Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences

M.Z. Naser, PhD, PE
School of Civil and Environmental Engineering and Earth Sciences, Clemson University, Clemson, SC, 29634, USA
Artificial Intelligence Research Institute for Science and Engineering (AIRISE) at Clemson University, Clemson, SC, 29634, USA
E-mail: mznaser@clemson.edu, m@mznaser.com, Website: www.mznaser.com

Amir H. Alavi, PhD
Department of Civil and Environmental Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA
E-mail: alavi@pitt.edu

Abstract

Machine learning (ML) is the field of training machines to achieve a high level of cognition and perform human-like analysis. Since ML is a data-driven approach, it seemingly fits into our daily lives and operations and complex and interdisciplinary fields. With the rise of commercial, open-source, and user-catered ML tools, a key question often arises whenever ML is applied to explore a phenomenon or a scenario: *what constitutes a good ML model?* Keeping in mind that a proper answer to this question depends on various factors, this work presumes that a good ML model *optimally performs and best describes the phenomenon on hand*. From this perspective, identifying proper assessment metrics to evaluate the performance of ML models is not only necessary but is also warranted. As such, this paper examines 78 of the most commonly-used performance fitness and error metrics for regression and classification algorithms, with emphasis on engineering applications.

Keywords: Error metrics; Machine learning; Regression; Classification.

1. Introduction

Learning is the process of seeking knowledge [1]. We, as humans, can learn from our daily interactions and experiences because we have the ability to communicate, reason, and understand. With the rapid technological advancement in computer sciences, computational intelligence has led to the development of modern cognitive and evaluation tools [2, 3]. One such tool is machine learning (ML) which is often described as a set of methods that, when applied, can allow machines to learn/understand meaningful patterns from data repositories; while maintaining minimal human interaction [4]. More specifically, a *"computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"* [5]. In other words, ML trains machines to understand real-world applications, use this knowledge to carry out pre-identified tasks with the goal of optimizing and improving the machines' performance with time and new knowledge. A closer look at the definition of ML infers that computers do not learn by reasoning but rather by algorithms.

From the perspective of this work, traditional statistical regression techniques are often used to carry out behavioral modeling wherein such techniques may suffer from large uncertainties, the need for the idealization of complex processes, approximation, and averaging widely varying prototype conditions. Furthermore, statistical analysis often assumes linear, or in some cases nonlinear, relationships between the output and the predictor variables, and these assumptions do not always hold true – especially in the context of engineering/real data. On the other hand, ML methods adaptively learn from experiences and extract various discriminators. One of the major advantages of ML approaches over the traditional statistical techniques is their ability to derive a relationship(s) between inputs and outputs without assuming prior forms or existing relationships. In other words, ML approaches are not confined to one particular space that requires the availability of physical representation but rather goes beyond that to explore hidden relations in data patterns [6–11].

While ML was initially developed for computer sciences, it is now an integral part of various fields including, energy/mechanical engineering [6–9], social sciences [10, 11], space applications [12, 13], among others [14–19]. Due to the availability of high-computationally powered machines and ease-of-access to data (thanks in part to the rise of Internet-of-Things and data-driven-applications), the utilization of ML into civil engineering, in general, and materials science, engineering in particular, has been duly noted in recent years [20–25].

An integral part of the wide spread of integrating ML into new research areas is due to the availability of user-friendly and easy-to-use software packages that simplifies the process of ML by utilizing pre-defined algorithms and training/validation procedure [26–30]. The availability of such tools, while facilitating ML analysis and providing new opportunities for researchers often unfamiliar with the ML fundamentals with means to easily carry out such analysis, could still be misused by providing a false sense of analysis interpretation [31]. Another concern of utilizing user-ready approaches to carry out ML analysis lies in the need for compiling proper observations (i.e. datapoints). In some classical fields (say material sciences, earthquake or fire engineering) where there is a limited number of observations due to expensive tests, or need for specialized instrumentation/facilities [32], then the use of ML may lead to a biased outcome – especially when combined with lack of expertise on ML [33, 34].

An examination of open literature raises a few questions: 1) are we developing accurate ML models? 2) are such models useful to our fields? 3) are we properly validating ML models? And 4) how to confidently answer "yes" to the aforementioned questions?

A distinction should be drawn in which we need to acknowledge that, we often apply existing ML algorithms to our problems rather than developing new algorithms. This acknowledgment goes hand in hand with that similar to applying other numerical tools such as the finite element method, to investigate the response of materials and structures (say concrete beams) under harsh environments (i.e. fire conditions) [35, 36]. From this perspective, we use an existing tool, say a finite element (FE) software (ANSYS [37], ABAQUS [38] etc.), to investigate how failure mechanism occurs in a concrete beam under fire. The accuracy of this FE model is often established through a validation procedure in which a comparison of predictions from the FE model (say temperature rise in steel rebars or mid-span deflection during a fire, or in some cases, point in time when the beam fails) is plotted against that measured in an actual fire test. If the comparison is deemed well, then the FE model is said to be valid and hence can be used to explore the effect of key response parameters (i.e. magnitude of loading, strength of concrete, intensity of fire etc.). From this perspective, the validity of an FE model is established if the variation between predicted results and measured observations is between 5-15%[*] [39].

Unlike the use of FE simulation, ML is often used in two domains: 1) to show the applicability of ML to understand a phenomenon [40, 41], and 2) to identify hidden patterns governing a phenomenon [33, 42]. In the first domain, ML is primarily used to show that an ML algorithm can replicate a phenomenon – or in other words, to validate the applicability of that particular ML algorithm to a material science problem (i.e. can deep learning be applied to predict the compressive strength of concrete given that information regarding the components in a concrete mix is available?). While works in this domain showcase the diversity of ML, these also provide an additional validation platform/case studies to already well-established algorithms. The contribution of such works to our knowledge base is to be thanked and acknowledged.

The second domain is where ML shines and can be proven as a powerful ally to researchers. This is because ML strives on data and is designed to explore hidden features and patterns. The integration of these two items has not been thoroughly applied into our fields and, if applied properly, cannot only open new opportunities but also revolutionize our perspective into our fields. Unfortunately, the open literature continues to lack works in this domain, and hence such works are to be encouraged.

Whether ML is used in the first or second domain, ML models need to be rigorously assessed [43, 44]. This is a critical key to ensure: 1) the validity of the developed ML model in understanding a complex phenomenon given a limited set of data points, and 2) proper extension of the same models towards new/future datasets. Traditionally, the adequacy of ML models is often established through performance fitness and error metrics (PFEMs). Performance and error measures are vital elements in the process of evaluating ML models/frameworks. These are defined as logical and/or mathematical constructs intended to measure the closeness of actual observations to that expected (or predicted). In other words, PFEMs are used to establish an understanding of how predictions from a model compare to real (or measured) observations. Such metrics often relate to the variation between predicted and measured observations in terms of errors [45–47].

[*] One should note that the validation of an FE model is also governed by satisfying convergence criteria input in the FE software. More on this can be found elsewhere [37, 38].

Diverse sets of performance metrics have been noted in the open literature i.e. correlation coefficient (*R*), root mean squared error (RMSE), etc. In practice, one, a multiple, or a combination of metrics are used to examine the adequacy of a particular ML model. However, there does not seem to be a systematic view into which scenarios specific metrics are preferable to use. In order to bridge this knowledge gap, this work compiles the commonly-used PFEMs and highlights their use in evaluating the performance of regression and classification ML models.

2. Performance Fitness and Error Metrics

This section presents the most widely-used PFEMS and highlights fundamentals, recommendations, and limitations associated with their use in assessing ML models[†]. In this work, PFEMs are grouped under two categories; traditional and modern. In this section, these reoccurring terms are used; *A*: actual measurements, *P*: predictions, *n*: number of data points.

2.1 Regression

Regression ML methods deal with predicting a target value using independent variables. Some of these methods include artificial neural networks, genetic programing, etc. PFEMs grouped herein belong to a group of metrics that are based on methods to calculate point distance primarily using subtraction or division operations. These metrics contain fundamental operations, either *A-P* or *P/A,* and can be supplemented with absoluteness or squareness. These are the most widely-used metrics in literature. The simplest form of common PFEMs results from subtracting a predicted value from its corresponding actual/observed value. This is often straightforward, easy to interpret, and most of all yields the magnitude of error (or difference) in the same units as those measured and predicted and can indicate if the model overestimates or underestimates observations (by analyzing the sign of the reminder). One should remember that an issue could arise where due to the opposite between predictions and observations i.e. canceling positive and negative errors. In this scenario, a zero error could be calculated, indicating false accuracy.

This can be avoided by using an absolute error (i.e. |*A-P*|) which only yields non-negative values. Analogous to traditional error, the absolute error also maintains the same units of predictions (and observations), and hence is easily relatable. However, due to its nature, the bias in absolute errors cannot be determined.

