Investigation of performance metrics in regression analysis and machine learning-based prediction models

Vagelis Plevris Plevris[1], German Solorzano Solorzano[2], Nikolaos Bakas Bakas[3], Mohamed El Amine Ben Seghier Seghier[4]

[1] Qatar University
[2] Oslo Metropolitan University
[3] RDC Informatics
[4] Ton Duc Thang University

INFORMATION

Keywords:
Error metric
Performance metric
Evaluation
Regression model
Machine learning
Neural network
Prediction model.

DOI: 10.23967/eccomas.2022.155

Published: 25/11/2022

The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5 – 9 June 2022, Oslo, Norway

INVESTIGATION OF PERFORMANCE METRICS IN REGRESSION ANALYSIS AND MACHINE LEARNING-BASED PREDICTION MODELS

VAGELIS PLEVRIS[1], GERMAN SOLORZANO[2],
NIKOLAOS P. BAKAS[3] AND MOHAMED EL AMINE BEN SEGHIER[4]

[1] Department of Civil and Architectural Engineering, Qatar University
P.O. Box: 2713, Doha, Qatar
e-mail: vplevris@qu.edu.qa

[2] Department of Civil Engineering and Energy Technology, OsloMet–Oslo Metropolitan University
Pilestredet 35, Oslo 0166, Norway
e-mail: germanso@oslomet.no

[3] Research and Development Department, RDC Informatics
Athens, Greece
e-mail: n.bakas@rdc.gr

[4] Faculty of Civil Engineering, Ton Duc Thang University
Ho Chi Minh City, Vietnam
e-mail: benseghier@tdtu.edu.vn

Key words: Error metric, Performance metric, Evaluation, Regression model, Machine learning, Neural network, Prediction model.

Abstract. Performance metrics (Evaluation metrics or error metrics) are crucial components of regression analysis and machine learning-based prediction models. A performance metric can be defined as a logical and mathematical construct designed to measure how close the predicted outcome is to the actual result. A variety of performance metrics have been described and proposed in the literature. Knowledge about the metrics' properties needs to be systematized to simplify their design and use. In this work, we examine various regression related metrics (14 in total) for continuous variables, including the most widely used ones, such as the (root) mean squared error, the mean absolute error, the Pearson correlation coefficient, and the coefficient of determination, among many others. We provide their mathematical formulations, as well as a discussion on their use, their characteristics, advantages, disadvantages, and limitations, through theoretical analysis and a detailed numerical example. The 10 unitless metrics are further investigated through a numerical analysis with Monte Carlo Simulation based on (i) random guessing and (ii) the addition of random noise with various noise ratios to the predicted values. Some of the metrics show a poor or inconsistent performance, while others exhibit good performance as evaluation measures of the "goodness of fit". We highlight the importance of the usage of the right metrics to obtain good predictions in machine learning and regression models in general.

1 INTRODUCTION

Regression analysis [1] is a statistical predictive modelling technique, which investigates the relationship between a dependent (target) and independent variable(s) (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between variables. Regression analysis is an important tool for modelling and analyzing data. There are many types or regression analysis. In Linear Regression, the nature of regression line is linear, and the analysis yields a predicted value for the target resulting from a linear combination of the predictors. Other types or regression is Logistic regression, where the dependent variable is binary, Polynomial Regression, where the power of independent variable is more than one, Stepwise Regression, Ridge Regression, Lasso Regression, ElasticNet Regression and others.

Machine learning (ML) is a type of artificial intelligence (AI) technique that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. ML algorithms use historical data as inputs to predict new output values. A ML model aims at making sure that every time a sample is presented to it, the predicted outcome corresponds to the true outcome. Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: (i) supervised learning, (ii) unsupervised learning, (iii) semi-supervised learning, and (iv) reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict. Supervised ML requires the data scientist to train the algorithm with both labeled inputs and desired outputs. Supervised learning algorithms are good for the following tasks: (i) Binary classification, (ii) Multi-class classification, (iii) Regression modeling, and (iv) Ensembling.

In the present study we focus on Regression modeling, i.e. predicting values of continuous variables. Predictive analytics combines techniques like predictive modeling with machine learning to analyze past data to predict future trends. ML methods for regression include Decision Tree Regression, Random Forests [2], Support Vector regression Machines [3], Neural Network Regression, and others. An artificial neural network (ANN) is a ML predictive model designed to work the way a human brain does. ANNs may be used for solving problems the human brain is very good at, such as recognizing sounds, pictures, or text, among others. Neural networks have a multilayer structure: neurons in one layer transmit data to several neurons on the next one, and so on. Linear regression models use only input and output nodes to make predictions. The ANN also uses the hidden layer to make predictions more accurate.

ANN models have been applied in many problems in economics, engineering and other scientific fields. Particularly in structural engineering [4, 5], they have been successfully used for modeling masonry failure [6-12], predicting the properties of FRP-Confined Concrete Cylinders [13], predicting the compressive strength of concrete containing recycled aggregate [14, 15], modeling the corrosion rate in cables of suspension bridges [16], predicting the compressive strength of CRM samples [17], predicting the bond stress of corroded steel reinforcing bars in concrete members [18], designing reinforced concrete footings [19], predicting the periods of buildings [20], predicting the capacity of concrete walls [21], determining the nominal shear capacity of steel fiber reinforced concrete beams [22], optimizing large-scale 3d trusses [23], among other interesting and innovative applications.

In ML regression models, performance metrics are used to compare the trained model predictions with the actual (observed) data from the testing data set [24]. Forecasting has a long history of using such performance metrics to measure the deviation of forecasts from the observations and assess the quality of the forecasting method used [25]. In general, the higher the difference between the real outcome 'r' and the predicted outcome 'p', the more "off" the model is from being an accurate representation of the phenomenon. On the other hand, the closer the values, the better the performance of the system. Performance metrics for regression (regression-related metrics, or regression error metrics) usually involve calculating an error score to summarize the predictive skill of a model. The most widely used performance metrics for evaluating and reporting the performance of a regression model are the (root) mean squared error, the mean absolute error, the Pearson correlation coefficient, and the coefficient of determination. Other than these well-known metrics, many other exist and are being used in several application areas.

It should be highlighted that performance metrics are different from loss functions. Loss functions show a measure of model performance, they are used to train a machine learning model (using some kind of optimization like Gradient Descent), and they are usually differentiable in the model's parameters. Model performance metrics on the other hand, are used to monitor and measure the performance of a model usually after training, and don't need to be differentiable. They help us evaluate the model's accuracy and measure the performance of a trained model. By using different metrics for performance evaluation, one can improve the overall predictive power of the model. In terms of machine learning performance, it is key to define that when we talk about errors, we specifically refer to the difference between the actual target value and the predicted value.

Botchkarev [26] analyzed various performance metrics and approaches to their classification and developed a new typology in an effort to advance knowledge of metrics and facilitate their use in machine learning regression algorithms. He proposed the classification of metrics in four main categories: primary metrics, extended metrics, composite metrics, and hybrid sets of metrics. The paper identified three key components that determine the structure and properties of primary metrics: method of determining point distance, method of normalization, and method of aggregation of point distances over a data set. In another work [27], Botchkarev evaluated the performance of regression machine learning models using multiple error metrics in Azure Machine Learning Studio.

The objective of this paper is to review a variety of existing performance metrics and test them using numerical examples, in order to help improve our knowledge and understanding of the metrics and facilitate their use in machine learning regression, forecasting and prognostics. The rest of this paper is organized as follows: In section 2, the prediction error metrics and some relevant statistics are introduced. In section 3 a simple numerical example is presented, and the relevant metrics and statistics are calculated. Section 4 includes a numerical investigation of the various error metrics, with different scenarios simulated with Monte Carlo simulation, followed by section 5 where the conclusions are discussed.

2 PREDICTION ERROR METRICS AND RELEVANT STATISTICS

2.1 Statistics and error metrics

Predictions obtained by a regression model, i.e. a Neural Network or a similar prediction model are usually not completely accurate and need not be completely accurate, to avoid the problem of overfitting [28]. Normally, each prediction contains an error which has to be quantified in order for one to be able to compare the results obtained from different models and assess the performance of each model. Various metrics measuring the prediction error can be used for this purpose. Let $\boldsymbol{p}$ (N×1 vector) be the predicted values and $\boldsymbol{r}$ (N×1 vector) be the real values of a quantity calculated (or measured) and then predicted a number N of times. For example, we ask a ANN to predict some values, N times. N can be the number of input and output data in the whole data set, or in a subset of it such as the training or testing subset, or we can even use completely new data to make an unbiased independent assessment.

In the next paragraphs we define and discuss several metrics that can be used for the calculation of the prediction error of such a model. In this paper, we study the case of continuous variables. In the case of categorical ones, other metrics are used, such as the confusion matrix, accuracy, recall, precision, false positive rate, etc. Note that the formulas given below work for observations which have all positive values, i.e. there are no negative values or zero values in the observations or their predictions. In case of negative values or zero values being part of the data, some formulas may need to be properly adjusted.

