Metrics for Measuring Error Extents of Machine Learning Classifiers

Hong Zhu, Ian Bayley and Mark Green
School of Engineering, Computing and Mathematics
Oxford Brookes University
Oxford OX33 1HX, UK
Emails: (hzhu|ibayley|mgreen)@brookes.ac.uk

2022 IEEE International Conference On Artificial Intelligence Testing (AITest) | 978-1-6654-8737-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/AITest55621.2022.00016

***Abstract*—Metrics play a crucial role in evaluating the performance of machine learning (ML) models. Metrics for quantifying the extent of errors, in particular, have been intensively studied and widely used but only so far for regression models. This paper focuses instead on classifier models. A new approach is proposed in which datamorphic exploratory testing is used to discover the boundary values between classes and the distance of misclassified instances from that boundary is used to quantify the errors that the model makes. Empirical experiments and case studies are reported that validate and evaluate the proposed metrics.**

***Index Terms*—Machine learning, Classifier, Performance metrics, Extent of errors, Datamorphic testing, Exploratory testing**

I. Introduction

Assessing the quality of machine learning (ML) models is increasingly important due to the rapid growth in their use within computer applications [1], [2]. Such models are intrinsically imperfect due to the nature of inductive inference and the inherent uncertainty associated with ML algorithms. Quality assurance requires testing [3]–[8] to evaluate the model's performance but finding robust measures is still a formidable challenge despite great efforts in the AI research community over several decades.

Existing metrics for classification ML models are of three types:

- Statistical metrics, including *accuracy*, *precision*, *recall*, *F-score*, etc; these are the most commonly used.
- Information theory based metrics, such as *average and relative information scores* [9].
- Graph-based metrics, such as the areas under the receiver operating characteristic (ROC) curves [10], the precision-recall curves, and cost curves, etc. [11].

These metrics measure the frequency of errors, i.e. misclassifications, but not their extents. That is what this paper addresses. Regression model metrics, in contrast, do concentrate on the extent of errors in predictions. One such metric is the root mean square error, where error is also called *loss* in the literature and equals the difference between the expected and actual output. Many more have been proposed, studied and used in practice; see e.g. [12] for a recent review. Extending this approach to classifier models is difficult however. Data may be misclassified as one class α rather than another class β, for example, but a difference cannot be calculated because α and β are not numbers.

The notion of error extent is still important for model evaluation as reflected in terminology such as "near misses" and "serious errors". Suitable metrics are needed, there are none in the literature and formulating them remains an open problem. Our approach is to calculate the boundary between classes using datamorphic exploratory testing [13]–[15]. The classifier model is then tested on a set of labelled test cases and the distances of misclassified test cases from the nearest boundary point can then be used to estimate the extent of errors.

The paper is organised as follows. Section II is a brief introduction to the basic notions and notations. Section III formally defines the proposed metrics. Section IV reports an empirical validation and evaluation of the proposed metrics. Section V concludes the paper with a discussion of related works and future works.

II. Preliminaries

A. Classifiers

A classifier C is a mapping on a given data space $D \neq \emptyset$ to a set of classes $\{l_1, \cdots, l_k\}$, where $k \geq 2$. When $k = 2$, C is a *binary classifier*; otherwise, it is a *multi-class classifier*.

Let $C_i = \{x \in D | C(x) = l_i\}$. We assume that the classifier C is complete and disjoint. That is, $D = \bigcup_{i=1}^{k} C_i$ and $C_i \cap C_j = \emptyset$ for all $i \neq j$ in $\{1, \cdots, k\}$.

We also assume that there is a distance function $\|\cdot,\cdot\| : D \times D \to \mathbb{R}^+$ defined on the data space D, where $\mathbb{R}^+ = \{x \in \mathbb{R} \mid x \geq 0\}$ is the set of non-negative real numbers. Therefore, $\langle D, \|\cdot,\cdot\| \rangle$ forms a metric space. In other words, we have:

- $\forall x \in D.(\|x, x\| = 0)$;
- $\forall x, y \in D.(\|x, y\| \geq 0)$;
- $\forall x, y \in D.(\|x, y\| = \|y, x\|)$;
- $\forall x, y, z \in D.(\|x, y\| + \|y, z\| \geq \|x, z\|)$.

Distance functions are widely used by ML algorithms to build classifier models. For example, they are used in clustering algorithms to measure the similarities between data points [6]–[8]. Here, we use a distance function to define the notion of error extent.

In the remainder of the paper, we use C to notate a conceptual classifier and P_C for an implementation of C, such as a model built through ML. For the sake of simplicity, we will omit the subscript when there is no risk of confusion.

978-1-6654-8737-5/22/$31.00 ©2022 IEEE
DOI 10.1109/AITest55621.2022.00016

B. Pareto Front And Borders Between Classes

We will further assume that in both the conceptual and implemented models C and P_C, the classes are separated by borders. The extent of error can then be defined by how far away an error is from the correct border, at least in theory. However, the correct borders are usually either unknown, even impossible to determine. The implemented borders can, however, be computed with exploratory testing [13], [15] and approximately represented as a Pareto front, a concept first introduced in [14] and defined as follows.

