National Physical Laboratory

NPL Report MS 62

CHARACTERISING AND TESTING THE TRUSTWORTHINESS OF ARTIFICIAL INTELLIGENCE SYSTEMS

TAMEEM ADEL, MILES MCCRORY, CONNOR TYNAN, ANDREW THOMPSON AND PAUL DUNCAN

March 2025

Department for Science, Innovation & Technology

NPL Report MS 62

Characterising and Testing the Trustworthiness of Artificial Intelligence Systems

Tameem Adel, Miles McCrory, Connor Tynan, Andrew Thompson and Paul Duncan

Data Science & AI Department

ABSTRACT

In many applications, decision making has recently become dependent on artificial intelligence (AI) systems. In order to ensure a safe integration of such systems within these applications, not only should their accuracy and performance be tested, but also their trustworthiness. We discuss here the basic phases involved when testing the trustworthiness of an AI system, as well as some of the steps that can be taken to ensure an AI system is trustworthy. We begin by discussing characteristics which should be considered for most AI systems, prior to moving on to other characteristics of trustworthiness which can be essential for some AI systems, particularly those which are sensitive and have a direct impact on people's lives. We also shed light on the fact that trustworthiness, along with its evaluation, should be fit for purpose and should be aligned with the original context in which the respective AI system will be deployed. We also examine the role of third-party testing in the development and deployment of AI and ML systems, outlining some of the related benefits, risks, and best practices for mitigating these risks.

NPL Report MS 62

ISSN 1754-2960

https://doi.org/10.47120/npl.MS62

National Physical Laboratory
Hampton Road, Teddington, Middlesex, TW11 0LW

This work was funded by the UK Government's Department for Science, Innovation & Technology through the UK's National Measurement System programmes.

Approved on behalf of NPLML by Louise Wright,

Head of Science for Data Science & AI

CONTENTS

1 INTRODUCTION

Artificial Intelligence (AI) refers to technologies which enable machines to address problems and tasks typically requiring human intelligence. Such tasks include learning, automating repetitive work, problem solving, and making predictions. Machine learning (ML) is a subset of AI consisting of methods that are able to learn statistical models by analysing data. Such models may be used for classification and prediction tasks to aid decision-making. The focus of this report is on ML components of AI systems, but we will still use the term AI at times when referring to the more general technology.

AI systems, and the ML models which typically power them, are now ubiquitous in nearly every area of our daily lives. As such, it is of paramount importance to assess their trustworthiness in order to make sure we can benefit as much as possible from ML, while mitigating any potentially negative side effects [33] such as bias and opacity. Trustworthy AI systems should be reliable, ethical, and transparent in their operation and decision-making.

Testing the trustworthiness of ML-based AI systems is different from testing a traditional software system for at least two reasons. Firstly, ML is data-driven as opposed to rule-based, which means that traditional software testing approaches often do not apply. Its data-driven nature also means that the data on which the model depends must also be tested, in addition to the ML model (software) itself. Secondly, ML models can in principle adapt to take into account new data, causing the model to change its response to inputs over time [33].

This report addresses how to ensure and evaluate the trustworthiness of ML models during development. Its aim is to provide guidance to designers and developers of AI systems on characteristics they should take into consideration when designing trustworthy ML models. As previously noted, it is important that the data used to train ML models is also subject to testing and evaluation. We note however that, since the focus is upon the development of the ML model, data collection and its design [51, 56] are outside the scope of this report. We choose to focus on evaluating *classification* models, which are the most commonly encountered learning tasks in ML. Extensions to either regression or unsupervised learning are straightforward [36].

Software testing procedures (including in an ML context [33]) are often divided into two categories: *validation* and *verification*. Validation is about ensuring that the user requirements are met, whereas verification is about ensuring that the functional requirements are met. In other words, validation is centred around ensuring that the system solves the right problem for the user, and verification is about ensuring that the system is built correctly according to the specifications. The user requirements will certainly need to reflect the nature of the application and the context, and then the functional requirements are traceable to those user requirements (and specify how the user requirements will be achieved). For example, the use of ML in a clinical setting might require that the model attains a high level of accuracy as specified by the clinician (user as per this scenario), which might not be needed in a less sensitive setting. In this case, validation would be testing whether that level of accuracy is achieved. The requirement to provide explanations of results is also a user requirement, but the choice of how those explanations are established is a functional requirement (i.e. defining the specific behaviours and actions that a system performs), and testing that they

are provided correctly is verification. Testing of data can also be divided into validation ("is it the right data?") and verification ("is the data right?"). The guidance in this report covers the testing of both the data and the ML model, and comprises a mixture of validation and verification approaches.

In order for an AI or an ML system to be trustworthy, several characteristics should be considered. In Section 2, we begin by discussing some basic characteristics which are essential for nearly all AI systems. These characteristics are all related to ensuring the accuracy and performance of the model. Given that these are well-established characteristics, there are universally-agreed metrics which can be used to evaluate many of these characteristics.