The Bias Error $\boldsymbol{e} = e_{i\in[i]}$, with $[i]=\{1, 2, \ldots, N\}$, can be expressed as the difference between the predicted values and the real (target) values, it can take positive, negative or zero values for each observation, and is stated as

$$e_i = p_i - r_i \tag{1}$$

The **Mean Bias** (*MB*) is the average of the bias errors. It can also take both positive, negative or zero values and it is given by

$$MB = \bar{e} = \frac{1}{N}\sum_{i=1}^{N} e_i = \frac{1}{N}\sum_{i=1}^{N}\left(p_i - r_i\right) = \bar{p} - \bar{r} \tag{2}$$

Where $\bar{p}$ and $\bar{r}$ are the mean values of $\boldsymbol{p}$ and $\boldsymbol{r}$, respectively:

$$\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{3}$$

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i \tag{4}$$

Although for a perfect match between real and predicted values (i.e. having identical values), *MB* equals zero (a necessary condition), the condition *MB*=0 is not also a sufficient condition, as positive and negative errors can cancel each other out causing *MB*=0 in cases where the match is far from perfect.

The **Mean Absolute Gross Error** (*MAGE*) measures the average magnitude of the errors in a set of predictions, without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. It takes positive or zero values and it is given by

$$MAGE = \frac{1}{N}\sum_{i=1}^{N}\left|e_i\right| = \frac{1}{N}\sum_{i=1}^{N}\left|p_i - r_i\right| \tag{5}$$

The Mean Squared Error (*MSE*) is a popular regression-related metric having to do with the average squared error between the predicted and actual values. It takes positive or zero values and is given by

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(p_i - r_i\right)^2 \tag{6}$$

One major disadvantage of *MSE* is that it is not robust to outliers. In case a sample has an associated error way larger than the one of other samples, the square of the error will be even larger. This, paired to the fact that *MSE* calculates the average of errors, makes *MSE* prone to outliers.

The **Root Mean Squared Error** (*RMSE*) is also a frequently used measure of the differences between values (sample or population values) predicted by a model, or an estimator and the values observed. It is the square root of *MSE*. Unlike *MSE*, *RMSE* provides an error measure in the same unit as the target variable. It takes values in the range [0, +∞) and it is given by

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(p_i - r_i\right)^2} \tag{7}$$

The Centered Mean Square Difference (*CMSD*) is given by

$$CMSD = \frac{1}{N}\sum_{i=1}^{N}\left[\left(p_i - \bar{p}\right) - \left(r_i - \bar{r}\right)\right]^2 \tag{8}$$

The **Centered Root Mean Square Difference** (*CRMSD*) is the square root of *CMSD*, expressed in the same unit as the target variable, and given by

$$CRMSD = \sqrt{CMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\left(p_i - \bar{p}\right) - \left(r_i - \bar{r}\right)\right]^2} \tag{9}$$

The *CRMSD* metric is used in a Taylor diagram [29] to express the prediction error of a model, as will be discussed in detail in section 2.3.

The **Mean Normalized Bias** (*MNB*, unitless), usually expressed as a percentage, is the average value of the normalized bias error values, given by

$$MNB = \frac{1}{N}\sum_{i=1}^{N}\frac{p_i - r_i}{r_i} = \frac{1}{N}\left(\sum_{i=1}^{N}\frac{p_i}{r_i}\right) - 1 \tag{10}$$

The **Mean Normalized Gross Error** (*MNGE*, unitless) is usually expressed as a percentage, and it is also known as "Mean Absolute Percentage Error". By expressing the error as a percentage, one can have a better understanding of how off the predictions are in relative terms. It is given by

$$MNGE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|p_i - r_i\right|}{r_i} \tag{11}$$

However, *MNGE* is highly sensitive in cases where some real values r_i are very close to zero. In these cases, high values of *MNGE* can occur even if the corresponding prediction p_i is not very different to the real value r_i.

The **Normalized Mean Bias** (*NMB*, unitless), usually expressed as a percentage, is given by

$$NMB = \frac{\sum_{i=1}^{N}\left(p_i - r_i\right)}{\sum_{i=1}^{N} r_i} = \frac{\overline{p}}{\overline{r}} - 1 \tag{12}$$

The **Normalized Mean Error** (*NME*, unitless) is given by

$$NME = \frac{\sum_{i=1}^{N}\left|p_i - r_i\right|}{\sum_{i=1}^{N} r_i} = \frac{MAGE}{\overline{r}} \tag{13}$$

The **Fractional Bias** (*FB*, unitless) is given by

$$FB = \frac{2}{N}\sum_{i=1}^{N}\frac{p_i - r_i}{p_i + r_i} = \frac{2}{N}\sum_{i=1}^{N}\frac{e_i}{p_i + r_i} \tag{14}$$

The **Fractional Gross Error** (*FGE*, unitless) is given by

$$FGE = \frac{2}{N}\sum_{i=1}^{N}\frac{\left|p_i - r_i\right|}{p_i + r_i} \tag{15}$$

The **Theil's UI** (*UI*, unitless) [30] takes values between 0 and 1 and is given by

$$UI = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(p_i - r_i\right)^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} r_i^2} + \sqrt{\frac{1}{N}\sum_{i=1}^{N} p_i^2}} \tag{16}$$

The **Index of agreement** (*IOA*, unitless) takes values between 0 and 1, and is given by

$$IA = 1 - \frac{\sum_{i=1}^{N}\left(p_i - r_i\right)^2}{\sum_{i=1}^{N}\left(\left|p_i - \bar{r}\right| + \left|r_i - \bar{r}\right|\right)^2} = 1 - \frac{N \cdot MSE}{\sum_{i=1}^{N}\left(\left|p_i - \bar{r}\right| + \left|r_i - \bar{r}\right|\right)^2} \tag{17}$$

The **Pearson correlation coefficient** (*R*) [31, 32] is a unitless measure of linear correlation between two sets of data, commonly used in linear regression. In our case the two sets of data are the predicted values ***p*** and their real (or accurate) values ***r***. *R* is the covariance of the two variables, divided by the product of their standard deviations. Thus, it is essentially a normalized measurement of the covariance, such that the result always has a value between -1 and 1. An *R* value close to 1 indicates a strong positive relationship, while a value close to -1 indicates a strong negative relationship. Values near zero indicate no correlation. *R* is given by

$$R = \frac{Cov(r,p)}{Std(r) \cdot Std(p)} \tag{18}$$

where *Std*(*r*) and *Std*(*p*) are the standard deviations of the variables *r* and *p*, respectively, and *Cov*(*r*,*p*) is their covariance. The standard deviation is the square root of the variance, for each variable. According to statistics, in case we examine the whole population (not only a sample of it), the observed variances and the observed covariance are given by the formulas

$$Var_P(r) = \left(Std_P(r)\right)^2 = \sigma_r^2 = \frac{1}{N}\sum_{i=1}^{N}\left(r_i - \bar{r}\right)^2 \tag{19}$$

$$Var_P(p) = \left(Std_P(p)\right)^2 = \sigma_p^2 = \frac{1}{N}\sum_{i=1}^{N}\left(p_i - \bar{p}\right)^2 \tag{20}$$

$$Cov_P(r,p) = Cov_P(p,r) = \frac{1}{N}\sum_{i=1}^{N}\left(p_i - \bar{p}\right)\left(r_i - \bar{r}\right) \tag{21}$$

It can be statistically proven that in case a sample of the population is only examined (not the entire population), it is more precise to use (*N*-1) in the denominator of the above three formulas, to get an unbiased estimation of the variance and the covariance of the population. In this case, the estimated variances and the estimated covariance are given by

$$Var_S(r) = s_r^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(r_i - \bar{r}\right)^2 \tag{22}$$

$$Var_S(p) = \left(Std_S(r)\right)^2 = s_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(p_i - \bar{p}\right)^2 \tag{23}$$

$$Cov_S(r,p) = \left(Std_S(r)\right)^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(p_i - \bar{p}\right)\left(r_i - \bar{r}\right) \tag{24}$$

In Eq. (18) of *R*, it does not matter if the population formulas (observed covariance and observed standard deviations) or the sample formulas (estimated covariance and estimated standard deviations) are used, but the three must match. The reason that we can use either version of these values is because the *N*s or (*N*-1)s will “cancel” as they appear the same number of times in the numerator as in the denominator. Thus, the value of R does not depend on N (or N-1). Another property of the correlation coefficient is that it has no units. The correlation coefficient is a unitless measure with fixed extremes. It should also be noted that there is a special relationship between the quantities R, σ_r, σ_p and *CRMSD*, which is the basis for the construction of a Taylor diagram, as will be discussed in section 2.3.