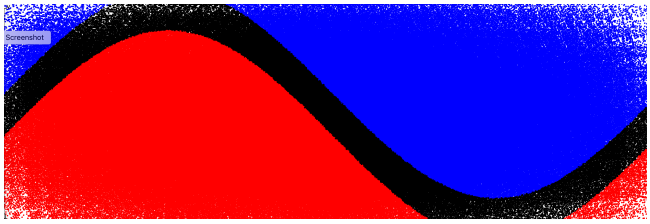
Definition 1: (Pareto Front of Classification)

Let $P : D \to \{l_1, \cdots, l_k\}$, be a classifier, $\|\cdot,\cdot\| : D \times D \to \mathbb{R}^+$ be a distance metric defined on the input space D, and $\delta > 0$ be any given real number. A set $\{\langle a_i, b_i\rangle \mid a_i, b_i \in D, i = 1, \cdots, n\}$ of data pairs is a *Pareto front* between the classes l_u and l_v $(u \neq v)$ according to P with respect to $\|\cdot,\cdot\|$ and δ, denoted by $\Phi_{u,v}$, if for all $i = 1, \cdots, n, P(a_i) = l_u \wedge P(b_i) = l_v$ and $\|a_i, b_i\| \leq \delta$. We also write

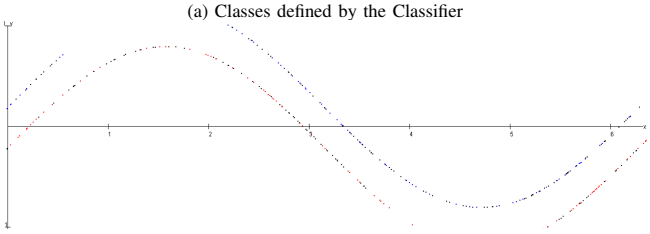
$$\Phi_P = \bigcup_{u \neq v \in \{1,\cdots,k\}} \Phi_{u,v}.$$

□

Fig. 1(a) shows an example of a classifier that classifies the data space into three classes: red, blue and black; and (b) depicts a Pareto front generated by datamorphic exploratory testing. Each dot in Fig. 1(b) is a pair of points in the Pareto front; the points in a pair are too close together to be visually distinguished. The distance between the pair of points represents a tolerable error margin δ that a Pareto front is from the actual border.



(a) Classes defined by the Classifier



(b) Pareto Front Between The Classes

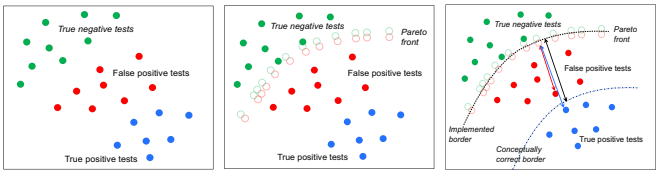
Fig. 1. Example of Pareto Front of Classifier

Empirical studies reported in [14], [15] demonstrated that Pareto fronts to represent borders between classes can be obtained efficient with any reasonable tolerable error margin $\delta > 0$.

C. Process of Testing and Measuring Extents of Errors

To measure the error extent in a classifier ML model, we require a labelled dataset T as the test cases, where each element $x \in T$ is associated with a class label $L(x) = \{l_1, \cdots, l_k\}$ that x should belong to. The proposed process for model evaluation has the following three steps, as illustrated in Fig. 2.

- test the classifier on a labelled dataset, so that the test cases can be marked as true positive, false positive, true negative and false negative as usual.
- apply exploratory datamorphic testing to find a Pareto front, marked here as pairs of red and green rings.
- use the Pareto front and error test cases (i.e. the false positive and false negative test cases) to estimate the distance between the conceptually correct border (marked in blue) and the implemented border of the classifiers.



(a) (b) (c)

Fig. 2. Illustration of the Proposed Approach

Note that the conceptually correct border used by this approach is not usually available for ML applications. The distance between the two borders gives a measure of the extent of errors and in the next section we will propose a set of metrics based on it but first we will introduce some notations.

D. Notations

A labelled test set T can be partitioned according to the labels associated to the test cases and also according to the classifier P. For partitioning according to the labels, we write $T_i = \{x \in T | L(x) = l_i\}$; thus we have that $T = \bigcup_{i=1}^{k} T_i$. We also write t_i to denote $|T_i|$. Thus, we have that $\sum_{i=1}^{k} t_i = |T|$. For partitioning according to a classifier P, we write that $T_i^P = \{x \in T | P(x) = l_i\}$; thus we have that $T = \bigcup_{i=1}^{k} T_i^P$. We also write p_i to denote $|T_i^P|$. Thus, we have that $\sum_{i=1}^{k} p_i = |T|$.

For all $i, j = 1, \cdots, k$, we write $T_{i,j}$ to denote the subset of test cases in T labelled as l_i but classified as l_j by model P. That is,

$$T_{i,j} = \{x \in T | L(x) = l_i \wedge P(x) = l_j\}. \quad (1)$$

Note that, for all $x \in T_{i,i}$, the classifier P's output on x is correct with respect to its label in T. When $i \neq j$, for all $x \in T_{i,j}$, the classifier P produces an incorrect classification of x w.r.t. labels in T. Such an error will be called an i/j error.

In particular, for binary classifiers that have labels p for positive and n for negative, the p/n errors are *false negative* errors. The n/p errors are *false positive* errors. The test cases in the set $T_{p,p}$ and $T_{n,n}$ are true positive, and true negative test cases, respectively.

Let $\Phi_{i,j}$ be a Pareto front of classifier P between classes l_i and l_j. For all $i, j \in \{1, \cdots, k\}$ and $i \neq j$, we define $\Psi_{i/j}$ as follows.

$$\Psi_{i/j} = \{x | \langle x, y \rangle \in \Phi_{i,j}\} \tag{2}$$

Informally, $\Psi_{i/j}$ contains all data points in the Pareto front $\Phi_{i,j}$ that are classified as in class l_i.