We proceed in Section 3 to discuss some additional trustworthiness characteristics which are important in many AI systems. To integrate AI technology into critical applications that affect people's lives, e.g. healthcare and self-driving cars, performance on its own does not suffice anymore. The need for specification of further characteristics that AI systems should possess has therefore arisen. For instance, an automated decision-making system that is to be used as part of a clinical decision-support system should not only be accurate, but should also possess the capability of explaining its reasoning to the clinician who will be using it.

It is often more challenging to quantitatively evaluate the characteristics presented in Section 3, for two main reasons. The first is that these characteristics have been introduced by the AI and ML community more recently, and ways to assess them are not yet widely accepted. The second, and probably more compelling, reason is that these characteristics are strongly dependent on the background knowledge of their (human) users. For instance, the automated decision-making system referred to above should provide an explanation of the reasoning behind its predictions. While evaluating the performance of the prediction itself is rather straightforward (for instance it either depicts a correct or incorrect diagnosis), the same explanation can be provided to two clinicians, where one thinks it is a sound and viable explanation, whereas the other believes that it is an unacceptable explanation. We also address the role that generation of synthetic data can play in a trustworthy AI system. This is a powerful tool for creating scenarios that can help in evaluating the trustworthiness characteristics of AI systems.

There are many instances where the capabilities needed to evaluate the AI systems being developed are not found in-house or where sensitivity/regulation requires that a third party provide an independent assessment. Section 4 examines the role of third-party testing in the development and deployment of AI and ML systems across diverse industries. It outlines the key benefits such as external validation provides, such as ensuring independence, supporting regulatory compliance, and driving continuous improvement, while also addressing the potential risks related to data privacy, compliance and reputational damage. Finally, it offers guidance on best practices for mitigating these risks to help organisations confidently and responsibly incorporate third-party evaluation into their AI governance frameworks. We then close by providing a few concluding remarks in Section 5.

2 Essential Characteristics of Testing Environments for AI Systems

Evaluating an AI system consists of assessing several basic characteristics, along with some other characteristics which depend on the respective system. This can typically be per-

formed within a testing environment whose primary purpose is to identify defects, ensure that the model is performing as expected, and ensure that negative side effects such as bias and/or opacity are being mitigated as much as possible. In this section, we elaborate on the former group which includes the characteristics that should be tested for nearly every AI system.

A schematic diagram of the principal phases of a typical AI system evaluation can be found in Figure 1. First of all, the input data (which covers both the training and test data) should be validated, after which the training and test phases should be assessed. In addition, depending on the context of the application, other characteristics related to the trustworthiness of the system should be evaluated, such as the ability to adapt, mitigate bias, quantify the uncertainty in the system, and provide explanations of the involved predictions. Some of these tests can be automated throughout, such as the manner in which missing data values can be detected, and using regression tests to assess adaptability. Furthermore, integration tests can be utilised to ensure a correct and smooth integration of the different components of the system, particularly between the training and test procedures. Synthetic data generation can also be used to assess issues related to bias and imbalance that might exist within the data.

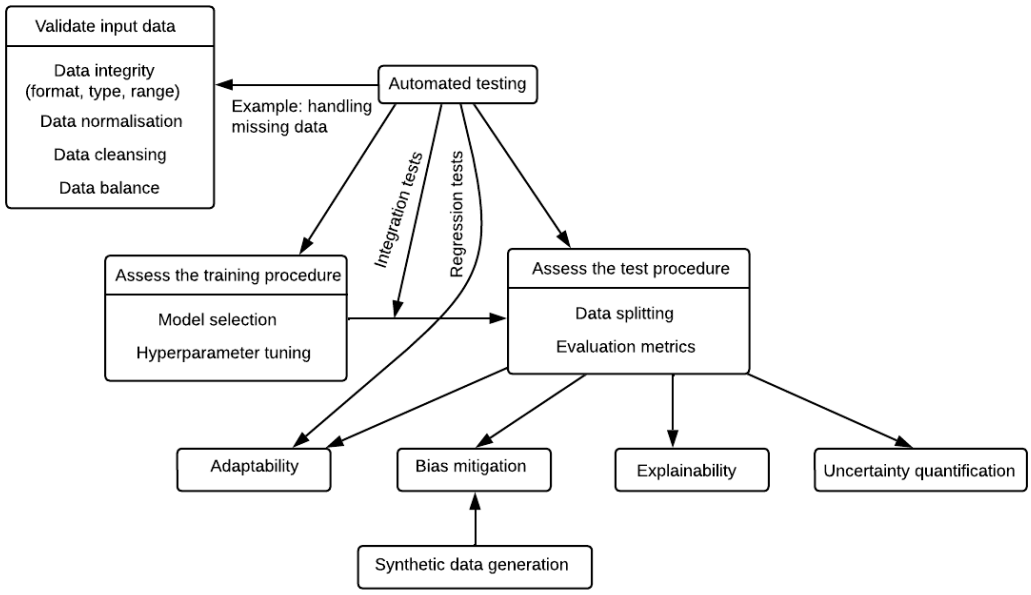

Figure 1: A schematic diagram denoting the main phases an AI system should go through for testing purposes.