Similar to the same concept of absolute error, the squared error also mitigates mutual cancellation of errors. This metric can be continuously differentiable and thus facilitates optimization. However, this metric emphasizes relatively large errors (as opposed to small errors), unlike absolute error, and could be susceptible to outliners. The fact that the units of squared error is squared leads to unconventional units for error (i.e. squared days); which are not intuitive. Other metrics may also include logarithmic quotient error (i.e. *ln(P/A)*) as well as absolute logarithmic

[†] It should be noted that other works have used a different classification for PFEMs [2]. Botchkarev [2] went even further to survey the most preferred metrics reported by researchers during the 1980-2007 era and also explored multiplication and addition point distance methods.

quotient error (i.e. $|ln(P/A)|$). Table 1 lists other commonly used metrics, together with some of their limitations and shortcomings as identified by surveyed studies.

Table 1 List of commonly used PFEMs for ML regression models as collected from open literature

No.	Metric	Definition	Formula	Remarks
1	Error (E)	The amount by which an observation differs from its actual value.	$E = A - P$	• Intuitive • Easy to apply • Works with numeric data
2	Mean error (ME)	The average of all errors in a set.	$ME = \frac{\sum_{i=1}^{n} E_i}{n}$	• May not be helpful in cases where positive and negative predictions cancel each other out. • Works with numeric data
3	Mean Normalized Bias (MNB)	Associated with observation-based minimum threshold.	$MNB = \frac{\sum_{i=1}^{n} E_i/A_i}{n}$	• Biased towards overestimations. • Works with numeric data
4	Mean Percentage Error (MPE)	Computed average of percentage errors.	$MPE = \frac{\sum_{i=1}^{n} E_i/A_i}{n/100}$	• Undefined whenever a single actual value is zero. • Works with numeric data
5	Mean Absolute Error (MAE)*	Measures the difference between two continuous variables.	$MAE = \frac{\sum_{i=1}^{n} \|E_i\|}{n}$	• Uses a similar scale to input data [48]. • Can be used to compare series of different scales. • Works with numeric data
6	Mean Absolute Percentage Error (MAPE)*	Measures the extent of error in percentage terms.	$MAPE = \frac{100}{n} \sum_{i=1}^{n} \|E_i\|/\|A_i\|$	• Commonly-used as a loss function [49] • Cannot be used if there are actual zero values. • Percentage error cannot exceed 1.0 for small predictions. • There is no upper limit to percentage error in predictions that are too high. • Non-symmetrical (adversely affected if a predicted value is larger or smaller than the corresponding actual value) [49]. • Works with numeric data
7	Relative Absolute Error (RAE)	Expressed as a ratio comparing the mean error to errors produced by a trivial model.	$RAE = \sum_{i=1}^{n} \|E_i\|/\|A_i - A_{mean}\|$	• E_i ranges from zero (being ideal) to infinity. • Works with numeric data
8	Mean Absolute Relative Error (MARE)	Measures the average ratio of absolute error to random error.	$MARE = \frac{1}{n} \sum_{i=1}^{n} \|E_i\|/\|A_i\|$	• Sensitive to outliers (especially of low values). • Division by zero may occur (if actuals contain zeros). • Works with numeric data
9	Mean Relative Absolute Error (MRAE)	Ratio of accumulation of errors to cumulative error of random error.	$MRAE = \frac{\sum_{i=1}^{n} \|E_i\|/\|A_i - A_{mean}\|}{n}$	• For a perfect fit, the numerator equals to zero [50]. • Works with numeric data
10	Geometric Mean Absolute Error (GMAE)*	Defined as the n-th root of the product of error values.	$GMAE = \sqrt[n]{\prod_{i=1}^{n} \|E_i\|}$	• GMAE is more appropriate for averaging relative quantities as opposed to arithmetic mean [51]. • This metric can be dominated by large outliers and minor errors (i.e. close to zero). • Works with numeric data
11	Fractional Absolute Error (FAE)	Evaluates the absolute fractional error.	$FAE = \frac{1}{n} \sum_{i=1}^{n} \frac{2 \times \|E_i\|}{\|A_i\| + \|P_i\|}$	• Works with numeric data
12	Mean Squared Error (MSE)	Measures the average of the squares of the errors.	$MSE = \frac{\sum_{i=1}^{n} {E_i}^2}{n}$	• Scale dependent [52]. • Values closer to zero present adequate state • Heavily weights outliers. • Highly dependent on fraction of data used (low reliability) [53]. • Works with numeric data
13	Root Mean Squared Error (RMSE)	Root square of average squared error.	$RMSE = \sqrt{\frac{\sum_{i=1}^{n} {E_i}^2}{n}}$	• Scale dependent. • A lower value for RMSE is favorable. • Sensitive to outliers. • Highly dependent on fraction of data used (low reliability) [53]. • Works with numeric data
14	Sum of Squared Error (SSE)	Sums the squared differences between each observation and its mean.	$SSE = \sum_{i=1}^{n} {E_i}^2$	• A small SSE indicates a tight fit [54]. • Works with numeric data
15	Relative Squared Error (RSE)	Normalizes total squared error by dividing by the total squared error.	$RSE = \sum_{i=1}^{n} {E_i}^2/(A_i - A_{mean})^2$	• A perfect fit is achieved when the numerator equals to zero [50]. • Works with numeric data
16	Root Relative Squared Error (RRSE)	Evaluates the root relative squared error between two vectors.	$RRSE = \sqrt{\sum_{i=1}^{n} {E_i}^2/(A_i - A_{mean})^2}$	• Ranges between zero and 1, with zero being ideal [50]. • Works with numeric data
17	Geometric Root Mean Squared Error (GRMSE)	Evaluates the geometric root squared errors.	$GRMSE = \sqrt[2n]{\prod_{i=1}^{n} {E_i}^2}$	• Scale dependent. • Less sensitive to outliners than RMSE [52]. • Works with numeric data
18	Mean Square Percentage Error (MSPE)*	Evaluates the mean of square percentage errors.	$MSPE = \frac{\sum_{i=1}^{n} (\|E_i\|/\|A_i\|)^2}{n/100}$	• Non-symmetrical [49]. • Works with numeric data
19	Root Mean Square Percentage Error (RMSPE)*	Evaluates the mean of squared errors in percentages.	$RMSPE = \sqrt{\frac{\sum_{i=1}^{n} (\|E_i\|/\|A_i\|)^2}{n/100}}$	• Scale independent. • Can be used to compare predictions from different datasets. • Non-symmetrical [49]. • Works with numeric data