The Pearson correlation coefficient R between the two variables is also given by the formulas

$$R = \frac{\sum_{i=1}^{N}(p_i - \bar{p})(r_i - \bar{r})}{\sqrt{\sum_{i=1}^{N}(p_i - \bar{p})^2} \cdot \sqrt{\sum_{i=1}^{N}(r_i - \bar{r})^2}} \tag{25}$$

$$R = \frac{N\sum_{i=1}^{N} x_i p_i - \sum_{i=1}^{N} r_i \cdot \sum_{i=1}^{N} p_i}{\sqrt{N\sum_{i=1}^{N} r_i^2 - \left(\sum_{i=1}^{N} r_i\right)^2} \cdot \sqrt{N\sum_{i=1}^{N} p_i^2 - \left(\sum_{i=1}^{N} p_i\right)^2}} \tag{26}$$

Observing Eq. (25), one can see some similarities with the formula of the angle θ between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, based on the dot product of the two vectors:

$$\cos\theta = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\| \, \|\boldsymbol{v}\|} \tag{27}$$

In fact, for centered data (i.e., data which have been shifted by the sample means of their respective variables so as to have an average of zero for each variable), the correlation coefficient can also be viewed as the cosine of the angle θ between the two observed vectors in the N-dimensional space (i.e. N observations of each variable) [33].

Although the Pearson correlation coefficient R between two variables, in our case the predicted values $\boldsymbol{p}$ and the real values $\boldsymbol{r}$, has a clear definition, this is not the case for the *Coefficient of Determination*, usually denoted as R^2, which has various meanings and definitions and it is many times a source of confusion, especially when used in nonlinear regression models [34, 35]. More about the Coefficient of Determination will be discussed later on, in section 2.2.

Another unitless error metric is **Variance Accounted For** (*VAF*), also called "Explained Variance", and it is given by

$$VAF = 1 - \frac{VAR(p\text{-}r)}{VAR(r)} = 1 - \frac{VAR(e)}{VAR(r)} = 1 - \frac{\sum_{i=1}^{N}(e_i - \bar{e}_i)^2}{\sum_{i=1}^{N}(r_i - \bar{r})^2} \tag{28}$$

Table 1 summarizes the 14 error metrics presented above and gives details on their units, their ranges and their value in case of a perfect match (i.e. zero error case). *MSE* is not included in the table, as it is directly related to *RMSE* which is included in the table. Similarly, *CMSD* is not included in the table, but *CRMSD* is. Note that the ranges reported in the table are valid for data sets with positive values only, i.e. $x_i>0$, $p_i>0$, for $i=1, 2, \ldots, N$.

Table 1: Error metrics, their units, range, and perfect match (target) value.

ID	Metric	Abbreviation	Units	Range	Perfect match value
1	Mean Bias	*MB*	Units of x, p	[-∞, +∞]	0
2	Mean Absolute Gross Error	*MAGE*	Units of x, p	[0, +∞]	0
3	Root Mean Squared Error	*RMSE*	Units of x, p	[0, +∞]	0
4	Centered Root Mean Square Difference	*CRMSD*	Units of x, p	[0, +∞]	0
5	Mean Normalized Bias	*MNB*	Unitless	[-1, +∞]	0
6	Mean Normalized Gross Error	*MNGE*	Unitless	[0, +∞]	0
7	Normalized Mean Bias	*NMB*	Unitless	[-1, +∞]	0
8	Normalized Mean Error	*NME*	Unitless	[0, +∞]	0
9	Fractional Bias	*FB*	Unitless	[-2, 2]	0
10	Fractional Gross Error	*FGE*	Unitless	[0, 2]	0
11	Theil's UI	*UI*	Unitless	[0, 1]	0
12	Index of agreement	*IOA*	Unitless	[0, 1]	1
13	Pearson correlation coefficient	*R*	Unitless	[-1, 1]	1
14	Variance Accounted For	*VAF*	Unitless	[-∞, 1]	1

2.2 Linear Regression model and the Coefficient of Determination R^2

Linear Regression model

Linear regression is a simple way to model the relationship between two (or more) variables. The equation in case of two variables has the general form $\hat{p}=a+b\cdot r$, where $\hat{p}$ is the dependent variable (the outcome of the linear regression model) and r is the independent variable. In other words, given the r_i values which are the real (target) values and the p_i values which are the predictions of our original model, $\hat{p}$ is the best prediction made by another model, the linear regression model. When the independent variable r is plotted on the horizontal axis, and $\hat{p}$ is plotted on the vertical axis, then b is the slope of the line and a is the $\hat{p}$ -intercept, as follows:

$$a=\frac{\left(\sum_{i=1}^{N}p_i\right)\cdot\left(\sum_{i=1}^{N}r_i^2\right)-\left(\sum_{i=1}^{N}r_i\right)\cdot\left(\sum_{i=1}^{N}r_i\cdot p_i\right)}{N\sum_{i=1}^{N}r_i^2-\left(\sum_{i=1}^{N}r_i\right)^2} \tag{29}$$

$$b = \frac{N\left(\sum_{i=1}^{N} r_i \cdot p_i\right) - \left(\sum_{i=1}^{N} r_i\right)\cdot\left(\sum_{i=1}^{N} p_i\right)}{N\sum_{i=1}^{N} r_i^2 - \left(\sum_{i=1}^{N} r_i\right)^2} \tag{30}$$

The equation of the linear model has the form

$$\hat{p}_i = a + b \cdot r_i \tag{31}$$

For this linear regression model, we define the following three variables, *RSS*, *ESS* and *TSS*, as follows

$$RSS = \sum_{i=1}^{N} \left(\hat{p}_i - p_i\right)^2 \tag{32}$$

$$ESS = \sum_{i=1}^{N} \left(\hat{p}_i - \overline{p}\right)^2 \tag{33}$$

$$TSS = \sum_{i=1}^{N} \left(p_i - \overline{p}\right)^2 = N \cdot \sigma_p^2 \tag{34}$$

where RSS is the Residual Sum of Squares, ESS is the Explained Sum of Squares and TSS is the Total Sum of Squares, which is also equal to the sum of ESS and RSS:

$$TSS = ESS + RSS \tag{35}$$

Coefficient of Determination R^2 in linear regression

Using the linear model, the Coefficient of Determination (R^2) can be defined in terms of RSS and TSS as

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{N} \left(\hat{p}_i - p_i\right)^2}{\sum_{i=1}^{N} \left(p_i - \overline{p}\right)^2} \tag{36}$$

or equivalently in terms of ESS and TSS as

$$R^2 = \frac{ESS}{TSS} = \frac{\sum_{i=1}^{N} \left(\hat{p}_i - \overline{p}\right)^2}{\sum_{i=1}^{N} \left(p_i - \overline{p}\right)^2} \tag{37}$$

ESS is the squared error that can be explained by the linear model and TSS is the total squared error. Using Eq. (37) we can conclude that the Coefficient of Determination (R^2) is the ratio of the variance that can be explained by the linear model, to the total variance. Therefore, the higher the R^2 value, the more useful the linear model is.

When using the above formulas and the linear model, it can be proven that R^2 is in fact the square of the Pearson correlation coefficient R and that R^2 takes values in the range [0, 1] (between 0 and 100%). It is important to note the assumption of linear relationship for all the above statistics and formulas to work fine.

Coefficient of Determination in nonlinear regression

Things about the Coefficient of Determination get quite complicated when using nonlinear regression models or other nonlinear models, such as ANN-based predictions. In particular, in these cases, if R^2 is calculated as the ratio of the variance explained by the model to the total variance, as for example Eqs (36) and (37) suggest, then weird statistics arise, and these underlying assumptions are incorrect. In such cases, explained variance (*ESS*) plus error variance (*RSS*) do not add up to the total variance (*TSS*). As a result, the calculated Coefficient of Determination isn't necessarily between 0 and 100% and it can even take negative values! In these cases, a negative Coefficient of Determination is not a mathematical impossibility. It simply means that the chosen model fits the data very poorly.

This problem completely undermines R^2 in the context of nonlinear regression and has been highlighted by a number of researchers [34-37]. Kvålseth [35] presents 8 different formulas for the definition of R^2 that appear throughout the literature, highlighting their differences, the confusion caused and some common mistakes in their use. The author presents the properties that R^2 should have as a measure of "goodness of fit", and he states that none of the eight alternative formulas provided for R^2 possesses all of these properties, although some come close.

Many times, in nonlinear regression (i.e. when the model used is not the linear regression model), the following formula, which is "equivalent" to the one of Eq. (36) of the linear model, is used for the calculation of the Coefficient of Determination:

$$RSquared = 1 - \frac{\sum_{i=1}^{N} \left(r_i - p_i\right)^2}{\sum_{i=1}^{N} \left(r_i - \bar{r}\right)^2} \tag{38}$$

In the above formula, we intentionally use the term *RSquared* to differentiate it from R^2 (the square of the Pearson correlation coefficient R) given by the previous formulas. Although Eq. (38) uses the concept of the ratio of the "variance that can be explained by the model" to the "total variance", the value it yields is not the square of the Pearson correlation coefficient R and it can take negative values. Actually, the range of Eq. (38) is (-∞, 1]. *RSquared* is a widely used measure in the industry to measure the performance of regression models, but there are serious problems with its use that can misguide machine learning engineers and researchers. In general, in nonlinear modeling, one always needs to make a certain choice for the definition for the Coefficient of Determination and state it very clearly, to avoid confusion and misunderstandings.

According to Spiess and Neumeyer [36] "*Researchers should be aware that R^2 is inappropriate when used for demonstrating the performance or validity of a certain nonlinear model. It should ideally be removed from scientific literature dealing with nonlinear model fitting or at least be* *supplemented with other methods such as AIC or BIC or used in context to other models in question*".