This section begins by shedding light on assessing the validity of the input data which is the cornerstone of every ML model. We then move on to the assessment of the two main phases of ML model development which are the training and the test phases. We also point out that several characteristics related to assessing such models can be automated.

2.1 Assessing the Input Data

Given that AI systems learn from data, it is important to evaluate the current level of quality of the input data and assess how this impacts the decision making of the system as a whole.

This includes verifying the integrity of the data, verifying the fact that the data is bias-free, assessing how noisy the data is, and understanding the sources of the data. In addition, a data pre-processing phase [31, 58] is often required which, most notably, includes data cleansing and data normalisation.

As stated in Section 1, data collection and its design is outside the scope of this report. The aim, then, here is not to reject data that is not up to acceptable standards. Instead, the main objectives are to assess and mitigate any deficiencies that exist within the data. For instance, assessment might involve detection of missing data values, while mitigation might be a procedure where the system developers decide to either discard data examples with missing data or perform data imputation.

For example, consider a clinical decision-support system where MRI scans represent the input data, and the goal is to automate the predictions representing diagnosis of the respective scan into either healthy, Alzheimer infected, or a mild state of Alzheimer referred to as mild cognitive impairment [5]. With a scenario of this kind, assessing the input data, along with performing any required pre-processing technique will be key to the overall performance of the system. This can involve checking whether the MRI voxel values are within the correct range and applying any normalisation needed to adjust the voxel values accordingly. Importantly, the aforementioned classes can be massively imbalanced. One potential reason for this can be that people do not usually have a scan unless there is a reason to do so, and consequently it can well be the case that the input data contain many more scans belonging to the infected classes than the healthy class. This assessment would also recommend some strategies that can be used to address this imbalance.

2.1.1 Integrity of the Input Data

Checking the integrity of the input data is important to verify that the input data meets the learning criteria [10]. This means that the data should be verified for its format, type and range. Data type checks signify checking that every data column is of the type that the AI/ML algorithm expects. Data range checks are meant to inspect the numerical, categorical or text ranges of each data field to ensure they all fall within the expected range and flag any potential violation of these ranges.

The data integrity phase is essential not only for preventing eventual errors, but also to mitigate inconsistencies that can appear with ML algorithms (i.e. not necessarily in the form of an error or a bug). The latter issue can well be more damaging since the respective algorithm would still give a result with the underlying inconsistencies possibly going unnoticed.

2.1.2 Bias in the Input Data

In some applications, mitigating bias can be a necessity [49]. Actions taken to mitigate bias can consist solely of data balancing mechanisms. However, the need for more advanced actions can arise in some applications where bias resulting from sensitive attributes (e.g. gender or race) should be mitigated as much as possible. It is also worth noting that bias cannot be eliminated completely from the data in AI and ML. As such, the general aim is more about mitigating bias, rather than completely eliminating it. In a testing environment, the main task herein is to identify and quantify bias, and then the model's developer can take action based on this quantification to mitigate bias.

MITIGATING UNDER-REPRESENTATION VIA BALANCING TECHNIQUES In AI and ML, the training dataset should represent a balanced representation of the real-world population that the model aims to learn from [11, 42]. In cases where certain data groups are under-represented, data balancing techniques should be used. A notable example is classifying legitimate emails versus spam emails. Given the fact that most datasets typically contain many more legitimate emails than spam emails, this is an example of imbalanced data where the legitimate emails depict the majority class whereas the spam emails represent the minority class. Techniques like oversampling the minority class [26], or undersampling the majority class [18], can help improve the balance levels of the data in this case.

ELIMINATING BIAS RESULTING FROM SENSITIVE ATTRIBUTES This is a more complex procedure which is generally required in specific applications. An example of where this is needed is when AI systems are used to automate the prediction of recidivism for those released from prison [6]. The original problem here was that numerous false positives belonged to non-white people, which has been considered a sign of discrimination within the automated decisions made by the respective AI system. To that end, choosing the sensitive attribute as race and then aiming to eliminate bias has been a hot topic for research in ML over the last decade [6, 7]. The issue is not as simple as removing the sensitive attribute from the data since the prowess of deep models enables them to infer the values of such attributes with high precision in cases when a superficial removal is applied. For instance, deep models working on this prediction problem are capable of inferring information about race with high precision (even when the race attribute is removed), mainly from the neighbourhood and address. There are different ways in which systems can evaluate this type of bias. The most commonly used metric for such a purpose is referred to as disparate impact [24]. Roughly speaking, disparate impact refers to the concept that a model can disproportionately have a negative impact on one particular group of people (for instance based on race or gender), which can be seen as a form of (rather unconscious) discrimination. We will elaborate further on how to deal with this type of bias in Section 3.