				• An extension of RMSE
20	Normalized Root Mean Squared Error (NRMSE)**	Normalizes the root mean squared error.	$NRMSE = \frac{\sqrt{\frac{\sum_{i=1}^{n} {E_i}^2}{n}}}{A_{mean}}$	• Can be used to compare predictions from different datasets [55]. • Works with numeric data • An extension of RMSE
21	Normalized Mean Squared Error (NMSE)	Estimates the overall deviations between measured values and predictions.	$NMSE = \frac{\frac{\sum_{i=1}^{n} {E_i}^2}{n}}{variance^2}$ $variance = \frac{\sum(x_i - mean)^2}{n-1}$	• Biased towards over-predictions [56]. • Works with numeric data • An extension of MSE
22	Coefficient of Determination (R^2)	The square of correlation.	$R^2 = 1 - \sum_{i=1}^{n}(P_i - A_i)^2 / \sum_{i=1}^{n}(A_i - A_{mean})^2$	• R^2 values close to 1.0 indicate strong correlation. • Can be used in predicting material properties. • Works with numeric data • Related to R
23	Correlation coefficient (R)	Measures the strength of association between variables.	$R = \frac{\sum_{i=1}^{n}(A_i - \overline{A}_i)(P_i - \overline{P}_i)}{\sqrt{\sum_{i=1}^{n}(A_i - \overline{A}_i)^2 \sum_{i=1}^{n}(P_i - \overline{P}_i)^2}}$	• R>0.8 implies strong correlation [57]. • Does not change by equal scaling. • Can be used in predicting material properties. • Works with numeric data
24	Mean Absolute Scaled Error (MASE)	Mean absolute errors divided by the mean absolute error.	$\frac{\sum_{i=1}^{n}\frac{E_i}{A_i}}{n/100} / (\frac{1}{n} - 1)\sum_{i=1}^{n}\lvert A_i - A_{i-1}\rvert$	• Scale independent. • Stable near zero [52]. • Works with numeric data
25	Golbraikh and Tropsha’s [58] criterion		*At least one slope of regression lines (k or k') between the regressions of actual (A_i) against predicted output (P_i) or P_i against A_i through the origin, i.e. $A_i = k \times P_i$ and $P_i = k' A_i$, respectively.* $k = \frac{\sum_{i=1}^{n}(A_i \times P_i)}{{A_i}^2}$ $k' = \frac{\sum_{i=1}^{n}(A_i \times P_i)}{{P_i}^2}$ $m = \frac{R^2 - {R_o}^2}{R^2}$ $n = \frac{R^2 - {R_{o'}}^2}{R^2}$	• *k* and *k'* need to be close to 1 or at least within the range of 0.85 and 1.15. • *m* and *n* are performance indexes and their absolute value should be lower than 0.1. • Works with numeric data
26	QSAR model by Roy and Roy [59]	-	$R_m = R^2 \times (1 - \sqrt{\lvert R^2 - {R_o}^2 \rvert})$ *where,* $- \frac{\sum_{i=1}^{n}(P_i - A_i^o)^2}{\sum_{i=1}^{n}(P_i - P_{mean})^2}, A_i^o = k \times P_i {R'_o}^2$ $= 1 - \frac{\sum_{i=1}^{n}(A_i - P_i^o)^2}{\sum_{i=1}^{n}(A_i - A_{mean})^2}, P_i^o = k' \times A_i$	• R_m is an external predictability indicator. $R_m > 0.5$ implies a good fit. • Works with numeric data
27	Frank and Todeschini [60]	-	*Recommend maintaining a ratio of 3-5 between the number of observations and input parameters.*	-
28	Objective function by Gandomi et al. [61]	A multi-criteria metric.	*Function* $= (\frac{No._{Training} - No._{Validation}}{No._{Training} + No._{Validation}})\frac{RMSE_{Training} + MAE_{Learning}}{R_{Learning} + 1}$ $+ \frac{2No._{Validation}}{No._{Training} + No._{Validation}} \frac{RMSE_{Validation} + MAE_{Validation}}{R_{Training} + 1}$ *where, $No._{Training}$ and $No._{Validation}$ are the number of training and validation data, respectively.*	• This function needs to be minimized to yield highest fitness. • Can be used in predicting material properties. • Works with numeric data
29	Reference index (RI) by Cheng et al. [62]	A multi-criteria metric that uniformly accounts for RMSE, MAE and MAPE.	$RI = \frac{RMSE + MAE + MAPE}{3}$	• Each fitness metric is normalized to achieve the best performance. • Works with numeric data • An extension of RMSE, MAE and MAPE
30	Scatter index (SI) [63]	Applied to examine whether RMSE is good or not.	$SI = \frac{\sqrt{\frac{\sum_{i=1}^{n}(P_{max(A)} - P_{max(p)})^2}{n}}}{P_{max(p)}}$ *where, n = number of data sets used during the training phase. $P_{max(p)}$ = mean actual observations data*	• SI is RMSE normalised to the measured data mean • If SI is less than one, then estimations are acceptable. • Works with numeric data • “excellent performance” when SI < 0.1, a “good performance” when 0.1 < SI < 0.2, a “fair performance” when 0.2 < SI < 0.3, and a “poor performance” when SI > 0.3
31	Synthesis index (SyI) [64]	Comprehensive performance measure a based on MAE, RMSE, and MAPE a	$\mathrm{SyI} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i=1}}^{\mathrm{n}}\left(\frac{\mathrm{P_i} - \mathrm{P_{min,i}}}{\mathrm{P_{max,i}} - \mathrm{P_{min,i}}}\right)$ *where, n = number of performance measures; and P_i = ith performance measure.*	• The SI ranged from 0 to 1; an SI value close to 0 indicated a highly accurate predictive model. • Works with numeric data
32	Relative root mean squared error (RRMSE) [65]	Present percentage variation in accuracy	$RRMSE = \sqrt{\frac{1}{n}\sum(A - P)^2}$	• Lower RRMSE values result in more accurate model predictions. • Works with numeric data
33	Performance index (PI) [65]	Performance index to evaluate predictivity of a model	$PI = \frac{RRMSE}{1 + R}$	• Lower PI values result in more accurate model predictions. • Works with numeric data

34	$a_{20\text{-index}}$ [66]	Performance index to evaluate predictivity of a model within 20% variation	$a_{20-index} = \frac{m_{20}}{M}$ *where, m_{20} is the number of samples with the ratio of experimental value over predicted value falling from 0.8 to 1.2 and M is the number of samples in the dataset.*	• Presents the number of samples with the difference between the predicted value and experimental value within ±20% • Works with numeric data
35	Fractional bias (FB) [67]	Measure of the shift between the observed and predicted values.	$FB = \frac{2\sum_{i=1}^{n}(\mathrm{A}-\mathrm{P})}{\sum_{i=1}^{n}(\mathrm{A}+\mathrm{P})}$	• Dimensionless metric, which is convenient for comparing the results from studies involving different scales • Symmetrical and bounded; values for the fractional bias range between -2.0 (extreme underprediction) to +2.0 (extreme overprediction) • Perfect model has FB of zero. • Works with numeric data
36	Relative index of agreement (RD) [68]	A standardized measure of the degree of model prediction error	$\mathrm{RD} = 1 - \frac{\sum_{i=1}^{N}(\frac{A-\mathrm{P}}{\mathrm{A}})}{\sum_{i=1}^{N}(\frac{\lvert \mathrm{P}-\bar{\mathrm{A}}\rvert + \lvert A-\bar{\mathrm{A}}\rvert}{\bar{\mathrm{A}}})^2}$	• A value of 1.0 indicates a perfect match, and zero indicates no agreement at all. • Overly sensitive to extreme values • Works with numeric data
37	Nash–Sutcliffe coefficient (NSE) [69]	A metric often used in flow predictions.	$\mathrm{NSE} = 1 - [\frac{\sum_{i=1}^{N}(A-P)^2}{\sum_{i=1}^{N}(\mathrm{A}-\bar{\mathrm{A}})^2}]$	• NSE = 1 indicates perfect correspondence • NSE = 0 indicates that the model simulations have the same explanatory power as the mean of the observations • NSE < 0 indicates that the model is a worse predictor than the mean of the observations • Works with numeric data
38	Kling–Gupta efficiency (KGE) [70]	A metric often used in flow predictions.	$\mathrm{KGE} = 1 - \sqrt{(r-1)^2 + (\alpha-1)^2 + (\beta-1)^2}$, where, r is the linear correlation between the predicted and actuals. α is the magnitude of the variability calculated as the standard deviation in predictions divided by the standard deviation in actuals. β is the bias term calculated as the predictions means divided by the actual mean. N is the number of dataset over the training and testing phases.	• KGE = 1 indicates perfect agreement between actuals and predictions. • KGE < 0 indicates that the mean of actuals provides better estimate than predictions • For other values of KGE, please refer to [71] • Works with numeric data

*has a median derivative

**can be normalized by standard deviation of actual observations

***The reader is encouraged to review the cited references for full details on specific metrics.

Most of the works conducted so far in the areas of engineering applications only utilized a few of the above PFEMs [20, 33, 61, 62, 72–92]. The bulk of the reviewed works continue to incorporate traditional metrics such as *R*, R^2, *MAE, MAPE,* and *RMSE* as primary indicators of adequacy of the regression-based ML models. This seems to stem from our familiarity with these indicators, as opposed to others; such as Golbraikh and Tropsha's [58] criterion, QSAR model by Roy and Roy [59], Frank and Todeschini [60], and specifically designed objective functions, often used in the realms of other fields and data sciences. It should be noted that out of the reviewed studies, the works of Gandomi et al. [90], Golafshani and Behnood [40] as well as Cheng et al. [62] applied a multi-criteria verification process that incorporated the use of traditional as well as modern PFEMs. Utilizing multi-criteria is not only beneficial to ensure the validity of a particular ML model but is also recommended to overcome some of the identified limitations of traditional metrics in Table 1 and hence should be encouraged.

2.2 Classification

In ML, classification refers to categorizing data into distinct classes. This is a supervised learning approach where machines learn to classify observations into binary or multi-classes. Binary classes are those with two labels (i.e. positive vs. negative etc.), and multi-classes are those having more than two labels (i.e. types of concrete e.g., normal strength, high strength, high performance etc.). Classification algorithms may include logistic regression, k-nearest neighbors, support vector machines, etc. [93, 94].

The performance of classifiers is often listed in a confusion matrix. This matrix contains statistics about actual and predicted classifications and lays the fundamental foundations necessary to understand accuracy measurements for a specific classifier. Each column in this matrix signifies predicted instances, while each row represents actual instances. This matrix was identified to be the "go-to" metric used in studies examining materials science and engineering problems [22, 95–98]. However, there are other PFEMs that can be used to evaluate classification models, and these, along with others, are listed in Table 2. Similar to Table 1, Table 2 also lists some of the remarks and limitations pointed out by surveyed works. In this table, *P (denotes number of real positives), N (denotes number of real negatives), TP (denotes true positives), TN (denotes true negatives), FP (denotes false positives), and FN (denotes false negatives)*.