Adjusted R^2

One of the drawbacks of the Coefficient of Determination (in either its R^2 or *RSquared* form) is that if more features are added to a model, its value increases. This happens even though the features added to the model are not intrinsically predictive. For this reason, the *Adjusted* R^2 was introduced, as a modified version of R^2 that has been adjusted for the number of predictors in the model. The *Adjusted* R^2 is always less or equal to R^2 as it adjusts for the increasing predictors and only shows improvement if there is a real improvement. It is given by the formula

$$R^2 = 1 - \frac{N-1}{N-k-1}\left(1-R^2\right) \tag{39}$$

Where N is the number of data points (observations) and k is the number of features (independent variables) in the model.

2.3 Taylor diagram

Taylor diagrams are mathematical diagrams designed to graphically indicate which of several approximate representations (or models) of a system, process, or phenomenon is most realistic. A Taylor diagram [29] combines three statistical quantities, namely the *CRMSD*, the Pearson correlation coefficient and the Standard Deviations in a single diagram that is easy to read and interpret. The Taylor diagram can be used to summarize the relative merits of a collection of different models or to track changes in performance as a model is modified. The *CRMSD* metric is used in a Taylor diagram to express the prediction error of a model. The following equation is the basis of the Taylor diagram:

$$CRMSD^2 = \sigma_r^2 + \sigma_p^2 - 2\sigma_r\sigma_p R \tag{40}$$

Every prediction vector (set), such as $\boldsymbol{p}$, can depicted as a single point in the Taylor diagram, which shows the following:

- Its *CRMSD* error metric with reference to the real values
- Its standard deviation σ_p and its relationship to the standard deviation of the real values, σ_r
- Its correlation with the real values, R.

The diagram also demonstrates the real values set as the "ground truth" reference point in its horizontal axis. It also shows the corresponding values for the real data, i.e. obviously zero for the error, σ_r for the standard deviation and $R = 1$ for the correlation with itself. In this diagram the distance between each model and the reference point (labeled "REF") is a measure of how realistically each model reproduces observations. The Taylor diagram will be explained in detail through a numerical example in section 3.6.

3 SIMPLE NUMERICAL EXAMPLE

3.1 Data sets

In this section we consider and examine a simple numerical example as a scenario where the "real" or "observed" (target) data $\boldsymbol{r}$ is a 10×1 vector and the predicted data $\boldsymbol{p}$ (as a regression model output) is another 10×1 vector, as shown in Table 2 and depicted in Figure 1.

Table 2: "Real" (target) values and model-predicted values for the numerical example.

Data ID	Real value, r_i	Predicted value, p_i
1	287	311
2	40	55
3	68	60
4	256	302
5	115	87
6	190	152
7	300	297
8	222	235
9	145	165
10	172	136

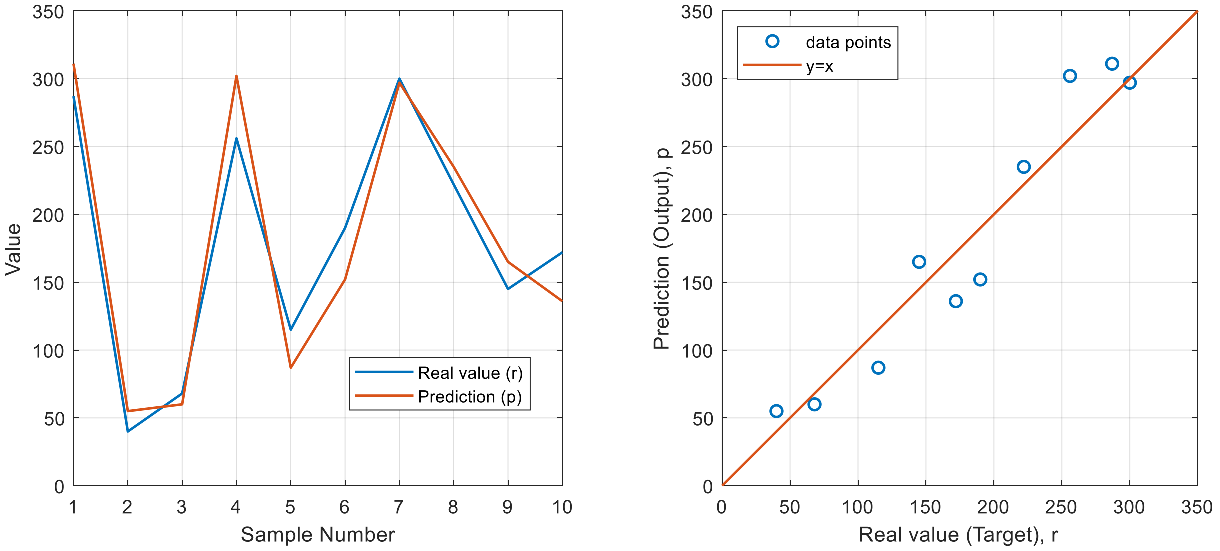

Figure 1: "Real" (target) values and model-predicted values for the numerical example.

3.2 Statistical quantities of the two sets

Table 3 shows some basic statistical quantities for the two data sets.

Table 3: Statistical quantities of the "Real" (target) values and model-predicted values.

Statistical quantity	Symbol	Real value, *r*	Predicted value, *p*
Minimum	Min	40	55
Maximum	Max	300	311
Range	Range	260	256
Mean	Mean	179.5	180
Median	Median	181	158.5
Variance (Population)	$\mathrm{Var_P} = \sigma^2$	$\sigma_r^2 = 7114.45$	$\sigma_p^2 = 9037.8$
Standard Deviation (Population)	$\mathrm{Std_P} = \sigma$	$\sigma_r = 84.34720$	$\sigma_p = 95.06734$
Variance (Sample)	$\mathrm{Var_S} = s^2$	$s_r^2 = 7904.94444$	$s_p^2 = 10042$
Standard Deviation (Sample)	$\mathrm{Std_S} = s$	$s_r = 88.90975$	$s_p = 100.20978$

3.3 Error metrics values

Table 4 shows the 14 calculated error metrics for the prediction ***p*** of the real data ***r***.

Table 4: Error metrics values.

ID	Error metric	Value (Target)	ID	Error metric	Value (Target)
1	*MB*	0.5 (0)	8	*NME*	0.12869 (0)
2	*MAGE*	23.1 (0)	9	*FB*	-0.01214 (0)
3	*RMSE*	26.61391 (0)	10	*FGE*	0.16151 (0)
4	*CRMSD*	26.60921 (0)	11	*UI*	0.06622 (0)
5	*MNB*	0.00544 (0)	12	*IOA*	0.97766 (1)
6	*MNGE*	0.16152 (0)	13	*R*	0.96302 (1)
7	*NMB*	0.00279 (0)	14	*VAF*	0.90048 (1)

3.4 Linear regression model

By applying Eqs (29) and (30) to our data set, we obtain a = -14.83122 and b = 1.085410678 for the linear regression model, i.e. $\hat{p} = -14.83122 + 1.085410678 \cdot r$, which means that the linear regression line crosses the vertical axis at p=b=1.085410678 and the horizontal axis at r=-b/a=0.073184196. Table 5 shows the predictions of the linear model among related other statistical quantities.

Table 5: Predictions of the linear model and other statistical quantities.

Data ID	**Real value, r_i**	**Predicted value, p_i**	$(p_i-\bar{p})^2$	**Linear model prediction, $\hat{p}_i$**	$\hat{p}_i-p_i$	$(\hat{p}_i-p_i)^2$	$\hat{p}_i-\bar{p}$	$(\hat{p}_i-\bar{p})^2$
1	287	311	17161	296.68165	-14.31835	205.01521	116.68165	13614.60696
2	40	55	15625	28.58521	-26.41479	697.74111	-151.41479	22926.43852
3	68	60	14400	58.97671	-1.02329	1.04712	-121.02329	14646.63687
4	256	302	14884	263.03392	-38.96608	1518.35563	83.03392	6894.63135
5	115	87	8649	109.99101	22.99101	528.58660	-70.00899	4901.25851
6	190	152	784	191.39681	39.39681	1552.10881	11.39681	129.88733
7	300	297	13689	310.79199	13.79199	190.21890	130.79199	17106.54379
8	222	235	3025	226.12995	-8.87005	78.67772	46.12995	2127.97264
9	145	165	225	142.55333	-22.44667	503.85292	-37.44667	1402.25297
10	172	136	1936	171.85942	35.85942	1285.89800	-8.14058	66.26904
SUM:			**90378 = TSS**			**6561.50201 = RSS**		**83816.49799 = ESS**

We see that the values of *RSS*, *ESS* and *TSS* satisfy the equality of Eq. (35). Also, R^2 can be calculated by either *ESS*/*TSS* = 83816.49799/90378 = 0.92740 or by 1-*RSS*/*TSS* = 1-6561.50201/90378 = 0.92740. The linear regression model is presented in Figure 2 as a line.

