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

M. Z. Naser1,2 · Amir H. Alavi3

Received: 29 July 2021 / Accepted: 15 November 2021 / Published online: 24 November 2021
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2021

Abstract
Artificial intelligence (AI) and Machine learning (ML) train machines to achieve a high level of cognition and perform human-like analysis. Both AI and ML seemingly fit into our daily lives as well as complex and interdisciplinary fields. With the rise of commercial, open-source, and user-catered AI/ML tools, a key question often arises whenever AI/ML is applied to explore a phenomenon or a scenario: what constitutes a good AI/ML model? Keeping in mind that a proper answer to this question depends on various factors, this work presumes that a good model optimally performs and best describes the phenomenon on hand. From this perspective, identifying proper assessment metrics to evaluate the performance of AI/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 and sciences applications.

Keywords Error metrics · Machine learning · Regression · Classification

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], ABAQUUS [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 rebar 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%\(^1\) [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].

Diverse sets of performance metrics have been noted in the open literature i.e. correlation coefficient \((R)\), root mean

\(^1\) 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].
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.

**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.

**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 squarness. 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 outliers. 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 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.

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.

**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.

---

2 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.
| 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                                               |
|    | Mean error (ME)         | The average of all errors in a set                                        | $ME = \frac{\sum E}{n}$      | • May not be helpful in cases where positive and negative predictions cancel each other out |
|    |                         |                                                                           |                                | • Works with numeric data                                               |
|    | Mean Normalized Bias (MNB) | Associated with observation-based minimum threshold                  | $MNB = \frac{\sum E/A}{n}$   | • Biased towards overestimations                                       |
|    | Mean Percentage Error (MPE) | Computed average of percentage errors                                | $MPE = \frac{\sum E/A}{n/100}$ | • Undefined whenever a single actual value is zero                     |
|    | Mean Absolute Error (MAE)* | Measures the difference between two continuous variables              | $MAE = \frac{\sum |E|}{n}$    | • Works with numeric data                                               |
|    | Mean Absolute Percentage Error (MAPE)* | Measures the extent of error in percentage terms          | $MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|E_i|}{|A_i|}$ | • Uses a similar scale to input data [48]                             |
|    |                         |                                                                           |                                | • Can be used to compare series of different scales                    |
|    |                         |                                                                           |                                | • Works with numeric data                                               |
|    | Relative Absolute Error (RAE) | Expressed as a ratio comparing the mean error to errors produced by a trivial model | $RAE = \frac{\sum |E_i|}{|A_i|}$ | • Commonly-used as a loss function [49]                                |
|    | Mean Absolute Relative Error (MARE) | Measures the average ratio of absolute error to random error | $MARE = \frac{1}{n} \sum_{i=1}^{n} \frac{|E_i|}{|A_i|}$ | • Cannot be used if there are actual zero values                      |
|    | Mean Relative Absolute Error (MRAE) | Ratio of accumulation of errors to cumulative error of random error | $MRAE = \frac{\sum |E|}{|A|}$ | • 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                                               |
|    |                         |                                                                           |                                | • $E_i$ ranges from zero (being ideal) to infinity                      |
|    |                         |                                                                           |                                | • Works with numeric data                                               |
|    |                         |                                                                           |                                | • Sensitive to outliers (especially of low values)                     |
|    |                         |                                                                           |                                | • Division by zero may occur (if actuals contain zeros)                |
|    |                         |                                                                           |                                | • Works with numeric data                                               |
|    |                         |                                                                           |                                | • For a perfect fit, the numerator equals to zero [50]                |
|    |                         |                                                                           |                                | • Works with numeric data                                               |
| No | Metric                                      | Definition                                                                 | Formula                                                                 | Remarks                                                                                     |
|----|--------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| 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)            | Evalues the absolute fractional error                                     | $FAE = \frac{1}{n} \sum_{i=1}^{n} \frac{2|E_i|}{|A_i|+|P_i|}$               | ● 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     | $RSE = \sum_{i=1}^{n} \frac{E_i^2}{(A_i - A_{mean})^2}$                  | ● 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             | $RRSE = \sqrt{\sum_{i=1}^{n} \frac{E_i^2}{(A_i - A_{mean})^2}}$          | ● 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                               | $GRMSE = \sqrt[n]{\prod_{i=1}^{n} E_i^2}$                               | ● Scale dependent                                                                          |