Table 2 List of the commonly-used PFEMs for ML classification models as collected from open literature

No.	Metric	Definition	Formula	Remarks
1	True Positive Rate (TPR) or Sensitivity or Recall	Measures the proportion of actual positives that are correctly identified as positives.	$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$	• Describes the proportion of actual positives that are correctly identified. • Does not account for indeterminate results. • Works with categorial data
2	True Negative Rate (TNR) or Specificity or selectivity	Measures the proportion of actual negatives that are correctly identified negatives.	$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$	• Describes the proportion of actual negatives that are correctly identified. • Works with categorial data
3	Positive Predictive Value (PPV) or Precision	The proportions of positive observations that are true positives.	$PPV = \frac{TP}{TP + FP} = 1 - FDR$	• Has an ideal value of 1 and the worst value of zero. • Works with categorial data
4	Negative Predictive Value (NPV)	The proportions of negative observations that are true positives.	$NPV = \frac{TN}{TN + FN} = 1 - FOR$	• Has an ideal value of 1 and the worst value of zero. • Works with categorial data
5	False Positive Rate (FPR)	Measures the proportion of positive cases in that are correctly identified as positives.	$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR$	• Describes proportion of negative cases incorrectly identified as positive cases. • Works with categorial data
6	False Discovery Rate (FDR)	Expected proportion of false observations.	$FDR = \frac{FP}{FP + TP} = 1 - PPV$	• Describes proportion of the individuals with a positive test result for which the true condition is negative. • Works with categorial data
7	False Omission Rate (FOR)	Measures the proportion of false negatives that are incorrectly rejected.	$FDR = \frac{FN}{FN + TPN} = 1 - NPV$	• Describes proportion of the individuals with a negative test result for which the true condition is positive. • Works with categorial data
8	Positive likelihood ratio (LR+)	Evaluates the change in the odds of having a diagnosis with a positive test.	$LR{+}{=}\frac{TPR}{FPR}$	• Measures the ratio of TPR (sensitivity) to the FPR (1 – specificity). • Presents the likelihood ratio for increasing certainty about a positive diagnosis. • Works with categorial data
9	Negative likelihood ratio (LR-)	Evaluates the change in the odds of having a diagnosis with a negative test.	$LR{-}{=}\frac{FNR}{TNR}$	• Describes the ratio of FNR to TNR (specificity). • Works with categorial data
10	Diagnostic odds ratio (DOR)	Measures the effectiveness of a (diagnostic) test.	$DOR = \frac{LR+}{LR-} = \frac{TP/FP}{FN/TN}$	• Often used in binary classification. • Works with categorial data
11	Accuracy (ACC)	Evaluates the ratio of number of correct predictions to the total number of samples.	$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$	• Presents performance at a single class threshold only. • Assumes equal cost for errors [96]. • Works with categorial data
12	F_1 score	Harmonic mean of the precision and recall.	$F_1 = \frac{2PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	• Describes the harmonic mean of precision and sensitivity. • Focuses on one class only. • Biased to the majority class [99]. • Works with categorial data
13	Matthews Correlation Coefficient (MCC)	Measures the quality of binary classifications analysis.	$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + PN)}}$	• Measures the quality of binary and multi-class classifications. • Can be used in classes with different sizes. • When MCC equals +1 → perfect prediction, → 0 equivalent to a random prediction and → −1 false prediction. • Considered as a balanced measures as it involves values of all the four quardants of a confusion matrix [100]. • Works with categorial data
14	Bookmaker Informedness (BM) or Youden's J statistic	Evaluates the discriminative power of the test [101].	$BM = TPR + TNR - 1$	• Describes the probability of an informed decision (vs. a random guess). • Has a range between zero and 1 (being ideal). • Considers both real positives and real negatives. • Takes into account all predictions [102]. • Works with categorial data • Counterpart of recall. • It is also suitable with imbalanced data. • It does not change concerning the differences between the sensitivity and specificity [101].
15	Markedness (MK)	Measures trustworthiness of positive and	$MK = PPV + NPV - 1$	• Measures trustworthiness of positive and negative predictions by a model [103].

		negative predictions.		• Considers both predicted positives and predicted negatives. • Counterpart of precision. • Specifies the probability that a condition is marked by the predictor (as opposed to luck/chance) [104] • Sensitive to data changes (not suitable for imbalanced data) [101]. • Works with categorial data
16	Average Class Accuracy (ACA)	Measures the average accuracy of predictions in a class.	$ACA = W\left(\frac{TP}{TP+FP}\right) + (1-W)\left(\frac{TN}{TN+FP}\right)$ $where\ \ 0\ < W\ < 1$	• Used with unbalanced data. • Choosing a good weighting factor a priori [99]. • When $W > 0.5$, minority class accuracy contributes more than majority class. • Presents performance at a single class threshold. • Works with categorial data
17	Receiver Operating Characteristic (ROC)	Plots the diagnostic ability of a binary classifier system as its discrimination threshold is varied.	*The ROC curve is plotted such that TPR is on vertical axis and FPR is on the horizontal axis (the line TPR = FPR represents a random guess of a specific class)* [105].	• Characterizes tradeoff between hit rate and false alarm rate. • Designates the relationship between sensitivity and specificity [106]. • Takes a value between zero and 1 to relate the probability distribution to a single state [107]. • A threshold of zero ensures highest sensitivity and 1 ensures best specificity. • Can be used to estimate cost ratio (slope of line tangent to ROC curve). • Should be used in datasets with roughly equal numbers of observations for each class [108, 109]. • Works with categorial data
18	Area under the ROC curve (AUC)	Measures the two-dimensional area underneath the entire ROC curve.	$AUC = \sum_{i=1}^{N-1} \frac{1}{2}(FP_{i+1} - FP_i)\,(TP_{i+1} - TP_i)$ *or* $AUC = \frac{1}{2} w\ (h + h'),$ *where, w = width, and h and h' = heights of the sides of a trapezoid histogram*	• Not dependent on a single class threshold. • Associated with increased training times. • Works with categorial data
19	Precision-Recall curve	Plots the tradeoff between precision and recall for different thresholds.	*Plots precision (in the vertical axis) and the recall (in the horizontal axis) for different thresholds.*	• Applicable in cases of moderate to large class imbalance [108]. • Used in binary classification.
20	Log Loss Error (LLE)	Measures the where the prediction input is a probability value.	$LLE = -\sum_{c=1}^{M} A_i logP,$ *where, M: number of classes, c: class label, y: binary indicator (0 or 1) if c is the correct classification for a given observation.*	• Measures the uncertainty of the probabilities by comparing predictions to the true labels. • Penalizes for being too confident in wrong prediction. • Has probability between zero and 1. • A log loss of zero indicates a perfect model. • Works with categorial data
21	Hinge Loss Error (HLE)	-	$HLE = max(0{,}1 - q \cdot y)$ *where, q= ±1 and y: classifier score*	• Linearly penalize incorrect predictions. • Primarily used in support vector machine.
22	Wilcoxon–Mann–Whitney (WMW) test [99]	-	$WMW = \frac{\sum_{i \in Minor\ class} \sum_{i \in Major\ class} I_{wmw}(P_i, P_j)}{\lvert Minor\ class \rvert \times \lvert Major\ class \rvert},$ *where, P_i and P_j: outputs when evaluated on an example from the minority and majority classes, respectively*	• Used in scenarios with unbalanced data. • The indicator function I_{wmw} returns 1 if $P_i > P_j$ and $P_i \geq 0$ or 0 if otherwise.
23	Fitness Function *Amse* (FFA) [99]	Measures pattern difference between input and output.	$FFA = \frac{1}{K}\sum_{c=1}^{K}\left(1 - \frac{\sum_{i=1}^{N_c}(1 - sig(P_{ci}) - T_c)}{N_c \times 2}\right)^2,$ $sig(x) = \frac{2}{1+e^{-x}} + 1$ *where, P_{ci}: output of a classifier evaluated on the ith example, N_c: number of examples, K: number of classes, T_c : target values (equals to -0.5 and 0.5 for majority and minority classes, respectively)*	• Used in scenarios with unbalanced data. • Appropriate for genetic programing. • Needs to be scaled to a range of [-1, 1] and hence the need for sigmoid function. • FFA = 1 presents an ideal scenario.
24	Fitness Function *Incr* (FFI) [99]	-	$Incr = \frac{1}{K}\sum_{c=1}^{K}\left(\frac{\sum_{j=1}^{M_c}\left[I_{zt}(j, D_{cj}, c).\sum_{i=1}^{N_c} Eq(D_{cj}, P_{ci})\right]}{\frac{1}{2}N_c(N_c+1)}\right)$ $I_{zt}(r,k,c) = \begin{cases} r, & if\ k \geq 0\ and\ c \in Minority\ class \\ & or\ if\ k < 0\ and\ c \in Majority\ class \\ 0, & otherwise \end{cases}$ $Eq(p,q) = \begin{cases} 1, & if\ p = q \\ 0, otherwise & \end{cases}$	• Used in scenarios with unbalanced data. • Assigns incremental rewards to predictions that fall further away from the class boundary. • Appropriate for genetic programming. • Ranges [0, 1] (zero being worst fitness).
25	Fitness Function Correlation (FFC)	-	$FFC = \frac{1}{K}\left(r + I_{zt}(1, \mu_{minor}, \mu_{major}\right),$ $r = \sqrt{\frac{\sum_{c=1}^{K} N_c(\mu_c - \bar{\mu})^2}{\sum_{c=1}^{K}\sum_{i=1}^{N_c}(P_{ci} - \bar{\mu})^2}}$ $\mu_c = \frac{\sum_{i=1}^{N_c} P_{ci}}{N_c},\ \bar{\mu} = \frac{\sum_{c=1}^{K} N_c\mu_c}{\sum_{c=1}^{K} N_c}.$ *where, r: correlation ratio, μ_{minor} and μ_{major}: mean for minor and major classes, respectively*	• Used in scenarios with unbalanced data.