2.1.3 Data Normalisation

This is an important pre-processing step, particularly with deep models [14]. Data normalisation typically involves scaling numeric data attributes within a common range, for instance between -1 and 1, between 0 and 1, or else within a pre-identified range that better fits the context of the data. The main potential advantage of adjusting the data ranges in such a manner is to mitigate the negative impact on the eventual ML training procedure, which can happen when large-valued attributes disproportionately dominate the learning process. The domination by large values can have a massively negative impact on learning performance.

2.1.4 Data Cleansing

Similarly to the data normalisation phase, data cleansing is a key step in order to prevent the training procedure of ML algorithms from having inconsistencies and to improve their accuracy and reliability. Data cleansing refers to identifying the data examples which contain errors, as well as identifying the appropriate (as per the learning problem at hand) manner of dealing with such erroneous entries. For example, dealing with data examples with missing values [21] can either involve removing data examples with missing attributes, or can involve data imputation [4]. Other cleansing issues include removing duplicates and handling outliers [22].

2.2 Assessing the Machine Learning Model

This assessment mainly consists of two phases: assessing the training phase of the respective ML model, and then assessing its test phase.

2.2.1 Assessing the Training Phase

The first phase in assessing the ML model is to ensure that the training phase of the ML model is proceeding as it should. This phase is centred around validating the obtained results, and ensuring that the resulting prediction reflects the aims of the model in the context it was designed for.

To assess the validity of the training phase, several characteristics should be evaluated. This begins with evaluating the quality and suitability of the training data as illustrated in detail in Section 2.1. Other important characteristics related to assessing the training procedure are described next.

Data quantity In Section 2.1, we have discussed issues relating to the quality of the training data. Here we also argue that data quantity is another important consideration. The minimal acceptable size of the training data depends on the complexity of the learning model as well as the context of the learning task at hand. Deep models need vast amounts of training data in order to have a viable opportunity of learning the hidden correlations between the input data features through the different layers of the network. Insufficient data can lead to underfitting, which refers to a model that is too simplistic to capture the underlying patterns that exist within the training data [44]. Furthermore, small sizes of data can as well exacerbate overfitting if the (deep) model tends to memorise the few training data examples it has encountered in its quest to learn a generalised concept about the data. Overfitting refers to the undesirable phenomenon in which a model learns to perform well on the training data in a way which fails to generalise to unseen data [3]. As such, consideration should be given to ensuring that there is sufficient data for the task in hand such that both overfitting and underfitting can be mitigated.

Model Selection The aim of this process is to choose the most appropriate machine learning model for the given task. This process involves identifying the best algorithm as well as the best modelling architecture. The most suitable model can be selected by considering the nature of the problem to be solved and the characteristics of the available data. An example of such a process is the reasoning that should be adopted to choose between decision trees and deep models. This is a choice that depends on the priorities set by the given task. Decision trees are more interpretable; they are therefore a good fit for rather simple tasks where understanding the dynamics of the prediction is fundamental. On the other hand, deep learning models have the potential to be more accurate. In addition, deep models are better at handling complex data. However, such advantages of deep models come at the cost of losing interpretability [34]. As such, the optimal choice involves weighing each of these factors according to the user requirements [34].

Returning to the MRI diagnosis example, the data quantity and model selection phases are key since MRI scans contain around one million voxels per scan. Deep models are very powerful and would be a good modelling fit for this learning problem in the case where vast amounts of data are available. If only moderate amounts of data are available, deep models

would be likely to overfit. Crucial modelling decisions need to be taken in such scenarios; see [5] for further illustration.

HYPERPARAMETER TUNING In most ML and AI models, the hyperparameter tuning procedure is so influential that it can make or break the whole system. Hyperparameter tuning refers to finding the best values for the hyperparameters of the model. Hyperparameters here refer to configuration settings of the model that have a direct impact on the learning process, but which are not learnt during training. Hyperparameters are typically set before the training procedure begins. Validation data is used to evaluate the model's performance on different hyperparameter settings in order to ultimately choose the best hyperparameter setting. Examples of hyperparameters include the learning rate of a deep model or number of hidden layers of a neural network. This is in contrast to the parameters (i.e. not hyperparameters) whose values are learnt by the model during the training procedure, like network weights and biases of a neural network. Hyperparameters have a significant impact on the performance of the model [48].

Performing hyperparameter tuning can potentially lead to improved accuracy and better performance of the overall system. In addition, tuning the hyperparameters can lead to a more efficient system via improving its convergence properties. Approaches that can be used for hyperparameter tuning include grid search where all the possible combinations of hyperparameter values (usually within a specific range) are exhaustively tested. This approach leads to an optimal solution, but it can be infeasible due to speed and computational constraints [8]. Another approach, which is less computationally intensive, yet is not guaranteed to reach the optimal solution, is random search where hyperparameter values are randomly sampled from a specified distribution, and then those that lead to the best performance on a validation set are ultimately selected [12]. More sophisticated approaches include Bayesian optimisation where a probabilistic model is established to guide the search process towards finding optimal values of the hyperparameters [59].