The distance from a given data point a to a set X of data points, written $\|a, X\|$, is defined as the minimal distance to the elements of X. Formally,

$$\|a, X\| = Min_{x \in X} \|a, x\|. \tag{3}$$

The distance from an error point $a \in T_{i,j}, i \neq j$, to a Pareto front Φ, written $\|a, \Phi\|$, is defined as follows.

$$\|a, \Phi\| = \|a, \Psi_{i/j}\| \tag{4}$$

For $a \in T_{i,j}$, we write $W(a, \Phi)$ to denote the subset of $\Psi_{i/j}$ whose elements are nearest to a. That is, $W(a, \Phi)$ contains points x whose the distance to a equals $\|a, \Phi_{i,j}\|$. Formally, W is defined as follows.

$$W(a, \Phi) = \{x \in \Psi_{i/j} | \|a, x\| = \|a, \Psi_{i/j}\|\} \tag{5}$$

The W function is extended to be on a set of error points as follows.

$$W(T_{i,j}, \Phi) = \bigcup_{x \in T_{i,j}} W(x, \Phi) \tag{6}$$

We write $W^{\Phi}_{i,j}$ to denote $W(T_{i,j}, \Phi)$ for the sake of simplicity.

E. Assumptions and Hypotheses

The proposed metrics rely on an intuition, usually valid but not always, that we formalise as an *admissibility condition* below:

Definition 2: (Admissibility Condition) Let C and P be classifiers on data space D where C is the conceptual model and P is an implementation of C. We say that P is *admissible* with respect to C, if for all $x \in D$, x is an error of i/j type with respect to C implies that the nearest point y to x on the border between classes i and j as defined by P is also an i/j error point. □

In 2(c), the red arrow connects a false positive test case to its the nearest point in the Pareto front. The admissibility condition states that any point along that arrow should also be a false positive.

When models C and P satisfy the admissible condition, we can infer that the points in $W^{\Phi}_{i,j}$ should also be errors. Moreover, they are the worst cases of errors that the model P could make as illustrated in Fig. 2(c). Therefore, $W^{\Phi}_{i,j}$ is the set of *inferred worst errors of type* i/j *on Pareto front* Φ.

The admissibility condition for P cannot be verified without direct access to the conceptual classifier C, a requirement that will not be met as the purpose of machine learning is to formulate P as an approximation to an unknown C. However, we believe the admissibility condition will hold in the majority of cases, based on two hypotheses.

The first hypothesis concerns the continuity of the classification models:

Definition 3: (Continuity Hypothesis) The *continuity hypothesis* is that the classes of the conceptual classifier C and its ML implementation model P are both formed from unions of a finite number of continuous/consecutive sub-domains of the data space. □

This hypothesis implies that the error points of the classifier P are in continuous (or consecutive) blocks (i.e. sub-domains) of the input data space. Typically, this requires both models C and P to have smooth curves or planes as borders between classes. Whether this requirement is satisfied or not depends heavily on the complexity of the application problem, i.e. the conceptual model.

The second hypothesis concerns the competency of the data scientist. This is needed for ML model P to be of high quality. In other words, it is not too far away from the conceptual classifier C in the sense that systematic errors only occur around the borders between classes and the test dataset T does not contain any label noise with respect to C.

Definition 4: (Competent Data Scientist Hypothesis)

The *competent data scientist* hypothesis assumes that the ML model P and the test set T are well developed by competent data scientists. □

Note that ML models are intrinsically imperfect due to the nature of inductive inference so they will have errors even if the developers make no mistakes. In this sense, it is similar to the *competent programmer* hypothesis that underlies the mutation testing method, which similarly does not require the program under test to be perfect. Imperfection in ML models also arises from the uncertainty associated with ML algorithms and the development and operation processes.

Here, we distinguish two types of mistakes that a data scientist may make in the development of an ML model: *systematic errors*, and *random errors*. Systematic errors are methodological and technical errors made when developing the classifier. For example, the wrong ML algorithm may be chosen with the wrong parameters and the wrong methods may be used when preparing and processing the datasets. Mistakes of this sort can result in the ML model being significantly different from the conceptual model but competent data scientists would not make them.

Random errors, in contrast, are neither methodological nor technical; for example, assigning a wrong label to a test case in the test data set. This causes label noise that can and should be dealt with by noise detection and removal techniques; see e.g. [8], [16].

The continuity hypothesis combined with the competent data scientist hypothesis together imply that the errors in an ML model will be continuous regions on the borders between classes. Therefore, they imply that the admissible condition is true.

In Section IV, we will conduct an experiment to evaluate how likely it is that the admissible conditions will hold.

III. Proposed Metrics

In this section, we formally define the metrics for measuring error extent first on individual errors, then on classes and then on the whole ML model.

A. Lower Bound Estimations of Error Extent

Consider an i/j error detected by testing a model P on a test case x. The distance from x to the segment $\Phi_{i,j}$ of Pareto front Φ on the border between classes i and j gives a lower bound between the correct border and the implemented border as illustrated in Fig. 2(c) where the distance is shown by the red arrow. This test case provides evidence that the extent of errors that the model may produce is at least $\|x, \Phi_{i,j}\|$. When there are many error points detected by the test, we can use the maximal value of the distances and the average of the distances to give different estimation of the error extent. Thus, we have the following two metrics ME and AE. Let $i \neq j \in \{l_1, \cdots, l_k\}$.

