravi, M. A. Hosen, S. Nahavandi, Neural network-based uncertainty quantification: A survey of methodologies and applications, IEEE access 6 (2018) 36218–36234.

[17] W. HE, A survey on uncertainty quantification methods for deepneural networks: An uncertainty source's perspective 2022-March (2022).

[18] T. Blasco, J. S. Sánchez, V. García, A survey on uncertainty quantification in deep learning for financial time series prediction, Neurocomputing 576 (2024) 127339.

[19] K. Zou, Z. Chen, X. Yuan, X. Shen, M. Wang, H. Fu, A review of uncertainty estimation and its application in medical imaging, Meta-Radiology (2023) 100003.

[20] Z. Guo, Z. Wan, Q. Zhang, X. Zhao, Q. Zhang, L. M. Kaplan, A. Jøsang, D. H. Jeong, F. Chen, J.-H. Cho, A survey on uncertainty reasoning and quantification in belief theory and its application to deep learning, Information Fusion (2023) 101987.

[21] B. van Stein, H. Wang, W. Kowalczyk, T. Bäck, A novel uncertainty quantification method for efficient global optimization, in: Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications: 17th International Conference, IPMU 2018, Cádiz, Spain, June 11-15, 2018, Proceedings, Part III 17, Springer, 2018, pp. 480–491.

[22] JCGM-100, Evaluation of measurement data—guide to the expression of uncertainty in measurement, Int. Organ. Stand. Geneva ISBN 50 (2008) 134.

[23] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural computation 4 (1) (1992) 1–58.

[24] U. Grenander, On empirical spectral analysis of stochastic processes, Arkiv för Matematik 1 (1951) 503–531.

[25] Z. Yang, Y. Yu, C. You, J. Steinhardt, Y. Ma, Rethinking bias-variance trade-off for generalization of neural networks, in: International Conference on Machine Learning, PMLR, 2020, pp. 10767–10777.

[26] J. Heiss, J. Weissteiner, H. Wutte, S. Seuken, J. Teichmann, Nomu: Neural optimization-based model uncertainty, arXiv preprint arXiv:2102.13640 (2021).

[27] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, PMLR, 2016, pp. 1050–1059.

[28] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, Advances in neural information processing systems 30 (2017).

[29] F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, S. Nowozin, How good is the bayes posterior in deep neural networks really?, arXiv preprint arXiv:2002.02405 (2020).

[30] F. K. Gustafsson, M. Danelljan, T. B. Schon, Evaluating scalable bayesian deep learning methods for robust computer vision, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 318–319.

[31] Loquercio et al., A general framework for uncertainty estimation in deep learning, IEEE Robotics and Automation Letters 5 (2) (2020) 3153–3160.

[32] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, Advances in neural information processing systems 30 (2017).

[33] JCGM-101, Evaluation of measurement data–supplement 1 to the "guide to the expression of uncertainty in measurement"–propagation of distributions using a monte carlo method, (Geneva: Organisation for Standardization) (2008).

[34] X. Cheng, B. Khomtchouk, N. Matloff, P. Mohanty, Polynomial regression as an alternative to neural nets (2018). arXiv:1806.06850. URL https://arxiv.org/abs/1806.06850

[35] L. Nyland, M. Snyder, Fast trigonometric functions using intel's sse2 instructions, Tech. rep., Intel Tech. Rep., available online at: http://www.weblearn.hs-bremen (2004).

[36] F. Temurtas, et al., A study on neural networks using taylor series expansion of sigmoid activation function, in: A. Laganà, et al. (Eds.), ICCSA (4), Vol. 3046 of Lecture Notes in Computer Science, Springer, 2004, pp. 389–397.

[37] Y. Katznelson, W. Rudin, The stone-weierstrass property in banach algebras, Pacific Journal of Mathematics 11 (1) (1961) 253–265.

[38] X. Tomàs, J. Cuadros, L. González, Introducción al cálculo numérico, IQS, 2006.

[39] T. F. Brooks, D. S. Pope, M. A. Marcolini, Airfoil self-noise and prediction, Tech. rep. (1989).

A. Proof by Induction of the Analyticity of Function Composition

To show that the composition of n analytic functions is also an analytic function, we will use mathematical induction.

Base of induction: First, we demonstrate that the composition of two analytic functions is analytic. Let $f(z)$ be analytic in a domain D and $g(z)$ analytic in a domain E, such that $f(D) \subset E$. By definition, $f(z)$ being analytic implies that it can be expressed as a convergent Taylor series around z_0 in D:

$$f(z) = \sum_{n=0}^{\infty} a_n(z - z_0)^n$$

Similarly, $g(z)$ being analytic implies that it can be expressed as a convergent Taylor series around w_0 in E:

$$g(z) = \sum_{n=0}^{\infty} b_n(z - w_0)^n$$

Now, consider the composition $h(z) = g(f(z))$. Substituting the Taylor series of $f(z)$ into the expression for $h(z)$:

$$h(z) = g(f(z)) = g\left(\sum_{n=0}^{\infty} a_n(z - z_0)^n\right)$$

Since $g(z)$ is analytic in E, we can apply the Taylor series of $g(z)$ around $f(z_0)$, as $f(z_0)$ lies in E:

$$g\left(\sum_{n=0}^{\infty} a_n(z - z_0)^n\right) = \sum_{n=0}^{\infty} b_n\left(\sum_{m=0}^{\infty} a_m(z - z_0)^m - w_0\right)^n$$

Thus, $h(z)$ can be expressed as a convergent power series around z_0 in D, implying that $h(z)$ is analytic in D. Therefore, the composition of two analytic functions is analytic.