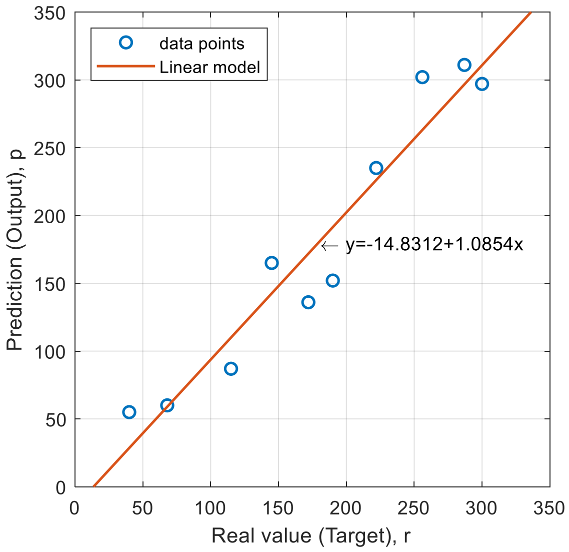

Figure 2: "Real" values, predicted values and the linear regression model.

Taking the square root of R^2 (square root of 0.92740), we find 0.96302 which is indeed the calculated value of the Pearson Correlation Coefficient *R* for the model, as shown in Table 4. Using the linear model, all formulas work correctly, and the calculated Coefficient of Determination (R^2) is indeed the square of *R*.

3.5 Direct calculation of the Coefficient of Determination for the model

We will now apply the formula of Eq. (38) for the predictive model itself, without going through the linear model first. In this case, for our example, we have:

$$\sum_{i=1}^{N}\left(p_i-r_i\right)^2=7083 \tag{41}$$

$$\sum_{i=1}^{N}\left(r_i-\bar{r}\right)^2=71144.5 \tag{42}$$

$$RSquared=1-\frac{\sum_{i=1}^{N}\left(p_i-r_i\right)^2}{\sum_{i=1}^{N}\left(r_i-\bar{r}\right)^2}=1-\frac{7083}{71144.5}=0.90044 \tag{43}$$

We see that this new value of *RSquared* is different from the one calculated before using the linear model (R^2=0.92740). As mentioned before, this formula can also give negative results for *RSquared* in specific cases. For example, if one changes only the 7th element of the ***p*** vector from p_7=297 to the new value of 40, then the above formula will give a value of *RSquared*=1-74674/71144.5 = -0.04961, while *R* using the linear model will then be *R*=0.57904 (and thus R^2=0.33528). It is clear that a definition of the formula used for the calculation of R^2 must always be given, to avoid confusion, misunderstanding and inaccuracies.

3.6 Taylor diagram

In this example case, the standard deviation of the real data is σ_{r} = 84.34720, the standard deviation of the predicted values is σ_{p} = 95.06734, while the Centered Root Mean Square Difference (*CRMSD*) error for the prediction model is 26.60921 and the Pearson correlation coefficient (*R*) is 0.96302. The Taylor diagram is presented in Figure 3. The values used for plotting the diagram for the reference point ("REF") and the prediction "Model" point are shown in Table 6. The reference point ("REF", ground truth values) always has zero error in comparison to itself (*CRMSD*=0) and perfect correlation with itself (*R*=1), but it still has its own standard deviation value which is depicted in the horizontal axis.

Table 6: Taylor diagram values.

Taylor parameter	REF point	Prediction Model point
Standard deviation, σ	84.34720	95.06734
CRMSD	0	26.60921
R	1	0.96302

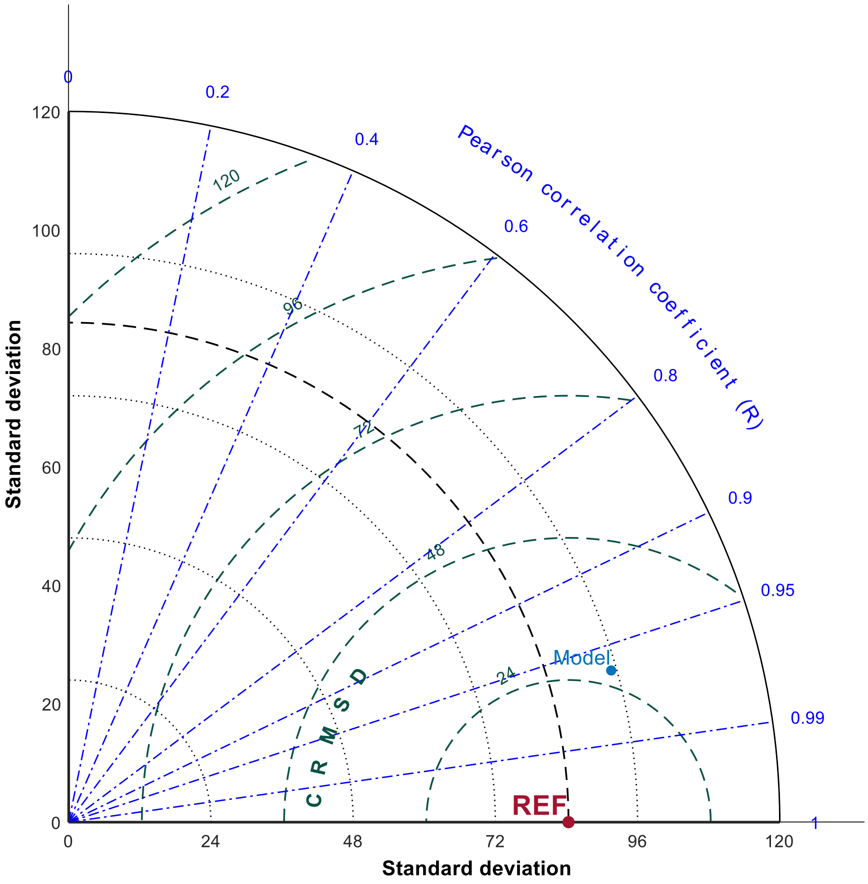

Figure 3: Taylor diagram for the numerical example.

4 NUMERICAL INVESTIGATION OF THE ERROR METRICS

4.1 Random guessing

First, we investigate the performance of the various error metrics in assessing a prediction which is based on pure random guessing. It is obvious that a good metric should give a low score when asked to assess the result of random guessing on the real values. To do this numerical test, we use the Monte Carlo Simulation (MCS) method [28]. We generate a vector ***r*** of 100 elements (100×1) having random values in the range [10, 100], following uniform distribution. Then we generate a vector ***p*** of "predictions" with 100 elements (100×1) having again random values in the range [10, 100]. As a result, the prediction is based on random guessing on the same interval as the one of the original data. We perform 1000 Monte Carlo simulations, and we examine the distributions of the values of the 10 normalized (unitless) metrics that have been presented, i.e. *MNB*, *MNGE*, *NMB*, *NME*, *FB*, *FGE*, *UI*, *IOA*, *R*, and *VAF*. Of these, the first 7 (*MNB*, *MNGE*, *NMB*, *NME*, *FB*, *FGE* and *UI*) will take the value of 0 for a perfect match, while the next 3 (*IOA*, *R* and *VAF*) will take the value of 1 for a perfect match. An example of this model (i.e. as one Monte Carlo simulation) is depicted in Figure 4 where the predictions ***p*** (100 elements) are completely random numbers in the interval [10, 100]. This causes the spread in the diagram on the right.

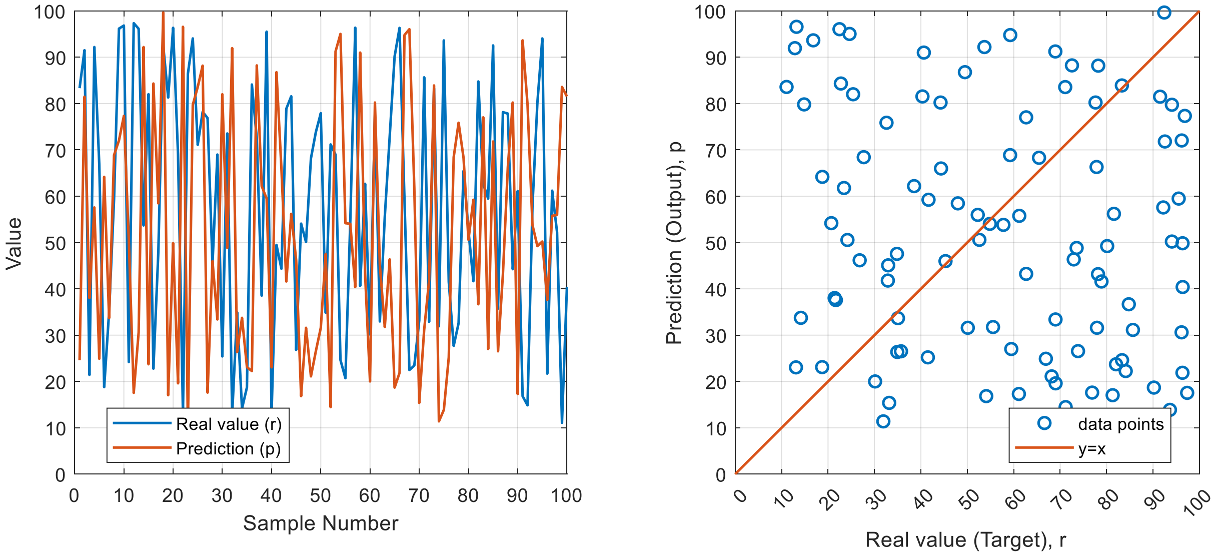

Figure 4: "Real" (target) values and randomly generated model-predicted values.