Definition 5: (Metrics as Lower Bound Estimations)

The metric of *maximal extent of i/j errors on test set T with respect to a Pareto front* Φ, written $ME_{i/j}(T, \Phi)$, is defined as follows.

$$ME_{i/j}(T,\Phi) = \begin{cases} 0, & T_{i,j} = \emptyset \\ \infty, & T_{i,j} \neq \emptyset \wedge \Phi_{i,j} = \emptyset \\ \max\limits_{x \in T_{i,j}} \{\|x, \Phi_{i,j}\|\}, & T_{i,j} \neq \emptyset \wedge \Phi_{i,j} \neq \emptyset \end{cases}$$

The metric of *average extent of i/j errors on test set T with respect to a Pareto front* Φ, written $AE_{i/j}(T, \Phi)$, is defined as follows.

$$AE_{i/j}(T,\Phi) = \begin{cases} 0, & T_{i,j} = \emptyset \\ \infty, & T_{i,j} \neq \emptyset \wedge \Phi_{i,j} = \emptyset \\ \frac{\sum\limits_{x \in T_{i,j}} \|x, \Phi_{i,j}\|}{|T_{i,j}|}, & T_{i,j} \neq \emptyset \wedge \Phi_{i,j} \neq \emptyset \end{cases}$$

□

B. Upper Bound Estimations of Error Extent

Given a point $a \in T_{i,j}$ of i/j type of errors detected by testing on T and a Pareto front Φ, we can infer that points in $W(a, \Phi)$ are also errors and they are the worst cases of i/j type of errors; see discussion in Section II-E. Using the set of test cases on which the model is correct, we can also make an upper bound estimation of the error extent as illustrated by the blue arrow in Fig. 2(c). It is the minimal value of the distances from the point $b \in W(a, \Phi)$ on Pareto front $\Phi_{i,j}$ to correctly classified points in class i. When considering all points in $W^{\Phi}_{i,j}$, we can calculate the maximal and average values of such distances as upper bound estimations of the maximal error extent and average error extent. Thus, we have the following two metrics MC and AC.

Definition 6: (Metrics of Upper Bound Estimations)

The metric of *maximal distance from points on Pareto front Φ to the set of correct points in test set T*, written $MC_{i/j}(T, \Phi)$, is defined as follows.

$$MC_{i/j}(T,\Phi) = \begin{cases} \infty, & T_{i,i} = \emptyset \\ 0, & T_{i,i} \neq \emptyset \wedge W^{\Phi}_{i/j} = \emptyset \\ \max\limits_{x \in W^{\Phi}_{i/j}} \{\|x, T_{i,i}\|\}, & T_{i,i} \neq \emptyset \wedge W^{\Phi}_{i/j} \neq \emptyset \end{cases}$$

The metric of *average distance from i/j Pareto front Φ to the set of correct points in test set T*, written $AC_{i/j}(T, \Phi)$, is defined as follows.

$$AC_{i/j}(T,\Phi) = \begin{cases} \infty, & T_{i,i} = \emptyset \\ 0, & T_{i,i} \neq \emptyset \wedge W^{\Phi}_{i/j} = \emptyset \\ \frac{\sum\limits_{x \in W^{\Phi}_{i/j}} (\|x, T_{i,i}\|)}{|W^{\Phi}_{i/j}|}, & T_{i,i} \neq \emptyset \wedge W^{\Phi}_{i/j} \neq \emptyset \end{cases}$$

□

C. Middle Estimation of Error Extent

Having defined the upper and lower bound estimations of error extents, we now define the middle estimation of error extent by taking the averages of the upper and lower estimations. Thus, we have the following two metrics WEE and AEE.

Definition 7: (Metrics of Worst and Average Extents of Errors)

The metric of *worst extent of errors* for i/j errors as demonstrated by test set T and Pareto front Φ, written $WEE_{i/j}(T, \Phi)$, is defined as follows.

$$WEE_{i/j}(T,\Phi) = \frac{ME_{i/j}(T,\Phi) + MC_{i/j}(T,\Phi)}{2}.$$

The metric of *average extent of errors* for i/j errors as demonstrated by test set T and Pareto front Φ, written $AEE_{i/j}(T, \Phi)$, is defined as follows.

$$AEE_{i/j}(T,\Phi) = \frac{AE_{i/j}(T,\Phi) + AC_{i/j}(T,\Phi)}{2}.$$

□

D. Metrics of Error Extent on A Class and A Model

Given a class l_i of an ML classifier model, there may be many types of errors, i.e. i/j errors for $j \neq i$. The extent of errors on class l_i can be calculated via the following metrics based on the error type specific metrics defined above.

Let μ be any of ME, AE, MC, AC, WEE, and AEE.

Definition 8: (Metrics of Errors Extent for A Class)

The metrics of the worst and average extents of errors for class i, denoted by μ_i^{Max} and μ_i^{Avg} respectively, are defined as follows.

$$\begin{aligned} \mu_i^{Max}(T,\Phi) &= Max_{j \neq i}\{\mu_{i/j}(T,\Phi)\} \\ \mu_i^{Avg}(T,\Phi) &= \frac{\sum_{j \neq i}(\mu_{i/j}(T,\Phi))}{k-1} \end{aligned}$$

□

Now, we define the metrics to estimate the overall error extents of a model P based on testing on T as follows.