|    |                                            |                                                                           |                                                                           | ● Less sensitive to outliers than RMSE [52]                                                 |
|    |                                            |                                                                           |                                                                           | ● Works with numeric data                                                                   |
| 18 | Mean Square Percentage Error (MSPE)*       | Evaluates the mean of square percentage errors                           | $MSPE = \sum_{i=1}^{n} \frac{|E_i|}{|A_i|} \frac{100}{n}$                 | ● Non-symmetrical [49]                                                                      |
|    |                                            |                                                                           |                                                                           | Works with numeric data                                                                     |
| 19 | Root Mean Square Percentage Error (RMSPE)* | Evaluates the mean of squared errors in percentages                      | $RMSPE = \sqrt{\sum_{i=1}^{n} \frac{100}{n} \frac{|E_i|}{|A_i|}}$          | ● Scale independent                                                                        |
|    |                                            |                                                                           |                                                                           | ● Can be used to compare predictions from different datasets                               |
|    |                                            |                                                                           |                                                                           | ● Non-symmetrical [49]                                                                      |
|    |                                            |                                                                           |                                                                           | ● Works with numeric data                                                                   |
|    |                                            |                                                                           |                                                                           | ● An extension of RMSE                                                                      |
| No. | Metric | Definition | Formula | Remarks |
|-----|--------|------------|---------|---------|
| 20  | Normalized Root Mean Squared Error (NRMSE)\(^\ast\) | Normalizes the root mean squared error | \[ \text{NRMSE} = \sqrt{\frac{\sum_{i=1}^{n} r^2}{n}} \] | • 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 | \[ \text{NMSE} = \frac{\sum_{i=1}^{n} (A_i - R_i)^2}{\text{variance}} \] | • Biased towards over-predictions [56]  
• Works with numeric data  
• An extension of MSE |
| 22  | Coefficient of Determination (\(R^2\)) | The square of correlation | \[ R^2 = 1 - \frac{\sum_{i=1}^{n} (P_i - A_i)^2}{\sum_{i=1}^{n} (A_i - A_{mean})^2} \] | • 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 | \[ R = \frac{\sum_{i=1}^{n} (A_i - \bar{A})(P_i - \bar{P})}{\sqrt{\sum_{i=1}^{n} (A_i - \bar{A})^2 \cdot \sum_{i=1}^{n} (P_i - \bar{P})^2}} \] | • \(R > 0.8\) implies strong correlation [57]  
• Does not change by equal scaling  
• Can be used in predicting material properties  
• Works with numeric data  
• Scale independent  
• Stable near zero [52] |
| 24  | Mean Absolute Scaled Error (MASE) | Mean absolute errors divided by the mean absolute error | \[ \text{MASE} = \frac{\sum_{i=1}^{n} \frac{A_i - P_i}{|A_i - A_{mean}|}}{n} \] | • 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. \(A_i = kX\) | \[ k = \frac{\sum_{i=1}^{n} (A_i - \bar{A}) (P_i - \bar{P})}{\sum_{i=1}^{n} (A_i - \bar{A})^2} \] | • \(k\) 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] | - | \[ R_m = R^2 \times (1 - \sqrt{R^2 - R_o^2}) \] | • \(R_m\) is an external predictability indicator.  
\(R_m > 0.5\) implies a good fit  
• Works with numeric data |
| 27  | Frank and Todeschini [60] | - | \[ \frac{\sum_{i=1}^{n} (P_i - A_i)^2}{\sum_{i=1}^{n} (P_i - A_i')^2}, \quad A_i' = k \times A_i \]  
Recommend maintaining a ratio of 3–5 between the number of observations and input parameters | • \(k\) is the slope of the regression line  
• Works with numeric data |
| 28  | Objective function by Gandomi et al. [61] | A multi-criteria metric | Function = \( \frac{\text{No}_{\text{Training}}}{\text{No}_{\text{Validation}}} \times \frac{\text{RMSE}_{\text{Training}} + \text{MAE}_{\text{Validation}}}{\text{RMSE}_{\text{Training}} + \text{MAE}_{\text{Validation}} + 1} \) | • This function needs to be minimized to yield highest fitness  
• Can be used in predicting material properties  
• Works with numeric data |
| No | Metric | Definition | Formula | Remarks |
|----|--------|------------|---------|---------|
| 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 | $SI = \sqrt{\frac{\sum (P_{max} - P_{ave})^2}{P_{ave}^2}}$ where, $n$ = number of data sets used during the training phase; $P_{ave}$ = 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 | $SYI = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{P_i - P_{max,i}}{P_{max,i} - P_{min,i}} \right)$ 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 | $a_{20}$-index [66] | Performance index to evaluate predictivity of a model within 20% variation | $a_{20} = \frac{m_{20}}{N}$ 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 | $FB = \frac{\sum_i (A - P)}{\sum_i (A + P)}$ | 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 | $RD = 1 - \frac{\sum_i (\frac{|A - P|}{P_{max} - P_{min}})}{\sum_i (\frac{|A - P|}{P_{max} - P_{min}})}$ | 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. |
| No | Metric                              | Definition                                                                 | Formula                                                                 | Remarks                                                                 |
|----|-------------------------------------|---------------------------------------------------------------------------|-------------------------------------------------------------------------|-------------------------------------------------------------------------|
| 37 | Nash–Sutcliffe coefficient (NSE) [69] | A metric often used in flow predictions                                   | $\text{NSE} = 1 - \left[ \frac{\sum (A-P)^2}{\sum (A-A)^2} \right]$ | • 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                                   | $\text{KGE} = 1 - \sqrt{(r-1)^2 + (\alpha - 1)^2 + (\beta - 1)^2}$, where, $r$ is the linear correlation between the predicted and actuals, $\alpha$ is the magnitude of the variability calculated as the standard deviation in predictions divided by the standard deviation in actuals, $\beta$ 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 |
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).