Induction hypothesis: Suppose the composition of n analytic functions is analytic.

Inductive step: Consider the composition of $n + 1$ analytic functions. If $h_n(z)$ is analytic (as shown by the induction hypothesis) and $f_{n+1}(z)$ is analytic in a domain containing $h_n(z)$, then the composition $h_{n+1}(z) = f_{n+1}(h_n(z))$ is also analytic.

Therefore, by mathematical induction, we conclude that the composition of n analytic functions is an analytic function for any n.

Figure 8: In item "U" of the figure, we can observe the uncertainty of measurement in calibration. Measurements cannot be absolute, and even with the most expensive equipment and controlled environments, there is always a degree of variation. The uncertainty value printed on the calibration certificate will take into account various factors such as repeatability of results, linearity, atmosphere, equipment used, etc., to provide a value that encompasses all these variabilities.(Certificate of calibration model-https://www.klipspringer.com/blogs/understanding-your-ukas-calibration-certificate/.)

Table 5
Estimated uncertainties for all vectors in the test group

Test vector	Random uncertainty before neural network propagation						Random uncertainty after neural network propagation							Uncertainty		
	up(x1) Hz	up(x2) °	up(x3) m	up(x4) m/s	up(x5) m	u(y) dB	u(x1) dB	u(x2) dB	u(x3) dB	u(x4) dB	u(x5) dB	u(Correl.) dB	u(y) dB	Epistemic dB	Predict dB	U(95%) dB
0	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0004	-0.0004	0.0004	0.0000	0.0000	0.0000	0.0020	0.0656	0.0656	0.136
1	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0000	0.0001	-0.0001	0.0003	-0.0104	0.0000	0.0020	0.0656	0.0664	0.137
2	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0000	0.0003	-0.0001	0.0000	0.0045	0.0000	0.0020	0.0656	0.0658	0.136
3	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0003	0.0000	0.0004	-0.0179	0.0000	0.0020	0.0656	0.0681	0.141
4	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0003	-0.0002	0.0006	-0.0328	0.0000	0.0020	0.0656	0.0735	0.151
5	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0003	-0.0004	-0.0001	0.0006	-0.0343	0.0000	0.0020	0.0656	0.0741	0.153
6	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0006	-0.0001	0.0006	-0.0253	0.0000	0.0020	0.0656	0.0705	0.146
7	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0004	0.0002	0.0000	0.0000	0.0209	0.0000	0.0020	0.0656	0.0689	0.142
8	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0004	-0.0004	-0.0001	0.0006	-0.0089	0.0000	0.0020	0.0656	0.0663	0.137
9	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0003	-0.0001	-0.0002	0.0006	-0.0149	0.0000	0.0020	0.0656	0.0673	0.139
10	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0002	-0.0005	0.0000	0.0009	-0.0134	0.0000	0.0020	0.0656	0.0671	0.139
11	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0003	-0.0001	0.0001	0.0003	0.0089	0.0000	0.0020	0.0656	0.0662	0.137
12	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0003	-0.0002	0.0001	0.0000	0.0119	0.0000	0.0020	0.0656	0.0667	0.138
13	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	0.0001	-0.0002	0.0003	-0.0075	0.0000	0.0020	0.0656	0.0660	0.137
14	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0003	-0.0001	-0.0002	0.0009	-0.0119	0.0000	0.0020	0.0656	0.0667	0.138
15	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	0.0001	-0.0001	0.0006	-0.0060	0.0000	0.0020	0.0656	0.0659	0.136
16	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0002	-0.0003	0.0001	0.0003	0.0060	0.0000	0.0020	0.0656	0.0659	0.136
17	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0005	-0.0001	0.0004	-0.0022	0.0000	0.0020	0.0656	0.0657	0.136
18	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0004	-0.0001	0.0006	-0.0194	0.0000	0.0020	0.0656	0.0685	0.142
19	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0004	0.0002	0.0003	-0.0001	0.0022	0.0000	0.0020	0.0656	0.0657	0.136
20	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0002	0.0001	-0.0002	0.0009	-0.0104	0.0000	0.0020	0.0656	0.0664	0.137
21	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0002	0.0003	0.0000	0.0003	-0.0104	0.0000	0.0020	0.0656	0.0664	0.137
22	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0007	-0.0001	0.0006	-0.0261	0.0000	0.0020	0.0656	0.0708	0.146
23	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	0.0000	-0.0002	0.0010	-0.0104	0.0000	0.0020	0.0656	0.0665	0.138
24	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0008	-0.0001	0.0007	-0.0238	0.0000	0.0020	0.0656	0.0700	0.145
25	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0002	-0.0001	-0.0003	0.0012	-0.0179	0.0000	0.0020	0.0656	0.0681	0.141
26	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0001	0.0006	0.0004	0.0000	-0.0060	0.0000	0.0020	0.0656	0.0659	0.136
27	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0002	0.0000	-0.0002	0.0009	-0.0097	0.0000	0.0020	0.0656	0.0663	0.137
28	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	0.0004	0.0003	0.0000	0.0003	0.0194	0.0000	0.0020	0.0656	0.0685	0.142
29	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0001	-0.0002	0.0003	-0.0089	0.0000	0.0020	0.0656	0.0662	0.137
30	0.0001	0.0012	0.0004	0.0049	0.0250	0.0020	-0.0001	-0.0008	-0.0001	0.0007	-0.0291	0.0000	0.0020	0.0656	0.0720	0.149