Definition 9: (Metrics of ML Model's Overall Error Extent)

The metrics of a model's overall worst and average error extent are denoted by μ^{Max} and μ^{Avg} respectively and defined as follows.

$$\begin{aligned} \mu^{Max}(T,\Phi) &= Max_{i=1}^{k}\{\mu_i^{Max}(T,\Phi)\} \\ \mu^{Avg}(T,\Phi) &= \frac{\sum_{i=1}^{K}(\mu_i^{Avg}(T,\Phi))}{k} \end{aligned}$$

where μ_i is an error extent metrics of class i defined in Definition 8. □

IV. Empirical Evaluation and Validation

This section reports the evaluation of the metrics with controlled experiments using manually-coded examples and empirical case studies using real-world datasets. We will address the following research questions.

RQ1: How likely is it that the admissible condition will hold?

RQ2: Do the metrics actually measure the performance of the ML classifiers?

A. Evaluating the Admissible Condition

Research question RQ1 cannot be answered without the knowledge of the conceptual classifiers. Therefore, we conducted a controlled experiment with a set of 10 manually coded classifiers as the subjects; shown in Fig. 3. They are all on the same input data space of two-dimensional real numbers in the region of $[0, 2\pi] \times [-1, 1]$.

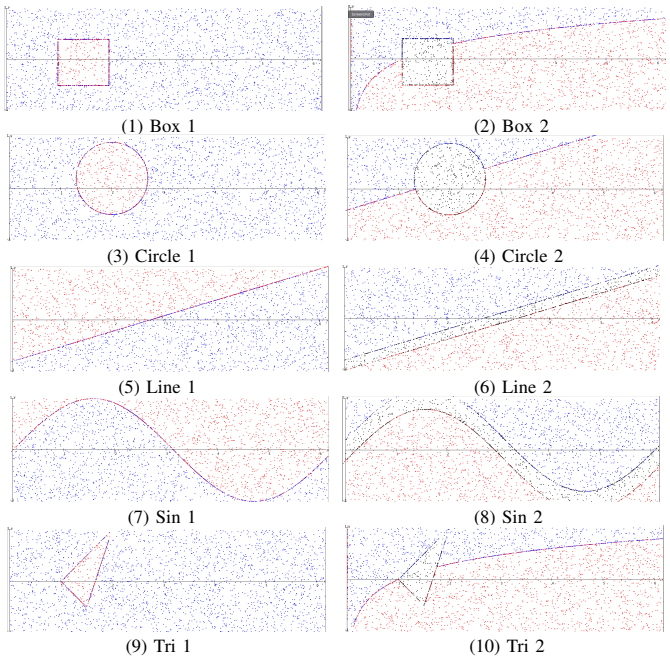


Fig. 3. Manually Coded Classifiers

The experiment consists of the following steps for each coded classifier.

1) Generate a dataset of 5000 labelled data by random sampling of the data space with the uniform distribution, where the labels are generated from the coded target classification models.
2) Build a set of 9 ML models for each dataset by using different ML techniques given in Table I. A total of 90 models were built.
3) Generate a Pareto front for each ML model using the Morphy automated test tool [17] to perform datamorphic exploratory testing via 5000 random walks.
4) Validate the hypotheses for each ML model by testing the model on 5000 random test cases to find test cases that are incorrectly classified. The nearest point on the Pareto front from each error test case is checked for the admissibility condition, i.e. whether it is also incorrectly classified using the manually coded classifier as the standard classification. If yes, it validates the admissible condition; otherwise it is regarded as an invalidating case.
5) Statistical analysis of data. The validity rate of the admissible condition on each model is calculated from the number of valid cases and the total number of cases. The results are shown in Table II.

TABLE I
Machine Learning Techniques Used in The Experiments

Name	ML Techniques Used
DNN	Deep neural network
DT	Decision tree
KNN	K-Nearest neighbors
LR	Logistic regression
NB	Naive Bayes
SVM	Supporting vector machine
HV	Ensemble via hard voting
SV	Ensemble via soft voting
Stack	Ensemble via stacking

It was found that the Pareto front was empty for two of the models. This means that exploratory testing found no borders between classes. The models were then investigated with intensive random testing. This confirmed that there were no borders to be found as the models did not classify the data space into two classes. These two models were therefore excluded from the validity calculation. Also excluded were two more in which error test cases were not found. All four exclusions are marked as - in Table II. The average validity rate over 86 models is 94.84%.

There are five ML models with a very poor validity rate, however. Fig. 4 compares them in the right column against the manually coded classifiers in the left column and it can be seen that they are very different. Error points in these ML models are not near to the borders between classes so they do not satisfy the competent data scientist hypothesis. When these five models are removed, the overall validity rate rises to 99.05%.

Fig. 5 for contrast shows nine models of Box 2. They are of high validity rate. The errors are in the continuous areas next to the borders, thus satisfying the hypothesis.

In summary, the admissibility condition holds in more than 99% of cases where the ML model is well developed. Poor quality models can easily be detected as their ME values will exceed their MC values or their AE values will exceed their AC values or both.

B. Validating The Metrics as Performance Measurements

To answer RQ2, we have used both manually coded classifiers and real world datasets to demonstrate a correlation between our proposed metrics and existing statistical metrics.