2.2.2 Assessing the Test Phase

The second principal phase of assessment of ML models addresses the test phase. It involves splitting the available data into training, validation and test sets, and then using metrics such as accuracy, precision and recall to evaluate the model's performance. It also involves evaluating how well a model generalises to unseen data [15].

DATA SPLITTING This process refers to dividing the dataset at hand into (a maximum of) three subsets, which are the training, validation, and test sets (partitions). This process aims to evaluate the ability of a model to generalise to unseen data, and to provide a reliable way of evaluating its overall performance, by ensuring that the data on which the model is trained is different from the data used for testing [38].

The training set is the partition of the dataset which is used to train the model. The validation set is used to tune the hyperparameters, and is used to evaluate the model's performance during training to ultimately prevent undesirable issues such as overfitting. Overfitting is typically identified when accuracy of the model on the training set is quite high, whereas the corresponding performance is low on the validation set. Validation is often, yet not always, needed, depending on how hyperparameters are optimised and tuned. Finally, the test set is used to evaluate the final performance of the model on data that have been completely

unseen during training (and validation), so that an unbiased estimate of the generalisation ability of the model can be obtained.

EVALUATION METRICS There are many quantitative metrics that can be used to evaluate the performance of a supervised learning model on a given task [39]. Note that how such metrics are interpreted depends on the context and on what is meant by being fit-for-purpose. The following provides some of the most commonly used evaluation metrics in binary classification [37]. Extensions of each of these metrics to multi-class classification are also possible; see for example [27].

- **Accuracy**: The proportion of correctly predicted instances out of the total instances.
- **Precision**: The proportion of true positive predictions among all of the positive predictions.
- **Recall**: The proportion of true positive predictions among all actual positive instances.
- **F1-score**: This is computed as the harmonic mean of precision and recall, which provides a balanced measure between the two.
- **Confusion matrix**: This is reported in the form of a table that summarises the classification performance of a model by displaying the true positives, true negatives, false positives and false negatives.
- **ROC Curves**: The Receiver Operating Characteristic (ROC) curve plots the true positive rate (also referred to as recall, as noted above) against the false positive rate.

2.3 AUTOMATED TESTING

Automated testing of AI models (and their software) aims to evaluate whether the models are working as expected, and to monitor the reliability and performance levels of the models as they evolve. Automated testing can as well involve using software tools to execute prescripted tests [52]. Automated testing can be crucial for performance evaluation of an AI model. Furthermore, automated testing ensures a consistent and reproducible outcome of the testing process.

Many of the automated testing tools described in this section apply to generic software systems, and so in that sense are not specific to AI systems. However, in what follows we describe the specific form that these tools take in an AI context. Evaluation tasks that can be automated in an ML context include measuring the accuracy of models, verifying that they perform as expected, and ensuring their predictions are both accurate and consistent. Automated testing can help confirm model accuracy, detect software bugs during the training process, and evaluate robustness by testing different versions of a model under various inputs and scenarios. Additionally, automation can support data validation, ensuring that the training data is of high quality and suitable for developing reliable models.

There are different forms of automated testing of AI systems, among which we highlight the following:

- **Unit tests**: They are one of the most commonly used forms of automated testing [55]. They usually focus on verifying the functionality of individual components within the model pipeline, such as data pre-processing, hyperparameter tuning or the training

procedure. An example of a unit test within an AI model is one that tests a data pre-processing function in order to ensure that missing data values have been handled correctly, i.e. in the way planned by the system's designer". (for example imputation, removing examples with missing values, etc).

- **Integration tests**: They test the interaction between different components of the system's pipeline [35]. In other words, integration testing verifies that different parts of a model, such as data pre-processing, the training phase, and the test phase, work together in tandem. Thus, integration testing is important for ensuring that the entire pipeline functions as intended, and to verify the end-to-end functionality. With (non-deterministic) ML models, it is also essential to take into consideration their stochastic nature, and to ensure that specific ML pipelines can be replicated (for example using random seeds).

- **Regression tests**: Suppose that a system has been tested and verified, and that a new functionality then needs to be added at a later date. How can we relate the updated status of the system to the one that was previously tested and verified? Regression testing ensures that the changes caused by the added (i.e. new) code do not negatively impact the already existing functionality. This can involve comparing performance metrics of the updated system with the previous version of the system. In more general terms, as AI systems evolve, regression tests can be used to monitor their performance and identify the loss of any functionality that was previously attained by the system [47].

Automated tests can lead to faster testing cycles and reduced maintenance effort and cost. Moreover, they improve the reliability and consistency levels of the overall testing process, even with new data and/or coding updates.