26	Fitness Function Distribution (FFD)	Measures the distance between class distributions as a function of class separability.	$FFD = \frac{\lvert\mu_{min} - \mu_{maj}\rvert}{\sigma_{min} + \sigma_{maj}} \times I_{zt}(2, \mu_{min}, \mu_{maj})$ $\mu_c = \frac{\sum_{i=1}^{N_c} P_{ci}}{N_c}, \ \sigma_c = \sqrt{\frac{1}{N_c}\sum_{i=1}^{N_c}(P_{ci} - \mu_c)^2}.$ *where, μ_c and σ_c: mean and standard deviation of the class distribution, respectively,*	• Used in scenarios with unbalanced data. • Treats predictions as independent distributions. • Measures separability (i.e. distance between class distributions) [110] – high separability (no overlap) and this distance turns large (go to +∞). • Uses I_{zt} to enforce zero class threshold.
27	Canberra Metric (CM)	Measures the distance between pairs of points in a vector space.	$CM = \sum_{i=1}^{n} \frac{\lvert E_i\rvert}{A_i + P_i}$	-
28	Wave Hedges Distance (WHD)	-	$WHD = \sum_{i=1}^{n} \frac{\lvert E_i\rvert}{max\ (A_i, P_i)}$	• Normalizes the difference of each pair of coefficients with its maximum [111–113].
29	Lift [114]	Measures the performance of a model at predicting or classifying cases.	$LIFT \ = \frac{\%of\ true\ positives\ above\ the\ threshold}{\%of\ dataset\ above\ the\ threshold}$	• Measures betterness of a classifier than a baseline classifier that randomly predicts positives. • Threshold is set as a static fraction of the positive dataset. • Lift and Accuracy do not always correlate well.
30	Mean Cross Entropy (MXE)	Measures the performance of a model where the output is a probability between zero and one.	$MXE = -\frac{1}{N}\sum True \times ln(Predicted) + (1 - True) \times ln(1 - Predicted)$ *(The assumptions are that Predicted ∈ [0, 1] and True ∈ {0, 1})*	• Minimizing MXE gives the maximum likelihood [102].
31	Probability Calibration (CAL)	-	*1. Order cases 1-100 by their predicted in the same bin.* *2. Evaluate the percentage of true positives.* *3. Calculate the mean prediction for true positives.* *4. Calculate the mean prediction calibration error for this bin (using the absolute value of the difference between the observed frequency and the mean).* *5. Repeat steps 1-4 for cases 2-101, 3-102, etc.* *6. CAL is calculated as the mean of these binned calibration errors* [102].	• Lengthy procedure.
32	Precision-recall break-even point	Point at which the precision-recall-curve intersects the bisecting line.	*Precision = Recall*	• Defines the point when precision and recall are equal.
33	Average precision (AP)	Combines recall and precision for ranking.	$AP = \sum_n (Recall_n - Recall_{n-1})Percision_n$	• Describes the weighted mean of precision in each threshold with the increase in recall from the previous threshold used.
34	Balanced accuracy [115]	Calculates the average of the correctly identified proportion of individual classes.	*Defined as the average of recall obtained on each class.*	• Used in binary and multiclass classification problems. • Accommodates imbalanced datasets.
35	Brier score (BS)	Measures the accuracy of probabilistic-based predictions.	$BS = \frac{1}{N}\sum_{i=1}^{N}(f_i - A_i)^2$ *in which f_i is the probability that was forecast, A_i the actual outcome of the event at instance i*	• Measures the mean squared difference between the predicted probability and the actual outcome. • Takes on a value between zero and 1 (the lower the score is, the better the predictions). • Composed of refinement loss and calibration loss. • Appropriate for binary and categorical outcomes. • Inappropriate for ordinal variables.
36	Cohen's kappa (CK) [116]	Measures interrater (agreement) reliability.	$\kappa = (p_o - p_e)/(1 - p_e)$ *where, p_o: empirical probability of agreement on the label assigned to any sample, p_e: expected agreement when both annotators assign labels randomly and this is estimated using a per-annotator empirical prior over the class labels.*	• Measures inter-annotator agreement. • Expresses the level of agreement between two annotators [117]. • Ranges between -1 and 1. The maximum value means complete agreement.
37	Hamming loss (HL)	Fraction of the wrongly identified labels.	$HL = \frac{1}{m}\sum_{i=1}^{m} 1_{\widehat{P_\imath \neq A_\imath}}$	• Describes fraction of labels that are incorrectly predicted. • Optimal value is zero [118].
38	Fitness (T) [119]	-	$Fitness(T) = Q(T) + \alpha * R(T) + \beta * Cost(T)$ *where, Q(T): accuracy, R(T): sum of R(T_i) in all multi-tests of the T tree, Cost(T): sum of the costs of attributes constituting multi-tests. The default parameters values are: α=1.0 and β=−0.5,* $R(T_i) = \frac{\lvert X_i\rvert}{\lvert X\rvert} * \sum_{j=1}^{\lvert mt_i\rvert - 1} r_{ij}$ *where, X: learning set, X_i: instances in i-th node, and $\lvert mt_i\rvert$: size of a multi-test.* $Cost(T_i) = \frac{\lvert X\rvert}{\lvert X_i\rvert} * C(a_{ij})$ *where: a_{ij}: j-th attribute of the i-th multi-test, C(a_{ij}): cost of the a_{ij} attribute.*	• Used for fitting decision trees. • This function needs to be maximized to achieve high performance.
39	F2 score [120]	Measured as the weighted average	$F_\beta = 1 + \beta 2 \times \frac{precision \times recall}{(\beta 2 \times precision) + recall}$ *where: β = 2.*	• Used in genetic programming and medical fields. • Computes a weighted harmonic mean of Precision and Recall.

		of precision and recall.		• Learning about the minority class.
40	Distance score (D score) [120]	-	$D_{sc} = \frac{2 \times C1 \times C2}{C1 + C2}$ *where:* $C1 = \frac{\sum_{i=0}^{N_{maj}} sig(P_{Maji}) \times \lvert T - sig(P_{Maji})\rvert}{N_{maj}} \times func(1, P_{Maji})$ $sig(x) = \frac{2}{1 + e - x} - 1$ $C2 = \frac{\sum_{i=0}^{N_{min}} sig(P_{Mini}) \times \lvert T - sig(P_{Mini})\rvert}{N_{min}} \times func(1, P_{Mini})$ $func(1, k) = \begin{cases} 1, & \text{if} k \leq 0 \text{formajorityclassinstance} \\ 1, & \text{if} k > 0 \text{forminorityclassinstance} \\ 0, & \text{otherwise} \end{cases}$ *C1* for majority class and *C2* for minority class.	• Used in genetic programming and medical fields. • Distance score (D score) which learns about both the classes by giving them equal importance and being unbiased. • The range of both C1 and C2 is 0 (worst score) to 1 (best score).

[*]The reader is encouraged to review the cited references for full details on specific metrics.

3. Closing Remarks

Our confidence in the accuracy of predictions obtained from ML algorithms heavily relies on the availability of actual observations and proper PFEMs. From this point of view, it is unfortunate that observations relating to the engineering discipline continue to be 1) limited in size, and 2) lack completeness. The lack of such observations is often related to limitations in conducting full-scale tests, the need for specialized equipment, and a wide variety of tested samples. For instance, one can think of how normal strength concrete mixes can significantly vary from one study to another simply due to variation in raw materials, mix proportions, and casting/curing procedures, etc.