TABLE II
SUBJECT MODEL'S VALIDITY RATE

Model	Box1	Box2	Circle1	Circle2	Line1	Line2	Sin1	Sin2	Tri1	Tri2	Average
DT	–	1	0.9677	0.9885	0.9828	0.9844	1	0.9894	1	0.9634	0.9862
HV	1	1	1	0.9804	0.9778	0.9891	1	0.9271	1	0.9891	0.9863
KNN	1	0.9789	1	1	1	0.9944	1	1	1	0.9663	0.9940
LR	0.0413	1	0.2475	1	1	1	1	1	0.1418	1	0.7431
NB	1	1	1	1	1	0.139	1	0.5744	1	0.979	0.8692
Stack	–	1	1	1	1	1	1	1	1	0.9737	0.9971
SV	1	1	1	0.9726	1	0.9934	0.9753	1	1	0.9857	0.9927
SVM	1	0.9948	1	1	1	1	1	1	–	0.8325	0.9808
DNN	1	1	1	1	1	1	1	1	–	0.8719	0.9858
Average	0.8630	0.9971	0.9128	0.9935	0.9956	0.9000	0.9973	0.9434	0.8774	0.9513	0.9484

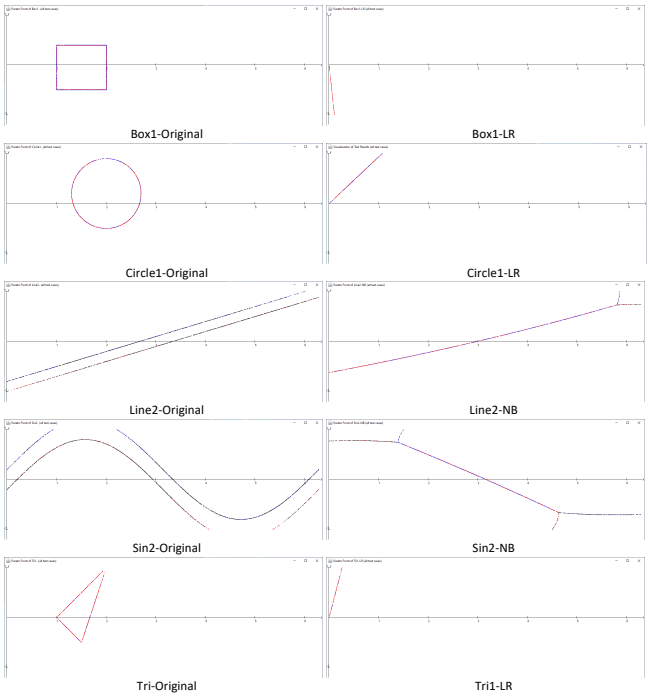


Fig. 4. Models where continuity hypothesis is broken

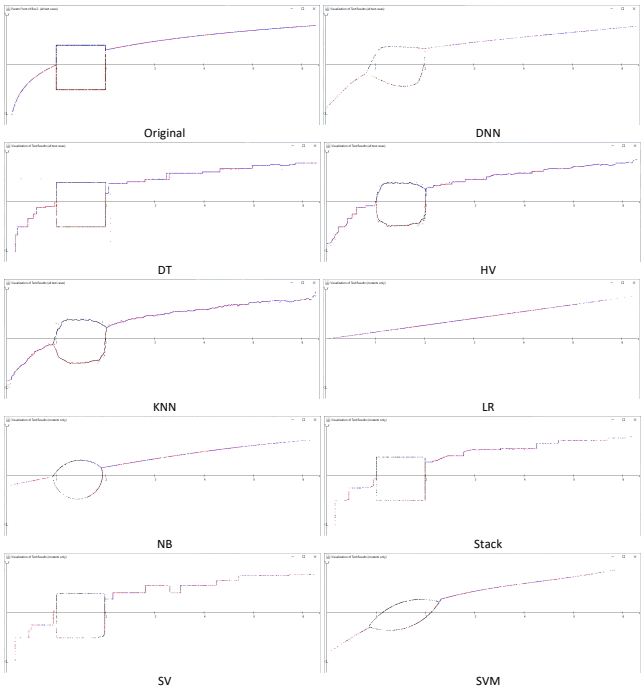


Fig. 5. Pareto fronts of Box2 models

These include *accuracy*, *precision* (also called *purity*), recall (also called *sensitivity* and *true positive rate*), specificity (also called *true negative rate*), *false positive rate*, and the *F-Score*. These are the most widely used performance metrics.

Ten datasets were selected at random from the well-known Kaggle collection and are summarised in Table III. Column *Size* is the number of records in the dataset and column *Cls* is the number of classes in the classification. Columns *DF*, *NF* and *CF* are the numbers of discrete non-numerical features, discrete numerical features and continuous numerical features, respectively.

The experiments consists of the following steps.

1) Build a set of 9 ML models from each dataset using the ML techniques listed in Table I.
2) Test each model and evaluate its performance using the statistic metrics.