Having said that, it is worth noting that there can be a privacy risk with automated testing [41] which should be inspected meticulously prior to the adoption of any powerful automated testing tool. Automating the testing process entails higher risks of adversarial attacks, since the latter consider the absence of humans as an opportunity for systemising attacks that can target any loophole in the automated testing process (if any) to gain further information about the system.

2.4 Uncertainty Quantification

ML classifiers are always to some extent uncertain about the predictions that they make. Uncertainty can arise from various sources, including uncertainty in the training data (both input data and output labels) and uncertainty about the optimal classification model. It is vital in some applications that a trustworthy ML classifier is transparent about the uncertainty in its predictions. In the diagnosis of disease in healthcare, for example, information about the degree of doubt in an ML classification is crucial for informing a clinician's decision-making.

When evaluating the performance of ML classifiers, attention is often given to metrics which treat the model output as being a single predicted class. Various metrics which assess performance on this basis were outlined in Section 2.2.2. However, it is also important to transparently report the degree of doubt in a prediction by assigning probabilities to each of the classes. Many ML classifiers are inherently probabilistic. In neural networks, for exam-

ple, class probabilities can be directly optimised using loss functions such as the categorical cross-entropy.

Uncertainty quantification is also a valuable development tool: it can be used to detect regions of the data space where the model is performing poorly, providing useful information on how the model might be improved [19].

2.4.1 Evaluating Uncertainties

It is also important that the class probabilities returned by a classifier are reliable. Various tools have been developed in the ML community for evaluating the output of probabilistic ML classifiers. We highlight one popular approaches for uncertainty evaluation: *calibration analysis*.

Calibration metrics compare the probabilities returned by an ML classifier with observed proportions, which requires additional labelled data that has not been used to train the ML classifier in the first place. This comparison is typically carried out by grouping together predictions with similar prediction probabilities; see [53] for a review of state-of-the-art methods.

We also refer the reader to [43] for an overview of uncertainty evaluation toolboxes that have been developed.

3 Additional Trustworthiness Characteristics

In this section, we turn to some additional trustworthiness characteristics for which the requirements depend upon the application context. This includes evaluating how the model would fare if the environment is (slightly) changed, which is a realistic scenario in a real-world setting. Other related issues include assessing the model's out-of-distribution capabilities, which refers to whether the model can detect data examples that do not belong to the same environment that it has been trained on. As mentioned earlier, the wide range of problems to which AI systems are applied means that there is sometimes a need to ensure that the automated predictions produced by such systems are not biased against certain groups of people. Furthermore, any automated decision-making systems producing such predictions should be capable of explaining their reasoning, since providing 'black box'-like predictions solely under the claim that they are accurate does not suffice, particularly in certain sensitive applications of AI, for example medical diagnosis and self-driving cars.

We next elaborate further on some important trustworthiness characteristics of AI systems.

3.1 Adaptability

Adaptability of AI systems refers to a system's ability to adjust its behaviour as a response to new data, or to changes in the modelling approach [23]. A system that is more adaptable is one that can maintain or improve its performance even when the underlying data distribution is shifted or when the modelling approach is varied (within reasonable limits). Evaluating the capability of AI systems to adapt is crucial, particularly for models deployed in real-world applications where the data distributions keep evolving over time. Consider the previously-mentioned MRI scenario and suppose that all the originally available MRI scans belonged to people from a particular region of the world. It might be the case that the model needs

some adaptation or tuning prior to applying it to people from a different part of the world who possibly possess rather different neurophysiological characteristics.

In order to address the aforementioned variations, the corresponding AI system should adopt a mechanism that allows it to continuously update itself. We next shed some light on some common techniques and algorithms that enable different forms of continuous adaptation:

- **Active learning rates**: The learning rate in deep models is a hyperparameter that controls the extent to which the model weights are updated during each iteration of the training procedure. A high learning rate value typically means the weights are updated more aggressively, potentially leading to faster convergence but also at the risk of continuously oscillating around the optimal solution without having the ability to ultimately capture it. On the contrary, lower values of the learning rate lead to a slower, yet possibly more stable, convergence. Finding the right balance is crucial in order for the learning process to be effective. The learning rate can be learnt as a hyperparameter, prior to the beginning of the training procedure. However, techniques have also been developed for adjusting the learning rate in an adaptive way [29, 50], depending on the data. This approach can provide some level of adaptation, especially when changing an already running algorithm, or when developing more sophisticated adaptation modules is not feasible.

- **Transfer Learning**: In AI and ML, transfer learning is a paradigm that reuses knowledge gained from one task to improve performance on another task that is related, yet not identical [57, 61]. Transfer learning typically involves first training a model on a task, and then rather than training another model from scratch on another (related) task, it then adapts the already existing model such that it becomes a fit for the latter task. This approach can lead to significant reductions in computational run-time, resources needed for training, as well as the amount of training data (since there is less data now needed to train the related task(s)).

- **Online Learning**: Online learning can also be referred to as incremental learning [28]. It is a machine learning paradigm where a model is continuously updated as new data arrives. Unlike traditional forms of learning where the training procedure is performed at once (or in batches) on the available training data, online learning allows the model to adapt to variations within the data distribution without having to retrain the model from scratch by processing the upcoming data at its disposal as soon as further data become available.