Combining the above two points with the notion of simply "applying ML" to understand a given phenomenon (say flexural strength of beams) without a thorough validation is deemed to fail. In fact, in many instances, researchers noted the validity of a specific ML model by reporting its performance against traditional PFEMs, only to be later identified that such a model does not properly represent actual observations – despite having good fitness. This can be avoided by adopting a rigorous validation procedure [121, 122]. Unfortunately, many of the published studies in the area of ML application in engineering do not include multi-criteria/additional validation phases and simply rely on conventional performance metrics such as R or R^2 of the derived models. Furthermore, adopting a set of PFEMs does not negate the occurrence of some common issues, most notably, overfitting, biasedness etc. As such, an analysis that utilizes ML should also consider some of the following techniques e.g. use of independent test datasets, varying degrees of cross-validation etc.

In order to ensure fruitful use of ML, it is our duty to seek proper application of ML. Besides, one of the major concerns about the ML-based models is their robustness under a wide range of conditions [123]. A robust ML model should not only provide reasonable PFEMs but should also be capable of capturing the underlying physical mechanisms that govern the investigated system [124]. An essential approach to verify the robustness of the ML models is to perform parametric and sensitivity analyses [123, 125]. These types of analyses ensure that the ML predictions are in sound agreement with the system's real behavior and physical processes rather than being merely a combination of the variables with the best fit on the data. Another item to consider is to develop a user-friendly phenomenon-specific recommendation system wherein novice users who apply pre-identified PFEMs are selected to evaluate the performance of a given problem (say using R^2 in a regression problem etc.).

The reader is to remember that the addition of one example to showcase recommended or important PFEMs negates the purpose of this paper (which is to compile commonly used performance metrics and list their key characteristics into one document to provide interested researchers in carrying out a ML analysis with a starting point to select proper performance metrics). Providing a comparison for all of the reviewed metrics will significantly extend this work beyond its scope and may not be feasible at the moment. We feel that this is best suited for a series of more in-depth reviews wherein metrics for classification and regression problems can be separately evaluated and reviewed under well-designed problems and a variety of conditions to ensure fairness and unbiasedness to come in the near future.

It is our intention to not specifically identify a measure (or a set of measures) due to the wide range of problems (as well as the quality of data) that a scientist could face. Please note that other researchers (which are quoted herein) also followed a similar approach.

- *"Although some methods clearly perform better or worse than other methods on average, there is significant variability across the problems and metrics. Even the best models sometimes perform poorly, and models with poor average performance occasionally perform exceptionally well."* [126].
- *"It is clearly difficult to convincingly differentiate ML algorithms (and feature reduction techniques) on the basis of their achievable accuracy, recall and precision."*[127].
- *"Different performance metrics yield different tradeoffs that are appropriate in different settings. No one metric does it all, and the metric optimized to or used for model selection does matter."*[102].

4. Conclusions

Based on the information presented in this note, the following conclusions can be drawn.

- ML is expected to rise into a key analysis tool in the coming few years; especially within material scientists and structural engineers. As such, the integration of ML is to be thorough and proper. Hence, the need for proper validation procedure.
- A variety of performance metrics and error metrics exists for regression and classification problems. This work recommends the utilization of multi-fitness criteria (where a series of metrics are checked on one problem) to ensure the validity of ML models as these metrics may overcome some of the limitations of induvial metrics. Such metrics can be of independent nature to each other such as, R^2, RSME, and $a_{20-\text{index}}$.
- The performance of the existing metrics and future fitness functions can be further improved through systematic collaboration between researchers of interdisciplinary backgrounds. For example, efforts are invited to identify and recommend metrics suitable for specific problems and datasets.
- Future works should be directed towards documenting and exploring performance metrics for other types of learnings such as unsupervised learning and reinforcement learning. This is ongoing research need that is to be addressed in the coming years.

Data Availability

No data, models, or code were generated or used during the study.

<u>*Declarations of interest*</u>: none.

5. References

1. Mahdavi S, Rahnamayan S, Deb K (2018) Opposition based learning: A literature review.

Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2017.09.010

2. Botchkarev A (2019) A new typology design of performance metrics to measure errors in machine learning regression algorithms. Interdiscip J Information, Knowledge, Manag 14:045–076. https://doi.org/10.28945/4184
3. Bishop C (2007) Pattern Recognition and Machine Learning. Technometrics. https://doi.org/10.1198/tech.2007.s518
4. Fu G-S, Levin-Schwartz Y, Lin Q-H, Zhang D (2019) Machine Learning for Medical Imaging. J Healthc Eng. https://doi.org/10.1155/2019/9874591
5. Michalski, R. S., Carbonell, J. G., & Mitchell TM (1983) Machine learning: An artificial intelligence approach.
6. Majidifard H, Jahangiri B, Buttlar WG, Alavi AH (2019) New machine learning-based prediction models for fracture energy of asphalt mixtures. Meas J Int Meas Confed. https://doi.org/10.1016/j.measurement.2018.11.081
7. Hu X, Li SE, Yang Y (2016) Advanced Machine Learning Approach for Lithium-Ion Battery State Estimation in Electric Vehicles. IEEE Trans Transp Electrif. https://doi.org/10.1109/TTE.2015.2512237
8. Voyant C, Notton G, Kalogirou S, et al (2017) Machine learning methods for solar radiation forecasting: A review. Renew. Energy
9. Shukla R, Singh D (2017) Experimentation investigation of abrasive water jet machining parameters using Taguchi and Evolutionary optimization techniques. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2016.07.002
10. Hindman M (2015) Building Better Models: Prediction, Replication, and Machine Learning in the Social Sciences. Ann Am Acad Pol Soc Sci. https://doi.org/10.1177/0002716215570279
11. Grimmer J (2014) We are all social scientists now: How big data, machine learning, and causal inference work together. In: PS - Political Science and Politics
12. Naser M, Chehab A (2018) Materials and design concepts for space-resilient structures. Prog Aerosp Sci 98:74–90. https://doi.org/10.1016/j.paerosci.2018.03.004
13. Rashno A, Nazari B, Sadri S, Saraee M (2017) Effective pixel classification of Mars images based on ant colony optimization feature selection and extreme learning machine. Neurocomputing. https://doi.org/10.1016/j.neucom.2016.11.030
14. Jordan MI, Mitchell TM (2015) Machine learning: Trends, perspectives, and prospects. Science 349:255–60. https://doi.org/10.1126/science.aaa8415
15. Seitllari A (2014) Traffic Flow Simulation by Neuro-Fuzzy Approach. In: Second International Conference on Traffic. Belgrade, pp 97–102
16. Naser MZ (2019) AI-based cognitive framework for evaluating response of concrete structures in extreme conditions. Eng Appl Artif Intell 81:437–449. https://doi.org/10.1016/J.ENGAPPAI.2019.03.004
17. Li X, Qiao T, Pang Y, et al (2018) A new machine vision real-time detection system for liquid impurities based on dynamic morphological characteristic analysis and machine learning. Meas J Int Meas Confed. https://doi.org/10.1016/j.measurement.2018.04.015
18. Oleaga I, Pardo C, Zulaika JJ, Bustillo A (2018) A machine-learning based solution for

chatter prediction in heavy-duty milling machines. Meas J Int Meas Confed. https://doi.org/10.1016/j.measurement.2018.06.028

19. Shanmugamani R, Sadique M, Ramamoorthy B (2015) Detection and classification of surface defects of gun barrels using computer vision and machine learning. Meas J Int Meas Confed. https://doi.org/10.1016/j.measurement.2014.10.009
20. Naser MZ (2019) Properties and material models for common construction materials at elevated temperatures. Constr Build Mater 10:192–206. https://doi.org/10.1016/j.conbuildmat.2019.04.182
21. Raccuglia P, Elbert KC, Adler PDF, et al (2016) Machine-learning-assisted materials discovery using failed experiments. Nature. https://doi.org/10.1038/nature17439
22. Alavi AH, Hasni H, Lajnef N, et al (2016) Damage detection using self-powered wireless sensor data: An evolutionary approach. Meas J Int Meas Confed. https://doi.org/10.1016/j.measurement.2015.12.020
23. Farrar CR, Worden K (2012) Structural Health Monitoring: A Machine Learning Perspective
24. Mcfarlane C (2011) The city as a machine for learning. Trans Inst Br Geogr. https://doi.org/10.1111/j.1475-5661.2011.00430.x
25. Chan J, Chan K, Yeh A (2001) Detecting the nature of change in an urban environment: A comparison of machine learning algorithms. Photogramm. Eng. Remote Sensing
26. King DE (2009) Dlibml: A Machine Learning Toolkit. J Mach Learn Res
27. Collobert R, Kavukcuoglu K, Farabet C (2011) Torch7: A Matlab-like Environment for Machine Learning
28. Hall M, Frank E, Holmes G, et al (2009) The WEKA data mining software. ACM SIGKDD Explor Newsl. https://doi.org/10.1145/1656274.1656278
29. Ramsundar B (2016) TensorFlow Tutorial. CS224d
30. Zaharia M, Franklin MJ, Ghodsi A, et al (2016) Apache Spark. Commun ACM. https://doi.org/10.1145/2934664
31. Korolov M (2018) AI’s biggest risk factor: Data gone wrong | CIO. In: CIO. https://www.cio.com/article/3254693/ais-biggest-risk-factor-data-gone-wrong.html. Accessed 5 Jul 2019
32. Kodur VKR, Garlock M, Iwankiw N (2012) Structures in Fire: State-of-the-Art, Research and Training Needs. Fire Technol 48:825–39. https://doi.org/10.1007/s10694-011-0247-4
33. Naser MZ (2019) Fire Resistance Evaluation through Artificial Intelligence - A Case for Timber Structures. Fire Saf J 105:1–18. https://doi.org/https://doi.org/10.1016/j.firesaf.2019.02.002
34. Domingos P (2012) A few useful things to know about machine learning. Commun ACM. https://doi.org/10.1145/2347736.2347755
35. Shakya AM, Kodur VKR (2015) Response of precast prestressed concrete hollowcore slabs under fire conditions. Eng Struct. https://doi.org/10.1016/j.engstruct.2015.01.018
36. Kodur VKR, Bhatt PP (2018) A numerical approach for modeling response of fiber reinforced polymer strengthened concrete slabs exposed to fire. Compos Struct 187:226–240. https://doi.org/10.1016/J.COMPSTRUCT.2017.12.051