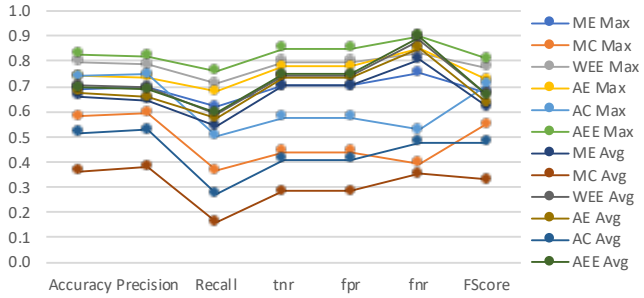
TABLE III
SUMMARY OF DATASETS

Dataset	**Size**	**DF**	**NF**	**CF**	**Cls**
Beer Ranking	5558	0	14	1	11
Bank Churners Prediction	10127	5	11	3	2
DDoS Detection	40000	0	48	24	3
Diabetes Diagnoses	390	1	10	3	2
Ethereum Fraud	9012	0	12	25	2
Hacker of Bank Accounts	23674	0	15	0	2
Heart Attack Likelihood	303	0	12	1	2
Mushroom Edibility	8124	22	0	0	2
Red Wine Quality	1599	0	0	11	10
Water Potability	2012	0	0	9	2

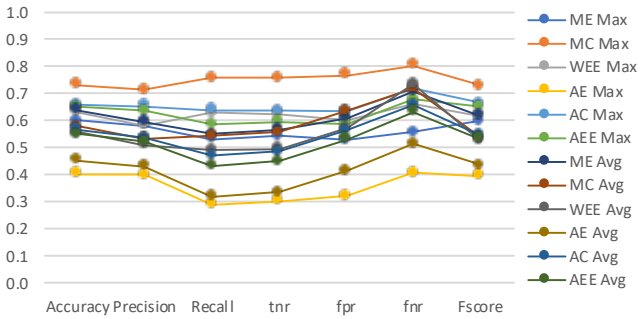
TABLE IV
OVERALL AVERAGES AND STDEVS OF CORRELATION COEFFICIENTS

Metrics	Averages of Correlation Coefficients			StDev of Correlation Coefficients		
	Real Dataset	Coded	Average	Real Dataset	Coded	Average
ME Max	0.5623	0.6926	0.6275	0.3879	0.2593	0.3236
MC Max	0.7510	0.4806	0.6158	0.1598	0.4800	0.3199
WEE Max	0.6211	0.7851	0.7031	0.2451	0.2195	0.2323
AE Max	0.3596	0.7561	0.5579	0.3468	0.2356	0.2912
AC Max	0.6572	0.6258	0.6415	0.2092	0.4433	0.3262
AEE Max	0.6267	0.8313	0.7290	0.2396	0.1996	0.2196
ME Avg	0.6131	0.6682	0.6407	0.3456	0.2484	0.2970
MC Avg	0.5853	0.3078	0.4466	0.4304	0.5614	0.4959
WEE Avg	0.5540	0.7176	0.6358	0.3715	0.2626	0.3170
AE Avg	0.4145	0.6962	0.5554	0.3663	0.2571	0.3117
AC Avg	0.5467	0.4420	0.4943	0.3663	0.5144	0.4403
AEE Avg	0.5204	0.7198	0.6201	0.3874	0.2535	0.3205

3) Generate a Pareto front for each model by datamorphic exploratory testing via 5000 random walks.
4) Measure model performances using the proposed metrics.
5) Calculate the Pearson correlation coefficients between the statistic metrics and proposed metrics. The overall results are summarised in Fig. 6.



(a) ML Models of Coded Classifiers



(b) ML Models of Real Datasets

Fig. 6. Distribution of Correlation Coefficients

The results, shown in Table IV, are that the WEE and AEE metrics have the highest average correlation coefficients with statistic metrics of model performance and they also have the lowest average standard deviations: 0.7031 for WEE^{Max} and 0.7290 for AEE^{Max}.

Therefore, we can conclude that the proposed metrics are valid measurements of the performance of an ML model.

V. CONCLUSION

A. Related Work

In this paper, we have proposed a set of metrics for measuring the extent of errors in ML classifiers based on datamorphic exploratory testing. The metrics are validated and evaluated via both controlled experiments and real dataset case studies. The results show that the proposed metrics are highly correlated to the performances of the ML models as measured by a set of statistical metrics. As far as we know, there are no such metrics for ML classifiers proposed and studied in the literature, even though the metrics for regression models are all concerned with the error extents.

The existing metrics for ML classifiers have been the subject of comparison and analysis in a large volume of research efforts. The following are some of the most well known.

In [18], Huang et al proposed two formal criteria for comparing metrics: (a) the degree of consistency and (b) the degree of discrimination. They also used these criteria to create new metrics of better discrimination power and guide training models. Seliya et al [19] analysed the relationships between performance metrics. In [20], Sokolova and Lapalme assessed performance metrics on their capability to deal with data imbalance and class skews. The notion of class skew is represented by transformations on confusion matrices. A set of statistical metrics were then assessed based on their invariance to such transformations. In [21], Raze et al conducted a statistical evaluation of 24 performance metrics on classifier ML models. They used three datasets to build classifier ML models with balance and unbalanced training data by applying 11 different ML algorithms. They observed that those most commonly used performance metrics are not the best ones. In a recent review paper [12], Naser and Alavi listed 38 Error metrics for regression ML models and 40 performance metrics for classifier ML models. They discussed the limitations of each metric in the context of their application in science and engineering. Huang et al. [22] assessed 14 metrics on their degree of discrimination in performance evaluation of ML models in the context of risk predictions for clinical decision-making. They demonstrated that commonly reported metrics may not have sufficient sensitivity to identify improvement of machine learning models.