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 related 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].

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
| 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 categorical 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 categorical data  
• Has an ideal value of 1 and the worst value of zero  
• Works with categorical 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 categorical data  
• Describes proportion of negative cases incorrectly identified as positive cases  
• Works with categorical data |
| 4  | Negative Predictive Value (NPV) | The proportions of negative observations that are true positives | $NPV = \frac{TN}{N+TN} = 1 - FOR$ | • Has an ideal value of 1 and the worst value of zero  
• Works with categorical data  
• Describes proportion of negative cases incorrectly identified as negative cases  
• Works with categorical 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 the individuals with a positive test result for which the true condition is negative  
• Works with categorical 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 negative test result for which the true condition is positive  
• Works with categorical data  
• Describes proportion of false negatives incorrectly rejected  
• Works with categorical data |
| 7  | False Omission Rate (FOR) | Measures the proportion of false negatives that are incorrectly rejected | $FDR = \frac{FN}{FP+FN} = 1 - NPV$ | • Describes proportion of false negatives incorrectly rejected  
• Works with categorical data  
• Describes proportion of false negatives incorrectly rejected  
• Works with categorical 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 categorical 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 categorical 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 categorical data  
• Presents performance at a single class threshold only |
| 11 | Accuracy (ACC) | Evaluates the ratio of number of correct predictions to the total number of samples | $ACC = \frac{TP+TN}{P+N} = \frac{TP+TN}{TPs+FN+FP+TN}$ | • Assumes equal cost for errors [96]  
• Works with categorical data |
| No | Metric                                      | Definition                                      | Formula                                                                 | Remarks                                                                 |
|----|--------------------------------------------|-----------------------------------------------|------------------------------------------------------------------------|-------------------------------------------------------------------------|
| 12 | F\(_1\) score                              | Harmonic mean of the precision and recall     | \(F_1 = \frac{2\text{PPV} \times \text{TPR}}{\text{PPV} + \text{TPR}}\) | • 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 | \(\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}\) | • Measures the quality of binary and multi-class classifications        |
|    |                                            |                                               |                                                                       | • Can be used in classes with different sizes                          |
|    |                                            |                                               |                                                                       | • When MCC equals +1 → perfect prediction, \(\rightarrow 0\) equivalent to a random prediction and \(\rightarrow -1\) false prediction |
|    |                                            |                                               |                                                                       | • Considered as a balanced measures as it involves values of all the four quadrants of a confusion matrix [100] |
|    |                                            |                                               |                                                                       | • Considered as a balanced measures as it involves values of all the four quadrants of a confusion matrix [100] |
| 14 | Bookmaker Informedness (BM) or Youden’s J statistic | Evaluates the discriminative power of the test [101] | \(BM = \text{TPR} + \text{TNR} - 1\)                                | • 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 negative predictions | \(MK = \text{PPV} + \text{NPV} - 1\)                                 | • Measures trustworthiness of positive and negative predictions by a model [103] |
|    |                                            |                                               |                                                                       | • 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                                            |
| No | Metric                                | Definition                                                                 | Formula                                                                 | Remarks                                                                 |
|----|---------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------|------------------------------------------------------------------------|
| 16 | Average Class Accuracy (ACA)          | Measures the average accuracy of predictions in a class                   | \[ \text{ACA} = W \left( \frac{TP}{TP+FP} + (1-W) \left( \frac{TN}{TN+FP} \right) \right) \] 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  
[132] |
| 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] |
| 18 | Area under the ROC curve (AUC)        | Measures the two-dimensional area underneath the entire ROC curve         | \[ \text{AUC} = \sum_{i=1}^{N-1} \frac{1}{2} (\text{FP}_i + 1) (\text{TP}_{i+1} - \text{TP}_i) \text{ or } \text{AUC} = \frac{1}{2}w (h + h') \] 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  
● 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 |
| 20 | Log Loss Error (LLE)                 | Measures the where the prediction input is a probability value            | \[ \text{LLE} = - \sum_{c=1}^{M} y \log p_c \] where, \( M \): number of classes, \( c \): class label, \( y \): binary indicator (0 or 1) if \( c \) is the correct classification for a given observation | ● Linearly penalize incorrect predictions  
● Primarily used in support vector machine |
| 21 | Hinge Loss Error (HLE)                | -                                                                         | \[ \text{HLE} = \max(0,1-q \cdot y) \] where, \( q = \pm 1 \) and \( y \): classifier score |
| No | Metric                             | Definition                                             | Formula                                                                                                                                                                                                 | Remarks                                                                                     |