37. Kohnke PC (2013) ANSYS. In: © ANSYS, Inc.
38. Abaqus 6.13 (2013) Abaqus 6.13. Anal User's Guid Dassault Syst
39. Franssen JM, Gernay T (2017) Modeling structures in fire with SAFIR®: Theoretical background and capabilities. J Struct Fire Eng. https://doi.org/10.1108/JSFE-07-2016-0010
40. Golafshani EM, Behnood A (2018) Automatic regression methods for formulation of elastic modulus of recycled aggregate concrete. Appl Soft Comput J. https://doi.org/10.1016/j.asoc.2017.12.030
41. Sadowski Ł, Nikoo M, Nikoo M (2018) Concrete compressive strength prediction using the imperialist competitive algorithm. Comput Concr. https://doi.org/10.12989/cac.2018.22.4.355
42. Alavi AH, Gandomi AH, Sahab MG, Gandomi M (2010) Multi expression programming: A new approach to formulation of soil classification. Eng Comput 26:111–118. https://doi.org/10.1007/s00366-009-0140-7
43. Mirjalili S, Lewis A (2015) Novel performance metrics for robust multi-objective optimization algorithms. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2014.10.005
44. Mishra SK, Panda G, Majhi R (2014) A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2014.01.001
45. Schmidt MD, Lipson H (2010) Age-fitness pareto optimization
46. Cremonesi P, Koren Y, Turrin R (2010) Performance of Recommender Algorithms on Top-N Recommendation Tasks Categories and Subject Descriptors. RecSys
47. Laszczyk M, Myszkowski PB (2019) Survey of quality measures for multi-objective optimization: Construction of complementary set of multi-objective quality measures. Swarm Evol Comput 48:109–133. https://doi.org/10.1016/J.SWEVO.2019.04.001
48. Willmott CJ, Matsuura K (2005) Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Clim Res. https://doi.org/10.3354/cr030079
49. Makridakis S (1993) Accuracy measures: theoretical and practical concerns. Int J Forecast. https://doi.org/10.1016/0169-2070(93)90079-3
50. Ferreira C (2001) Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. Ferreira, C (2001) Gene Expr Program a New Adapt Algorithm Solving Probl Complex Syst 13
51. (2016) Handbook of Time Series Analysis, Signal Processing, and Dynamics
52. Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. Int J Forecast. https://doi.org/10.1016/j.ijforecast.2006.03.001
53. Shcherbakov MV, Brebels A, Shcherbakova NL, et al (2013) A survey of forecast error measures. World Appl Sci J. https://doi.org/10.5829/idosi.wasj.2013.24.itmies.80032
54. Bain LJ (1967) Applied Regression Analysis. Technometrics. https://doi.org/10.1080/00401706.1967.10490452
55. Armstrong JS, Collopy F (1992) Error measures for generalizing about forecasting

methods: Empirical comparisons. Int J Forecast. https://doi.org/10.1016/0169-2070(92)90008-W

56. Poli AA, Cirillo MC (1993) On the use of the normalized mean square error in evaluating dispersion model performance. Atmos Environ Part A, Gen Top. https://doi.org/10.1016/0960-1686(93)90410-Z
57. Smith G (1986) Probability and statistics in civil engineering. Collins, London
58. Golbraikh A, Shen M, Xiao Z, et al (2003) Rational selection of training and test sets for the development of validated QSAR models. J Comput Aided Mol Des 17:241–253. https://doi.org/10.1023/A:1025386326946
59. Roy PP, Roy K (2008) On some aspects of variable selection for partial least squares regression models. QSAR Comb Sci 27:302–313. https://doi.org/10.1002/qsar.200710043
60. Frank I, Todeschini R (1994) The data analysis handbook
61. Gandomi AH, Yun GJ, Alavi AH (2013) An evolutionary approach for modeling of shear strength of RC deep beams. Mater Struct Constr. https://doi.org/10.1617/s11527-013-0039-z
62. Cheng MY, Firdausi PM, Prayogo D (2014) High-performance concrete compressive strength prediction using Genetic Weighted Pyramid Operation Tree (GWPOT). Eng Appl Artif Intell. https://doi.org/10.1016/j.engappai.2013.11.014
63. Alwanas AAH, Al-Musawi AA, Salih SQ, et al (2019) Load-carrying capacity and mode failure simulation of beam-column joint connection: Application of self-tuning machine learning model. Eng Struct. https://doi.org/10.1016/j.engstruct.2019.05.048
64. Chou JS, Tsai CF, Pham AD, Lu YH (2014) Machine learning in concrete strength simulations: Multi-nation data analytics. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2014.09.054
65. Sadat Hosseini A, Hajikarimi P, Gandomi M, et al (2021) Genetic programming to formulate viscoelastic behavior of modified asphalt binder. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2021.122954
66. Nguyen TT, Pham Duy H, Pham Thanh T, Vu HH (2020) Compressive Strength Evaluation of Fiber-Reinforced High-Strength Self-Compacting Concrete with Artificial Intelligence. Adv Civ Eng 2020:. https://doi.org/10.1155/2020/3012139
67. Sultana N, Zakir Hossain SM, Alam MS, et al (2020) Soft computing approaches for comparative prediction of the mechanical properties of jute fiber reinforced concrete. Adv Eng Softw 149:. https://doi.org/10.1016/j.advengsoft.2020.102887
68. Willmott CJ (1981) On the validation of models. Phys Geogr. https://doi.org/10.1080/02723646.1981.10642213
69. Nash JE, Sutcliffe J V. (1970) River flow forecasting through conceptual models part I - A discussion of principles. J Hydrol. https://doi.org/10.1016/0022-1694(70)90255-6
70. Gupta H V., Kling H, Yilmaz KK, Martinez GF (2009) Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. J Hydrol. https://doi.org/10.1016/j.jhydrol.2009.08.003
71. Knoben WJM, Freer JE, Woods RA (2019) Technical note: Inherent benchmark or not? Comparing Nash-Sutcliffe and Kling-Gupta efficiency scores. Hydrol Earth Syst Sci.