Some ML algorithms solve classification problems using regression models, such as artificial neural networks. Such a model produces a vector $\langle r_1, \cdots, r_k \rangle$ of real numbers between 0 and 1 where r_i represents the likelihood of the input data belongs to class l_i, and then selects the class l_c as the classification if r_c is the largest likelihood. The loss of the classification on the input data is defined as $1 - r_c$. A loss function, such as root mean square error (RMSE), on a set of test cases can be defined based on the losses on the individual test cases. Many loss functions have been studied in the research on ML algorithms and proven useful for training ML models including RMSE, hinge, Huber, log, logistic, Lipschitz, ramp, surrogate, etc. [7]. However, loss does not correctly represent the notion of error extent in the

context of classification problems. In particular, it is rare that $r_c = 1$ even if the classification is correct. Thus, even for correctly classify data, the loss is still greater than 0, while the extent of error should be 0. Moreover, a data of small loss could be seriously misclassified, while data of big loss could be correctly classified. As far as we know, all loss functions based on the notion of loss in the literature do not measure the extents of errors. It is worth noting that the zero-one loss function and all loss functions based on it are essentially the statistical performance metrics. Thus, they are not metrics for measuring the extent of errors. Yet, for some ML algorithms such as decision trees, a classifier model does not produce the loss of classifications at all.

B. Future Work

In addition to the work reported in this paper, we are conducting further experiments and case studies with the metrics on their power of discrimination. Another future work is to investigate their resilience to class skews and class imbalance of the test dataset. An observation that we have made in our experiment and case study is that the metrics seems useful to detect models that does not meet the continuity hypothesis. It is worthy further research.

ACKNOWLEDGEMENT

The research work reported in this paper is partly funded by the 2021 Research Excellence Award of Oxford Brookes University, UK. The authors are grateful to the members of Cloud Computing and Cybersecurity Research Group and the AI Software Engineering Reading Group of the School of Engineering, Computing and Mathematics, Oxford Brookes University, UK, for their contributions in the discussions on research related topic.

REFERENCES

[1] F. Khomh, B. Adams, J. Cheng, M. Fokaefs, and G. Antoniol, "Software engineering for machine-learning applications: The road ahead," *IEEE Software*, vol. 35, no. 05, pp.81–84, Sept 2018.

[2] S. Masuda, K. Ono, T. Yasue, and N. Hosokawa, "A survey of software quality for machine learning applications," in *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE Computer Society Press, Apr 2018, pp.279–284.

[3] T. M. King, J. Arbon, D. Santiago, D. Adamo, W. Chin, and R. Shanmugam, "AI for testing today and tomorrow: Industry perspectives," in *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*. IEEE Computer Society Press, Apr 2019, pp.81–88.

[4] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

[5] K. Stapor, "Evaluation of classifiers: current methods and future research directions," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*, Sept 2017, pp.37–40.

[6] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. The MIT Press, 2012.

[7] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[8] C. Aggarwal, *Data Mining: The Textbook*. Springer, 2015.

[9] I. Kononenko and I. Bratko, "Information-based evaluation criterion for classifiers performance," *Machine Learning*, vol. 6, pp.67–80, 1991.

[10] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recogn.*, vol. 30, no. 7, pp.11451159, Jul 1997.

[11] R. C. Prati, G. E. A. P. A. Batista, and M. C. Monard, "A survey on graphical methods for classification predictive performance evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 11, pp.1601–1618, 2011.

[12] M. Z. Naser and A. H. Alavi, "Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences," *Architecture, Structures and Construction*, 2021.

[13] H. Zhu, D. Liu, I. Bayley, R. Harrison, and F. Cuzzolin, "Datamorphic testing: A method for testing intelligent applications," in *Proc. of The First IEEE International Conference on Artificial Intelligence Testing (AITest 2019)*. IEEE Computer Society Press, Apr 2019, pp.149–156.

[14] ——, "Exploratory datamorphic testing of classification applications," in *Proc. of The 1st IEEE/ACM International Conference on Automation of Software Test (AST 2020)*, July 2020, pp.51–60.

[15] H. Zhu and I. Bayley, "Discovering boundary values of feature-based machine learning classifiers through exploratory datamorphic testing," *Journal of Systems and Software*, vol. 187, p.111231, May 2022.

[16] G. Pandey, V. Kumar, H. Xiong, and M. Steinbach, "Enhancing data analysis with noise removal," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 03, pp.304–319, Mar 2006.

[17] H. Zhu, I. Bayley, D. Liu, and X. Zheng, "Automation of datamorphic testing," in *Proc. of 2nd IEEE International Conference on Artificial Intelligence Testing (AITest 2020)*, May 2020.

[18] J. Huang and C. X. Ling, "Constructing new and better evaluation measures for machine learning," in *International Joint Conference on Artificial Intelligence (IJCAI'07)*, Jan 2007, pp.859–864.

[19] N. Seliya, T. M. Khoshgoftaar, and J. V. Hulse, "A study on the relationships of classifier performance metrics," in *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*. IEEE Computer Society Press, Nov 2009, pp.59–66.

[20] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, pp.427–437, Jul 2009.

[21] A. Rcz, D. Bajusz, and K. Hberger, "Multi-level comparison of machine learning classifiers and their performance metrics," *Molecules*, vol. 24, no. 15, 2019.

[22] C. Huang, S.-X. Li, C. Caraballo, F. A. Masoudi, J. S. Rumsfeld, J. A. Spertus, S.-L. T. Normand, B. J. Mortazavi, and H. M. Krumholz, "Performance metrics for the comparative analysis of clinical risk prediction models employing machine learning," *Circulation: Cardiovascular Quality and Outcomes*, vol. 14, no. 10, p.e007526, 2021.