|----|------------------------------------|--------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| 22 | Wilcoxon–Mann–Whitney (WMW) test [99] | -                                                      | \[ WMW = \sum_{i=1}^{N_c} \sum_{i \in \text{major/less}[i \in \text{Major/class}]} I_{\text{wmw}}(P_i, P_j) \] 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 \( I_{\text{wmw}} \) returns 1 if \( P_i \geq P_j \) and \( P_i \geq 0 \) or 0 if otherwise |
| 23 | Fitness Function Amse (FFA) [99]    | Measures pattern difference between input and output    | \[ FFA = \frac{1}{K} \sum_{i=1}^{K} \left( 1 - \frac{\sum_{c=1}^{C} \left(1 - \text{sig}(P_{ci}) - T_{ci}\right)\text{I}_{\text{zt}}}{N_{ci} \times 2} \right)^2 \]  
\( \text{sig}(x) = \frac{2}{e^x + 1} \)  
\( P_{ci} \) output of a classifier evaluated on the \( i \)th example, \( N_{ci} \): number of examples,  
\( K \): number of classes, \( T_{ci} \): target values (equals to -0.5 and 0.5 for majority and minority classes, respectively) | • Used in scenarios with unbalanced data  
• Appropriate for genetic programming  
• 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]    | -                                                      | \[ \text{Incr} = \frac{1}{K} \sum_{k=1}^{K} \left( \sum_{c=1}^{C} E_{\min,\text{major}}(r, k, c) \right) \]  
\( E_{\min,\text{major}}(r, k, c) = \begin{cases} 0, & \text{if } k < 0 \text{ and } c \in \text{Major} \\ \frac{1}{r_k^c}, & \text{otherwise} \end{cases} \)  
\( r_k^c \) correlation ratio, \( \mu_{\text{minor}} \) and \( \mu_{\text{major}} \): mean for minor and major classes, respectively | • 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 (FCC)  | -                                                      | \[ FCC = \frac{\sum_{i=1}^{N_c} \sum_{i \in \text{minor/less}[i \in \text{Major/class}]} \left( P_{ci} - \mu_{\text{minor}} \right) \left( P_{ci} - \mu_{\text{major}} \right)}{\sqrt{\sum_{i=1}^{N_c} \sum_{i \in \text{minor/less}[i \in \text{Major/class}]} \left( P_{ci} - \mu_{\text{minor}} \right)^2 \sqrt{\sum_{i=1}^{N_c} \sum_{i \in \text{minor/less}[i \in \text{Major/class}]} \left( P_{ci} - \mu_{\text{major}} \right)^2}}} \]  
\( \mu_{\text{minor}}, \sigma_{\text{minor}} = \frac{1}{N_{ci}} \sum_{i=1}^{N_{ci}} \left( P_{ci} - \mu_{\text{minor}} \right) \)  
\( \mu_{\text{major}}, \sigma_{\text{major}} = \frac{1}{N_{ci}} \sum_{i=1}^{N_{ci}} \left( P_{ci} - \mu_{\text{major}} \right) \)  
\( \mu_{\text{minor}}, \mu_{\text{major}} \): mean and standard deviation of the class distribution, respectively | • Used in scenarios with unbalanced data  
• Measures separability (i.e. distance between class distributions) [110] – high separability (no overlap) and this distance turns large (go to +\( \infty \))  
• Uses \( I_{\text{zt}} \) to enforce zero class threshold |
| 26 | Fitness Function Distribution (FFD) | Measures the distance between class distributions as a function of class separability  
| 27 | Canberra Metric (CM)               | Measures the distance between pairs of points in a vector space  
| 28 | Wave Hedges Distance (WHD)         | -                                                      | \[ CM = \sum_{i=1}^{N_c} \frac{|E_i|}{\max(E), P_i) \]  
\[ WHD = \sum_{i=1}^{N_c} \frac{|E_i|}{\max(E), P_i) \]  
where, \( E_i \): output of a classifier evaluated on the \( i \)th example | • Normalizes the difference of each pair of coefficients with its maximum [111–113]  
• 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 +\( \infty \))  
• Uses \( I_{\text{zt}} \) to enforce zero class threshold |
| No | Metric                              | Definition                                                                 | Formula                                                                 | Remarks                                                                 |
|----|-------------------------------------|---------------------------------------------------------------------------|-------------------------------------------------------------------------|-------------------------------------------------------------------------|
| 29 | Lift [114]                          | Measures the performance of a model at predicting or classifying cases    | \[ \text{LIFT} = \frac{\text{Number of true positives above the threshold}}{\text{Number of data instances above the threshold}} \] | • 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 | \[ \text{MXE} = - \frac{1}{N} \sum_{i=1}^{N} \left[ \text{True} \times \ln(\text{Predicted}) + (1 - \text{True}) \times \ln(1 - \text{Predicted}) \right] \] (The assumptions are that Predicted \( \in [0, 1] \) and True \( \in \{0, 1\} \)) | • 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  | \[ \text{Precision} = \text{Recall} \] | • Defines the point when precision and recall are equal |
| 33 | Average precision (AP)              | Combines recall and precision for ranking                                 | \[ \text{AP} = \text{Sum} \left( \text{Recall}_i - \text{Recall}_{i-1} \right) \times \text{Precision}_i \] | • 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 | \[ \text{Defined as the average of recall obtained on each class} \] | • Used in binary and multiclass classification problems  
• Accommodates imbalanced datasets |
| 35 | Brier score (BS)                    | Measures the accuracy of probabilistic-based predictions                 | \[ \text{BS} = \frac{1}{N} \sum_{i=1}^{N} \left( f_i - A_i \right)^2 \] 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 |
| No | Metric | Definition | Formula | Remarks |
|----|--------|------------|---------|---------|
| 36 | Cohen’s kappa (CK) [116] | Measures interrater (agreement) reliability | $\kappa = \frac{p_o - p_e}{1 - p_e}$ | • 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 | $HL = \frac{1}{m} \sum_{i=1}^{m} 1_{P_i \neq A_i}$ | • Describes fraction of labels that are incorrectly predicted  
• Optimal value is zero [118] |
| 38 | Fitness (T) [119] | - | $Fitness(T) = Q(T) + \alpha * R(T) + \beta * \text{Cost}(T)$ | • Used for fitting decision trees  
• This function needs to be maximized to achieve high performance |
| 39 | F2 score [120] | Measured as the weighted average of precision and recall | $F_\beta = \frac{1 + \beta^2 \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$ | • Used in genetic programming and medical fields  
• Computes a weighted harmonic mean of Precision and Recall  
• Learning about the minority class |
| 40 | Distance score (D score) [120] | - | $D_k = \frac{2 \times C_1 \times C_2}{C_1 + C_2}$ | • 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.*
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 indvduial metrics. Such metrics can be of independent nature to each other such as, $R^2$, RSME, and $a_{20}$-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.