In statistical analysis, we must be very careful when trying to interpret statistical results and assess their significance as many times, the metrics we use can be random variables themselves [38]. In the case of this test example, the distributions of the metrics' values are presented as histograms in Figure 5. We see that *NMB* (3rd on the top row) and *FB* (5th on the top row) take values close to 0 in many of the cases, which means that these metrics falsely identify random guessing as giving a good prediction. This indicates poor performance for these two metrics in this test example. The other 8 metrics have values which are far away from the "perfect match" value of either 0 (for the first seven) or 1 (for the last three).

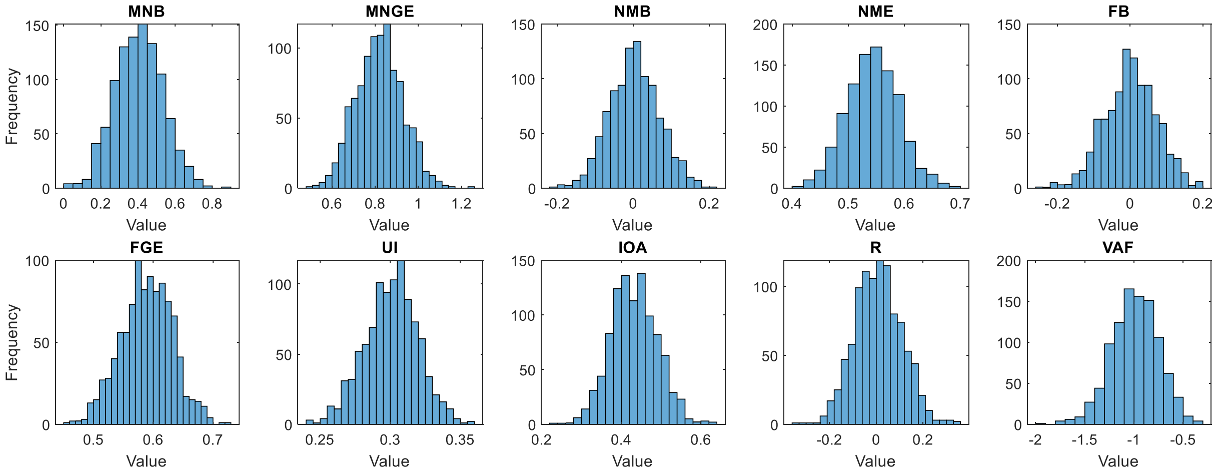

Figure 5: Histograms of the 10 error metric values for a prediction based on random guessing.

Figure 6 presents the obtained values of the 10 metrics as mean or median values over 1000 simulations. Again, we see that *NMB* and *FB* have values close to zero, that falsely indicate good prediction performance.

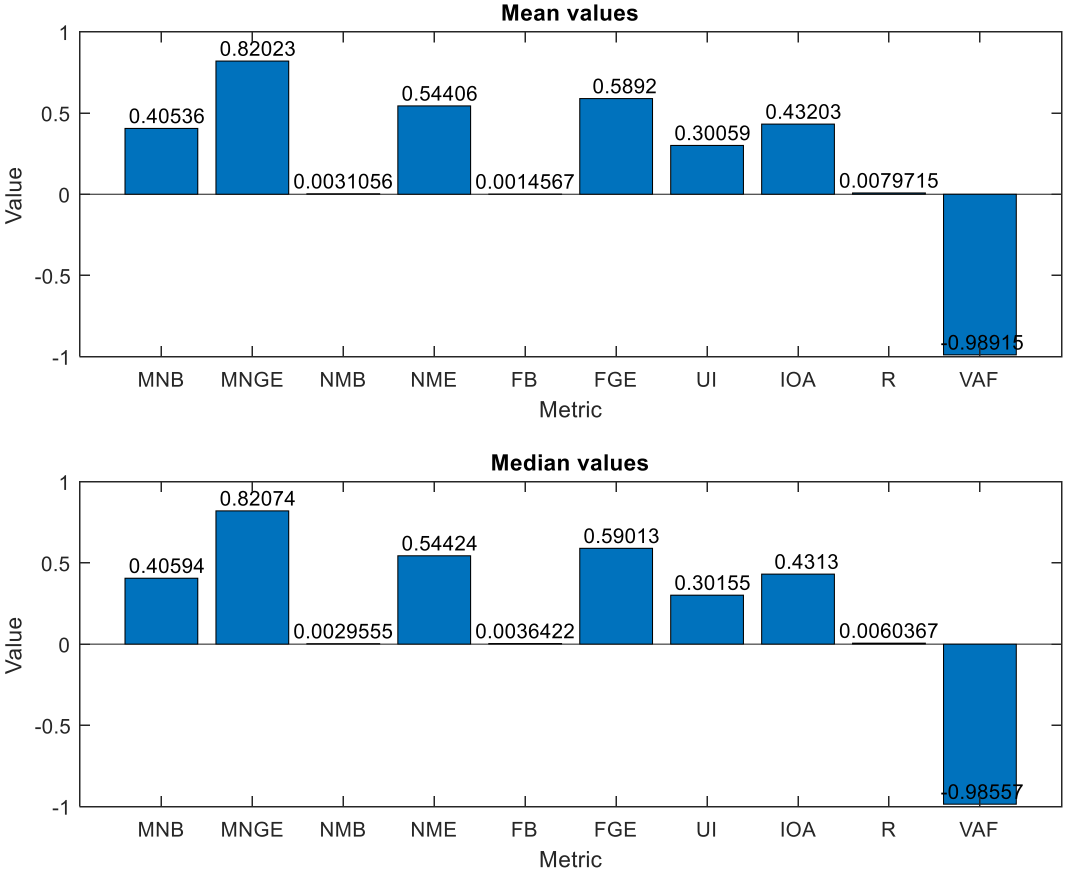

Figure 6: Mean and median values of the 10 unitless error metrics for a prediction based on random guessing, over 1000 Monte Carlo simulations.

4.2 Scenarios with random noise

In this investigation, we introduce some noise (error) in the prediction. We use again the Monte Carlo Simulation (MCS) method [28] with 1000 samples.

First random noise scenario

We generate a vector ***r*** of 100 elements (100×1) having random values in the range [10, 100]. Then we generate a vector ***p*** of "predictions" with 100 elements (100×1). This ***p*** vector is based on ***r***, with the introduction of some artificial "noise" (error) according to the following formula [39] for each element of ***p***:

$$p_i = r_i \cdot (1 + NR \cdot \xi) \tag{44}$$

where p_i is the *i*-th component of the prediction vector, r_i is the *i*-th component of the real (target) vector, *NR* (noise ratio) is the percentage of noise added to the predicted data, and ξ is a uniformly distributed random number in the range [-1, 1]. In our case, the noise ratio *NR* takes values from 0% to 100% with 10% increments. A value *NR*=0 means that there is no noise, i.e. no error in the prediction and as a result, $p_i = r_i$. A value *NR*=10%=0.1 means that the prediction, p_i will be a random number uniformly distributed in the range [0.9r_i, 1.1r_i], while a value *NR*=100%=1 means that the prediction, p_i will be a random number uniformly distributed in the range [0, 2r_i]. The results of the investigation are presented in Figure 7 for all the ten unitless error metrics.

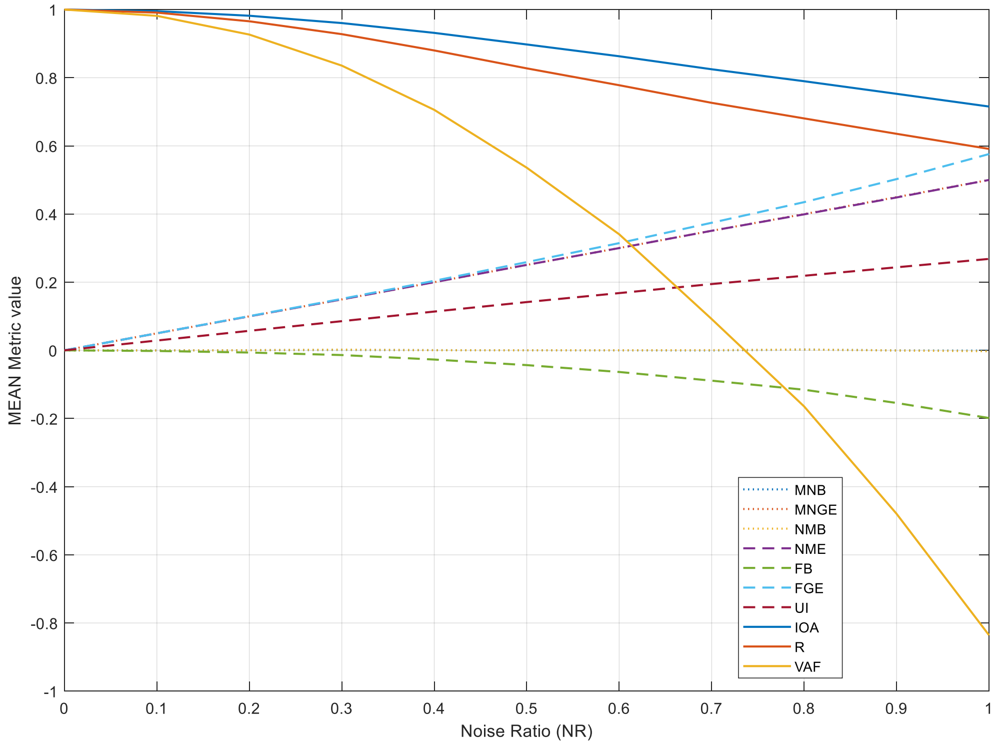

Figure 7: Scenario 1: Mean values of the 10 error metrics for various Noise Ratios (*NR*).