  An example of a scenario where an online learning approach is a particularly good fit is when encountering streaming data (e.g. sensor data). Other examples of applications of online learning include weather forecasting, where models are continuously updated with real-time data to improve predictions, and reinforcement learning, where agents are trained to interact with dynamic environments in such a way that optimal actions can be learnt.

3.1.1 Evaluating Adaptability

Evaluating the adaptability of AI systems refers to assessing their ability to adjust to changing conditions in the modelling approach or in the data distribution. This can be done by means of one of the following techniques:

- **Comparing the model performance on more than one dataset**: This can be done by training the model on one dataset, and then testing its performance on one or more datasets belonging to a different distribution or possessing different characteristics. Care should be taken to ensure that there is some form of similarity between the datasets, otherwise there is no feasible opportunity for adaptation. In the case where the model is capable of achieving nearly the same levels of accuracy and performance across these diverse datasets, then this is a model that can adapt. Otherwise, i.e. if the performance degrades significantly, then the adaptability of the respective model is far from optimal.

- **Observing model performance under different learning conditions**: Learning conditions could refer to either the training data or the modelling approach. It is not always possible to examine how a model would react to changes in the learning conditions. However, if this is possible, it is a useful way of checking its adaptability. For example, evaluating the impact of changing the distribution of the training data on the performance of the model can provide further insights into the regions of the input space in which the model struggles to learn the most, as well as its overall adaptability.

Adapting AI systems is not always possible. It can ultimately be infeasible to adapt an AI system that is already effective and performing optimally. In such cases, what matters most is to be aware of the limitations of the respective model. This awareness can provide the system owners and users with other measures that can mitigate the impact of such rigidity. For instance, the model can be disabled or rendered ineffective in cases where rigidity is expected to be harmful, and can then operate normally otherwise.

3.2 Bias Quantification

Bias is a critical challenge in AI testing. It arises from imbalanced datasets, inappropriate training processes, or biased-inducing algorithms. Bias can lead to unfair or discriminatory decisions, especially in high-stakes applications such as hiring, lending, or law enforcement. Concerns about underlying bias and unfairness often occur in the context of automated decision-making. In this setting [6, 62], fairness means discriminating against a particular group of people due to sensitive group characteristics such as gender or race. Concerns about unfairness are of paramount importance in applications like predictive policing [13], recidivism prediction [17] and credit scoring [32].

Given the scope of this report, we focus especially on algorithmic bias, which occurs when the algorithm and the respective model induces bias in data processing and decision making.

Bias should first be detected and evaluated. Some statistical metrics have been introduced in the literature:

- **Disparate impact**: This metric is centred around the idea that a model can negatively impact a particular group of people (e.g. a particular gender or race) much more than

other groups [54]. It therefore measures the negative impact of the model on each group and compares them to one another. This is the most commonly used metric to evaluate bias in ML and AI.

- **Demographic parity**: Suppose that there is a model which automates the admission decisions for a certain school. A positive outcome refers to the person being admitted, whereas a negative outcome refers to rejection. Demographic parity measures wether the same approval rate for a particular sensitive attribute (e.g. male and female applicants). According to demographic parity, a positive outcome should be produced at equal rates to for each gender [30].

3.3 Explainability

Explainable AI (XAI) has become a necessity in many AI technologies, ensuring that stakeholders understand how decisions are made. Public trust in AI systems and the way they are used also heavily relies upon the ability of AI models to explain their decisions. In order for AI to be effectively deployed for sensitive applications in a secure and ethical manner, it is imperative for the automated decision-making systems to provide a level of explanation for each and every sensitive decision taken therein. Two key aspects of explainability are transparency and interpretability. Transparency involves documenting the system's inner workings, while interpretability focuses on making the model understandable to its users.

Explaining the decisions made by AI systems includes understanding how, when, and why the algorithm is applied, the underlying data driving its decision-making processes, and the methods employed for data collection, processing, and interpreting results. Examples of explaining the predictions obtained by ML models used for MRI diagnosis can be found in [5, 63]. These techniques base their explanations on providing saliency maps where the voxels which are the most salient for the respective prediction decision are highlighted.

3.3.1 Adapting Explainability to be Fit for Purpose

Explainability is crucial for bridging the gap between ML models and human consumers, and in addressing cognitive biases. Explanation methods also need to adapt to the diverse needs of consumers. This includes varying the explanation according to its receiver, whether this is the affected user, decision maker, or regulator. For example, an explanation of a medical diagnosis decision which is directed at a clinician could include some minimal level of statistical reasoning since it can be assumed that the clinician possesses some (moderate) level of knowledge about statistics. On the other hand, an explanation of the same decision directed at the patient cannot be based on the same assumption [46], and must therefore be limited to concepts which are easy to understand by the public.