https://doi.org/10.5194/hess-23-4323-2019

72. Cheng MY, Chou JS, Roy AFV, Wu YW (2012) High-performance Concrete Compressive Strength Prediction using Time-Weighted Evolutionary Fuzzy Support Vector Machines Inference Model. Autom Constr. https://doi.org/10.1016/j.autcon.2012.07.004
73. Yaseen ZM, Deo RC, Hilal A, et al (2018) Predicting compressive strength of lightweight foamed concrete using extreme learning machine model. Adv Eng Softw. https://doi.org/10.1016/j.advengsoft.2017.09.004
74. Yang L, Qi C, Lin X, et al (2019) Prediction of dynamic increase factor for steel fibre reinforced concrete using a hybrid artificial intelligence model. Eng Struct. https://doi.org/10.1016/j.engstruct.2019.03.105
75. Qi C, Fourie A, Chen Q (2018) Neural network and particle swarm optimization for predicting the unconfined compressive strength of cemented paste backfill. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2017.11.006
76. Chou J-S, Chiu C-K, Farfoura M, Al-Taharwa I (2010) Optimizing the Prediction Accuracy of Concrete Compressive Strength Based on a Comparison of Data-Mining Techniques. J Comput Civ Eng. https://doi.org/10.1061/(asce)cp.1943-5487.0000088
77. Deepa C, SathiyaKumari K, Sudha VP (2010) Prediction of the Compressive Strength of High Performance Concrete Mix using Tree Based Modeling. Int J Comput Appl. https://doi.org/10.5120/1076-1406
78. Erdal HI (2013) Two-level and hybrid ensembles of decision trees for high performance concrete compressive strength prediction. Eng Appl Artif Intell. https://doi.org/10.1016/j.engappai.2013.03.014
79. Yan K, Shi C (2010) Prediction of elastic modulus of normal and high strength concrete by support vector machine. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2010.01.006
80. Rafiei MH, Khushefati WH, Demirboga R, Adeli H (2017) Supervised Deep Restricted Boltzmann Machine for Estimation of Concrete. ACI Mater J 114:. https://doi.org/10.14359/51689560
81. Yan K, Xu H, Shen G, Liu P (2013) Prediction of Splitting Tensile Strength from Cylinder Compressive Strength of Concrete by Support Vector Machine. Adv Mater Sci Eng. https://doi.org/10.1155/2013/597257
82. Anoop Krishnan NM, Mangalathu S, Smedskjaer MM, et al (2018) Predicting the dissolution kinetics of silicate glasses using machine learning. J Non Cryst Solids. https://doi.org/10.1016/j.jnoncrysol.2018.02.023
83. Okuyucu H, Kurt A, Arcaklioglu E (2007) Artificial neural network application to the friction stir welding of aluminum plates. Mater Des. https://doi.org/10.1016/j.matdes.2005.06.003
84. Lim CH, Yoon YS, Kim JH (2004) Genetic algorithm in mix proportioning of high-performance concrete. Cem Concr Res. https://doi.org/10.1016/j.cemconres.2003.08.018
85. Haghdadi N, Zarei-Hanzaki A, Khalesian AR, Abedi HR (2013) Artificial neural network modeling to predict the hot deformation behavior of an A356 aluminum alloy. Mater Des.

https://doi.org/10.1016/j.matdes.2012.12.082

86. Golafshani EM, Behnood A (2019) Estimating the optimal mix design of silica fume concrete using biogeography-based programming. Cem Concr Compos 96:95–105. https://doi.org/10.1016/J.CEMCONCOMP.2018.11.005
87. Naser MZ (2018) Deriving temperature-dependent material models for structural steel through artificial intelligence. Constr Build Mater 191:56–68. https://doi.org/10.1016/J.CONBUILDMAT.2018.09.186
88. Naser MZ (2019) Properties and material models for modern construction materials at elevated temperatures. Comput Mater Sci 160:16–29. https://doi.org/10.1016/J.COMMATSCI.2018.12.055
89. Mousavi SM, Aminian P, Gandomi AH, et al (2012) A new predictive model for compressive strength of HPC using gene expression programming. Adv Eng Softw. https://doi.org/10.1016/j.advengsoft.2011.09.014
90. Gandomi AH, Alavi AH, Sahab MG (2010) New formulation for compressive strength of CFRP confined concrete cylinders using linear genetic programming. Mater Struct Constr. https://doi.org/10.1617/s11527-009-9559-y
91. Mollahasani A, Alavi AH, Gandomi AH (2011) Empirical modeling of plate load test moduli of soil via gene expression programming. Comput Geotech. https://doi.org/10.1016/j.compgeo.2010.11.008
92. Erdal HI, Karakurt O, Namli E (2013) High performance concrete compressive strength forecasting using ensemble models based on discrete wavelet transform. Eng Appl Artif Intell. https://doi.org/10.1016/j.engappai.2012.10.014
93. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02
94. Galdi P, Tagliaferri R (2017) Data Mining: Accuracy and Error Measures for Classification and Prediction. In: Encyclopedia of Bioinformatics and Computational Biology
95. Valença J, Gonçalves LMS, Júlio E (2013) Damage assessment on concrete surfaces using multi-spectral image analysis. Constr Build Mater. https://doi.org/10.1016/j.conbuildmat.2012.11.061
96. Huang H, Burton H V. (2019) Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning. J Build Eng. https://doi.org/10.1016/j.jobe.2019.100767
97. Azimi SM, Britz D, Engstler M, et al (2018) Advanced steel microstructural classification by deep learning methods. Sci Rep. https://doi.org/10.1038/s41598-018-20037-5
98. Hore S, Chatterjee S, Sarkar S, et al (2016) Neural-based prediction of structural failure of multistoried RC buildings. Struct Eng Mech. https://doi.org/10.12989/sem.2016.58.3.459
99. Bhowan U, Johnston M, Zhang M (2012) Developing new fitness functions in genetic programming for classification with unbalanced data. IEEE Trans Syst Man, Cybern Part B Cybern. https://doi.org/10.1109/TSMCB.2011.2167144
100. Boughorbel S, Jarray F, El-Anbari M (2017) Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PLoS One.

https://doi.org/10.1371/journal.pone.0177678

101. Tharwat A (2018) Classification assessment methods. Appl. Comput. Informatics
102. Caruana R, Niculescu-Mizil A (2004) Data mining in metric space: an empirical analysis of supervised learning performance criteria. In: KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining
103. Jurman G, Riccadonna S, Furlanello C (2012) A comparison of MCC and CEN error measures in multi-class prediction. PLoS One. https://doi.org/10.1371/journal.pone.0041882
104. Powers DMW (2011) Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. J Mach Learn Technol. https://doi.org/10.1.1.214.9232
105. Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. https://doi.org/10.1016/S0031-3203(96)00142-2
106. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology. https://doi.org/10.1148/radiology.143.1.7063747
107. Zhang Y, Burton H V., Sun H, Shokrabadi M (2018) A machine learning framework for assessing post-earthquake structural safety. Struct Saf. https://doi.org/10.1016/j.strusafe.2017.12.001
108. Davis J, Goadrich M (2006) The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd international conference on Machine learning - ICML '06
109. Bi, J.; Bennett KPP (2003) Regression Error Characteristic Curves. Proc Twent Int Conf Mach Learn
110. Zhang M, Smart W (2006) Using Gaussian distribution to construct fitness functions in genetic programming for multiclass object classification. Pattern Recognit Lett. https://doi.org/10.1016/j.patrec.2005.07.024
111. Kocher M, Savoy J (2017) Distance measures in author profiling. Inf Process Manag. https://doi.org/10.1016/j.ipm.2017.04.004
112. Patel B V (2012) Content Based Video Retrieval Systems. Int J UbiComp. https://doi.org/10.5121/iju.2012.3202
113. Giusti R, Batista GEAPA (2013) An empirical comparison of dissimilarity measures for time series classification. In: Proceedings - 2013 Brazilian Conference on Intelligent Systems, BRACIS 2013
114. Vuk M, Curk T (2006) ROC Curve , Lift Chart and Calibration Plot. Metod Zv
115. Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010) The balanced accuracy and its posterior distribution. In: Proceedings - International Conference on Pattern Recognition
116. Cohen J (1960) A Coefficient of Agreement for Nominal Scales. Educ Psychol Meas. https://doi.org/10.1177/001316446002000104
117. Artstein R, Poesio M (2008) Inter-coder agreement for computational linguistics. Comput. Linguist.
118. Destercke S (2014) Multilabel Prediction with Probability Sets: The Hamming Loss Case. In: Communications in Computer and Information Science
119. Czajkowski M, Kretowski M (2019) Decision Tree Underfitting in Mining of Gene

Expression Data. An Evolutionary Multi-Test Tree Approach. Expert Syst Appl. https://doi.org/10.1016/J.ESWA.2019.07.019

120. Devarriya D, Gulati C, Mansharamani V, et al (2019) Unbalanced Breast Cancer Data Classification Using Novel Fitness Functions in Genetic Programming. Expert Syst Appl 112866. https://doi.org/10.1016/J.ESWA.2019.112866

121. Bhaskar H, Hoyle DC, Singh S (2006) Machine learning in bioinformatics: A brief survey and recommendations for practitioners. Comput Biol Med. https://doi.org/10.1016/j.compbiomed.2005.09.002

122. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. Proc 14th Int Jt Conf Artif Intell - Vol 2

123. Alavi AH, Gandomi AH (2011) Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing. Comput Struct. https://doi.org/10.1016/j.compstruc.2011.08.019

124. Kingston GB, Maier HR, Lambert MF (2005) Calibration and validation of neural networks to ensure physically plausible hydrological modeling. J Hydrol. https://doi.org/10.1016/j.jhydrol.2005.03.013

125. Kuo YL, Jaksa MB, Lyamin A V., Kaggwa WS (2009) ANN-based model for predicting the bearing capacity of strip footing on multi-layered cohesive soil. Comput Geotech. https://doi.org/10.1016/j.compgeo.2008.07.002

126. Caruana R, Niculescu-Mizil A (2006) An empirical comparison of supervised learning algorithms. In: ACM International Conference Proceeding Series. ACM Press, New York, USA, pp 161–168

127. Williams N, Zander S, Armitage G (2006) A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. Comput Commun Rev. https://doi.org/10.1145/1163593.1163596