We see that the first 7 metrics start from zero, correctly indicating no error for the case *NR*=0. The last 3 metrics (*IOA*, *R* and *VAF*) start from the value of 1, indicating again no error for the case *NR*=0. The various error metrics take different paths in general, with the exception of two pairs:

(i) *MNB* and *NMB* exhibit a similar poor performance and their mean values almost coincide, being close to zero for all ranges of *NR*, which does not make much sense for the metrics. As a result, these two metrics fail to predict the error for all *NR* scenarios. The *MNB* mean values range from -9.8650E-4 to 0.0023 and the *NMB* mean values range from -0.0027 to 0.0024.

(ii) The other two metrics that exhibit similar performance with each other, are *MNGE* and *NME* with their mean values almost coinciding, but not being close to zero, which is interesting. If we look closer at the data, we will see that the individual values of *MNGE* and *NME* are not the same for every simulation, but their mean values among 1000 simulations are very close to each other and as a result the two curves almost coincide. The mean of both metrics starts at zero (for *NR*=0) and ends to the value of 0.4999 for *MNGE* and 0.5 for *NME* (for *NR*=1).

The mean of the *VAF* metric exhibits a big variability along the *NR* values, starting at 1 (for *NR*=0) and ending to the value of -0.8358 (for *NR*=1). Similarly, the *R* value drops from 1 (for *NR*=0) to the value of 0.5911 (for *NR*=1). It is interesting that the *R* value is still quite high, at 0.5911 even in the case of *NR*=100%, which shows us that *R* values can sometimes be misleading.

Second random noise scenario

It is interesting to examine another scenario, where noise is added again using a similar pattern, with the difference being that it is applied uniformly to the whole vector ***r***, instead of its individual elements. We generate a vector ***r*** of 100 elements (100×1) having random values in the range [10, 100]. Then we generate a vector ***p*** of "predictions" with 100 elements (100×1), based on the formula

$$\boldsymbol{p} = \boldsymbol{r} \cdot \left(1 + NR \cdot \xi\right) \tag{45}$$

Now, the prediction vector ***p*** is simply a multiple of ***r***, as a whole. This is a theoretical case which is rarely expected to occur in practice, and it corresponds to a prediction model which systematically overestimates or underestimates all the values by the same factor. For example, all predicted values are *x*% larger (or smaller) than the real values, or similar. It is interesting to see the behavior of the different error metrics in this case and compare the results to the ones of the first scenario. Applying the same methodology as before, we end up to the diagram of Figure 8.

In this 2$^{\text{nd}}$ Scenario, the patterns we see for the mean curves of *MNB* (and the similar *NMB*), *MNGE* (and the similar *NME*), *FB*, *FGE*, *UI*, and *IOA* are similar to the ones of Scenario 1, with only small differences. Yet *R* exhibits a completely different pattern, as in Scenario 2 it has a constant value of 1 for all *NR* cases. This makes sense and it is because in all simulated cases, the predictions ***p*** are exact multiples or the real values ***r***. So, in every case, the correlation is perfect and *R*=1, although there is an error. This can be considered as a limitation of the *R* value as a performance metric for predictions, as a perfect prediction requires *R*=1 (a necessary condition), but *R*=1 itself does not guarantee a perfect prediction (not a sufficient condition) simply because *R* is in fact not an error metric but a correlation coefficient which tells us if a linear relationship exists between the real and the predicted values. Interestingly, *VAF* shows a different pattern in Scenario 2, in comparison to Scenario 1. The end point of the mean *VAF* curve is now 0.68325 compared to -0.83582 for Scenario 1.

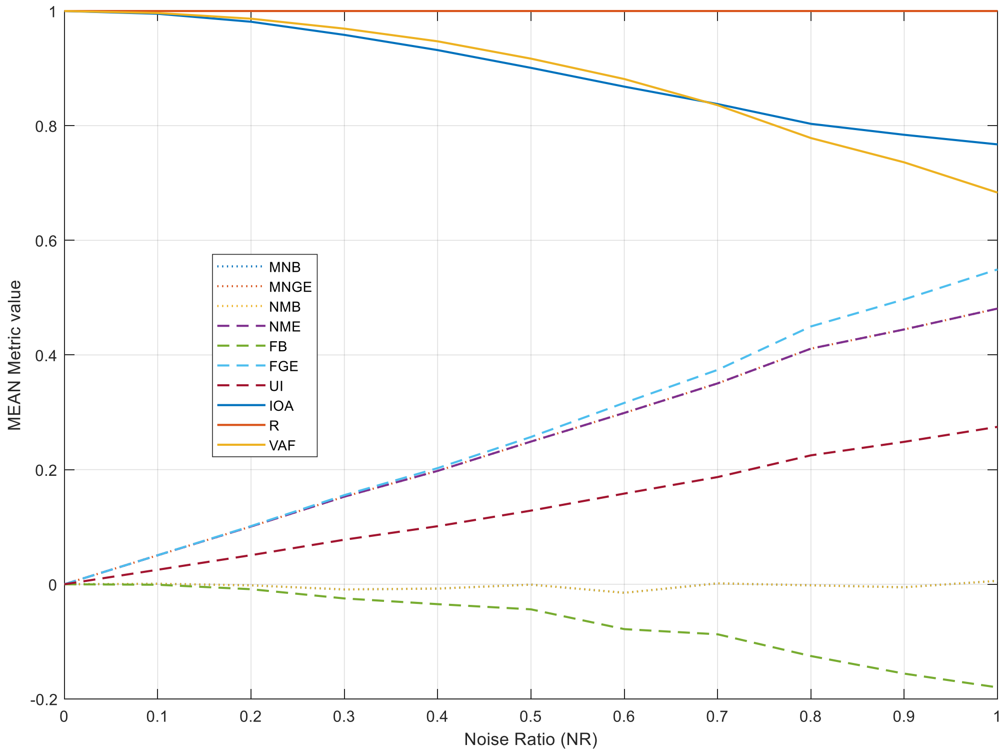

Figure 8: Scenario 2: Mean values of the 10 error metrics for various Noise Ratios (*NR*).

5 CONCLUSIONS

In this article, we examined some of the most popular regression related metrics used for evaluating the performance of regression and machine learning models and we highlighted the importance of the usage of the metrics to obtain good predictions. 14 error metrics were presented theoretically and using a simple numerical example. Based on three of these metrics, the Taylor diagram can be constructed which visualizes the standard deviation, the Pearson Correlation Coefficient, and the Centered Root Mean Square Difference metrics in a single elegant diagram. We examined the concept of the Coefficient of Determination (R^2) both in the linear regression model and in the more general case and we highlighted its limitations and how it can become a source of confusion. Ten of the examined metrics are unitless (*MB*, *MAGE*, *RMSE*, *CRMSD*, *MNB*, *MNGE*, *NMB*, *NME*, *FB*, *FGE*, *UI*, *IOA*, *R*, and *VAF*). These unitless metrics were further investigated through an analysis with Monte Carlo Simulation based on (i) random guessing and (ii) the addition of random noise with various noise ratios to the predicted values. Some of the metrics showed a poor performance, while others exhibit a good performance as evaluation measures of the “goodness of fit”.

REFERENCES

[1] Draper, N.R. and H. Smith, *Applied regression analysis*. Applied Regression Analysis. 2014. 1-716.

[2] Breiman, L., *Random Forests.* Machine Learning, 2001. **45**(1): p. 5-32 DOI: https://doi.org/10.1023/A:1010933404324.

[3] Drucker, H., et al., *Support vector regression machines*, in *Proceedings of the 9th International Conference on Neural Information Processing Systems*. 1996, MIT Press: Denver, Colorado. p. 155–161.

[4] Lagaros, N.D. and M. Papadrakakis, *Learning improvement of neural networks used in structural optimization.* Advances in Engineering Software, 2004. **35**(1): p. 9-25.

[5] Plevris, V. and G. Tsiatas, *Computational Structural Engineering: Past Achievements and Future Challenges.* Frontiers in Built Environment, 2018. **4**(21): p. 1-5 DOI: https://doi.org/10.3389/fbuil.2018.00021.