**Declarations**

**Conflict of interest** none.

**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 Electrific. 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. Grimmet 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, Sadiq S, Sarace 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 349:255–260. https://doi.org/10.1126/science.aaa8415
15. Seittlari 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. Shamugamani R, Sadique M, Ramamooorthy 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 Mat-lab-like Environment for Machine Learning
84. Lim CH, Yoon YS, Kim JH (2004) Genetic algorithm in mix
87. Naser MZ (2018) Deriving temperature-dependent material mod-
86. Golafshani EM, Behnood A (2019) Estimating the optimal mix
85. Haghdadi N, Zarei-Hanzaki A, Khalesian AR, Abedi HR (2013)
83. Okuyucu H, Kurt A, Arcaklioglu E (2007) Artificial neural net-
82. Anoop Krishnan NM, Mangalathu S, Smedskjaer MM et al
79. Yan K, Shi C (2010) Prediction of elastic modulus of normal and
78. Erdal HI (2013) Two-level and hybrid ensembles of decision
77. Deepa C, SathiyaKumari K, Sudha VP (2010) Prediction of
76. Chou J-S, Chiu C-K, Farfouna M, Al-Taharwa I (2010) Optimiz-
516 Architecture, Structures and Construction (2023) 3:499–517
1 3
107. Zhang Y, Burton HV, 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 BV (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/10.1177/00131644600200104
117. Arstein 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, Gualti 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 AV, 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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.