[6] Asteris, P.G. and V. Plevris, *Anisotropic masonry failure criterion using artificial neural networks.* Neural Computing and Applications, 2017. **28**(8): p. 2207-2229 DOI: https://doi.org/10.1007/s00521-016-2181-3.

[7] Plevris, V., G. Solorzano, and N. Bakas, *Literature review of historical masonry structures with machine learning*, in *7th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2019)*, M. Papadrakakis and M. Fragiadakis, Editors. 2019, ECCOMAS: Crete, Greece. p. 1547-1562. DOI: https://doi/org/10.7712/120119.7018.21053.

[8] Plevris, V., et al., *Literature review of masonry structures under earthquake excitation utilizing machine learning algorithms*, in *6th ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2017)*. 2017, ECCOMAS: Rhodes, Greece. p. 2685-2694. DOI: https://doi.org/10.7712/120117.5598.18688.

[9] Plevris, V. and P.G. Asteris, *Modeling of Masonry Failure Surface under Biaxial Compressive Stress Using Neural Networks.* Construction and Building Materials, 2014. **55**: p. 447-461 DOI: https://doi.org/10.1016/j.conbuildmat.2014.01.041.

[10] Asteris, P.G. and V. Plevris. *Neural network approximation of the masonry failure under biaxial compressive stress*. in *ECCOMAS Special Interest Conference - SEECCM 2013: 3rd South-East European Conference on Computational Mechanics, Proceedings - An IACM Special Interest Conference*. 2013.

[11] Plevris, V. and P.G. Asteris, *Modeling of masonry compressive failure using Neural Networks*, in *OPT-i 2014 - 1st International Conference on Engineering and Applied Sciences Optimization, Proceedings*. 2014. p. 2843-2861.

[12] Plevris, V. and P. Asteris, *Anisotropic failure criterion for brittle materials using Artificial Neural Networks*, in *5th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2015)*, M. Papadrakakis, V. Papadopoulos, and V. Plevris, Editors. 2015, ECCOMAS: Crete Island, Greece. p. 2259-2272. DOI: https://doi.org/10.7712/120115.3537.3648.

[13] Ahmad, A., V. Plevris, and Q.-u.-Z. Khan, *Prediction of Properties of FRP-Confined Concrete Cylinders Based on Artificial Neural Networks.* Crystals, 2020. **10**(9): p. 1-22 DOI: https://doi.org/10.3390/cryst10090811.

[14] Kandiri, A., F. Sartipi, and M. Kioumarsi, *Predicting Compressive Strength of Concrete Containing Recycled Aggregate Using Modified ANN with Different Optimization Algorithms.* Applied Sciences, 2021. **11**(2) DOI: http://doi.org/10.3390/app11020485.

[15] Dabiri, H., et al., *Compressive strength of concrete with recycled aggregate; a machine learning-based evaluation.* Cleaner Materials, 2022. **3**: p. 100044 DOI: https://doi.org/10.1016/j.clema.2022.100044.

[16] Ben Seghier, M.E.A., et al., *On the modeling of the annual corrosion rate in main cables of suspension bridges using combined soft computing model and a novel nature-inspired algorithm.* Neural Computing and Applications, 2021. **33**(23): p. 15969-15985 DOI: https://doi.org/10.1007/s00521-021-06199-w.

[17] Waris, M.I., et al., *Predicting compressive strength of CRM samples using Image processing and ANN*, in *IOP Conference Series: Materials Science and Engineering*. 2020, IOP Publishing. p. 012014. DOI: http://doi.org/10.1088/1757-899x/899/1/012014.

[18] Ahmadi, M., A. Kheyroddin, and M. Kioumarsi, *Prediction models for bond strength of steel reinforcement with consideration of corrosion.* Materials Today: Proceedings, 2021. **45**: p. 5829-5834 DOI: https://doi.org/10.1016/j.matpr.2021.03.263.

[19] Solorzano, G. and V. Plevris, *Design of Reinforced Concrete Isolated Footings under Axial Loading with Artificial Neural networks*, in *14th ECCOMAS Thematic Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control (EUROGEN 2021)*. 2021: Streamed from Athens, Greece. p. 118-131. DOI: https://doi.org/10.7712/140121.7955.18448.

[20] Plevris, V. and G. Solorzano, *Prediction of The Eigenperiods of MDOF Shear Buildings using Neural Networks*, in *8th ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering (COMPDYN 2021)*. 2021, ECCOMAS: Athens, Greece. p. 3894-3911. DOI: http://www.doi.org/10.7712/120121.8755.20415.

[21] Sharib, S., et al., *Prediction Models for Load Carrying Capacity of RC Wall through Neural Network*, in *14th ECCOMAS Thematic Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control (EUROGEN 2021)*. 2021, ECCOMAS: Streamed from Athens, Greece. p. 132-142. DOI: https://doi.org/10.7712/140121.7956.18529.

[22] Ahmadi, M., et al., *New empirical approach for determining nominal shear capacity of steel fiber reinforced concrete beams.* Construction and Building Materials, 2020. **234**: p. 117293 DOI: https://doi.org/10.1016/j.conbuildmat.2019.117293.

[23] Papadrakakis, M., N.D. Lagaros, and Y. Tsompanakis, *Optimization of large-scale 3-D trusses using evolution strategies and neural networks.* International Journal of Space Structures, 1999. **14**(3): p. 211-223 DOI: https://doi.org/10.1260/0266351991494830.

[24] Makridakis, S., E. Spiliotis, and V. Assimakopoulos, *Statistical and Machine Learning forecasting methods: Concerns and ways forward.* PLOS ONE, 2018. **13**(3) DOI: https://doi.org/10.1371/journal.pone.0194889.

[25] De Gooijer, J.G. and R.J. Hyndman, *25 years of time series forecasting.* International Journal of Forecasting, 2006. **22**(3): p. 443-473 DOI: https://doi.org/10.1016/j.ijforecast.2006.01.001.

[26] Botchkarev, A., *A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms.* Interdisciplinary Journal of Information, Knowledge, and Management, 2019. **14**: p. 45-76 DOI: https://doi.org/10.28945/4184.

[27] Botchkarev, A., *Evaluating Performance of Regression Machine Learning Models Using Multiple Error Metrics in Azure Machine Learning Studio.* SSRN, 2018 DOI: http://dx.doi.org/10.2139/ssrn.3177507.

[28] Plevris, V., *Innovative computational techniques for the optimum structural design considering uncertainties*. 2009, National Technical University of Athens: Athens, Greece. p. 312.

[29] Taylor, K.E., *Summarizing multiple aspects of model performance in a single diagram.* Journal of Geophysical Research: Atmospheres, 2001. **106**(D7): p. 7183-7192 DOI: https://doi.org/10.1029/2000JD900719.

[30] Bliemel, F., *Theil's Forecast Accuracy Coefficient: A Clarification.* Journal of Marketing Research, 1973. **10**(4): p. 444-446 DOI: https://doi.org/10.2307/3149394.

[31] Ahlgren, P., B. Jarneving, and R. Rousseau, *Requirements for a cocitation similarity measure, with special reference to Pearson's correlation coefficient.* Journal of the American Society for Information Science and Technology, 2003. **54**(6): p. 550-560 DOI: https:doi.org/10.1002/asi.10242.

[32] Benesty, J., et al., *Pearson Correlation Coefficient*, in *Noise Reduction in Speech Processing*, I. Cohen, et al., Editors. 2009, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 1-4. DOI: http://doi.org/10.1007/978-3-642-00296-0_5.

[33] Buda, A. and A. Jarynowski, *Life-time of correlations and its applications, Vol. 1*. 2010, ISBN: 978-83-915272-9-0.

[34] Nagelkerke, N.J.D., *A note on a general definition of the coefficient of determination.* Biometrika, 1991. **78**(3): p. 691-692 DOI: https://doi.org/10.1093/biomet/78.3.691.

[35] Kvalseth, T.O., *Cautionary Note about R^2*. The American Statistician, 1985. **39**(4): p. 279-285 DOI: https://doi.org/10.2307/2683704.

[36] Spiess, A.-N. and N. Neumeyer, *An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach.* BMC Pharmacology, 2010. **10**(1): p. 6 DOI: https://doi.org/10.1186/1471-2210-10-6.

[37] Wang, T. and S. Zhang, *Study on Linear Correlation Coefficient and Nonlinear Correlation Coefficient in Mathematical Statistics.* Studies in Mathematical Sciences, 2011. **3**(1): p. 58-63 DOI: http://doi.org/10.3968/j.sms.1923845220110301.4Z483.

[38] Altman, N. and M. Krzywinski, *P values and the search for significance.* Nature Methods, 2017. **14**(1): p. 3-4 DOI: 10.1038/nmeth.4120.

[39] Georgioudakis, M. and V. Plevris, *A Combined Modal Correlation Criterion for Structural Damage Identification with Noisy Modal Data.* Advances in Civil Engineering, 2018. **2018**(3183067): p. 20 DOI: https://doi.org/10.1155/2018/3183067.