It is important to remember that every explanation of an ML system aims to serve a beneficial purpose, even though what constitutes "beneficial" can vary. Different benefiting organisations, whether governmental, private, or social, have their own objectives behind adopting a given ML system. The purpose of the explanation should be indicated accordingly. As such, explanations are context-sensitive since every explanation is aiming to answer a set of questions, and such questions depend on the context. The context-sensitive nature of explainability renders it difficult to generally recommend solely one explainability approach that should be ideal to deploy under all circumstances.

3.3.2 Evaluating Explainability

Evaluating explainability refers to assessing how well the predictions of an ML model can be understood and interpreted by humans. On a high level, as noted earlier, explainability is an extremely human-oriented concept. In addition, it has been introduced into the ML literature rather recently. Given these two reasons, there are no universally agreed-upon quantitative metrics of explainability in the ML literature. As such, the bottom line, as far as its evaluation is concerned, is to somehow assess the level of the satisfaction of the targeted human (i.e. the human receiving the explanation) with the provided explanation [20].

The process of evaluating explanations should aim at generically unifying such a level of satisfaction in a consistent manner by basing it on the level of complexity of the provided explanation, with simpler explanations being preferred. This is based on the reasoning that a good explanation should be concise, more focussed, and easier to understand. For example, an explanation of image classification prediction should be as compact as possible, i.e. containing a small number of pixels.

3.4 Importance of Synthetic Data in Tailoring Certain Scenarios for Testing Purposes

Generative models [25] first learn the underlying structure of a real training data set, then use that knowledge to generate synthetic data with the same characteristics. Generative models often adopt a probabilistic approach where the probability distribution of the data is learnt such that data examples that are plausibly similar to the real data can be synthetically generated. Note that the synthetic data will only ever generate examples using the data structure and probability distribution identified during training.

Using generative modelling can enrich an existing data set with new synthetic examples which can (together with the existing real data) better cover the spectrum of the data space. The improved coverage of the spectrum of the data space can in turn lead to better performance over downstream tasks such as testing the trustworthiness of the underlying model. One of the principal advantages of the generative modelling approach is the ability to synthetically create controlled scenarios that do not necessarily have to be satisfied within the real data (particularly with medium-to-small amounts of real data). Such scenarios can then play a pivotal role in gaining further insights about evaluating the trustworthiness of the underlying model.

In addition, synthetically generated outputs can be useful for the automation of testing. That is, instead of manually creating a number of test cases on which models are tested, large amounts of synthetic data can be generated and then passed to the models. Furthermore, being able to generate large volumes of data is useful for training models such as classifiers. In particular, if there is a class-imbalance problem, or simply a small amount of data, being able to generate additional data stands to ameliorate some of the associated troubles. Appendix A describes an example of this process in more detail.

4 Third-Party Testing

Different sectors and industries are developing AI capabilities at varying rates. For developers of AI and ML systems, seeking external validation of the key trustworthiness characteristics of their technologies can be highly beneficial. This is often necessary to meet

safety or regulatory requirements, or to access expert advice not available in-house. The sectors most likely to engage in third-party testing tend to be those with stringent regulation due to their safety-critical nature and higher expectations for data integrity and decision-making transparency. Industries such as healthcare, security, pharmaceutical manufacturing, aerospace, and finance, as well as emerging areas such as autonomous vehicles, biologics, and renewable energy, are examples where quality, traceability, and trust are essential. In addition, small and medium-sized enterprises (SMEs) and manufacturers that are newly adopting AI or digital technologies may also require external validation to build confidence in autonomous decision-making systems.

Third-party testing can offer value throughout the development and deployment of AI/ML systems. One of the most significant advantages is the independence and impartiality that third parties bring. Internal evaluations, while useful, may be subject to organisational bias or internal pressures. Independent evaluators, with no vested interest in the outcome, can deliver credible and objective assessments, which foster trust among stakeholders and strengthen the legitimacy of results. This level of impartiality is especially valuable in sectors where transparency and accountability are critical. In many regulated industries, third-party testing is not only advantageous but required for certification, accreditation, or legal compliance. These assessments offer formal evidence that an organisation is meeting the relevant standards, legal requirements, and sector-specific expectations. This can reduce the risk of legal liability and enable smoother interactions with regulatory authorities. Third-party experts often bring specialised knowledge that may not be available internally. Their feedback can highlight overlooked risks, technical limitations, or areas of inefficiency, and these insights can inform improvements in model robustness, data quality, interpretability, and governance.

While third-party testing offers clear advantages, it also carries risks that must be managed carefully. Data privacy is a primary concern, as sensitive information may be exposed through poor handling or security breaches. Ensuring secure testing environments and limiting data access is essential. There are also compliance risks if the third party fails to meet legal or regulatory requirements, which can lead to liability for the organisation. These risks can be mitigated through due diligence, strong data governance, and regular oversight of third-party partners.

In conclusion, third-party testing plays a vital role in ensuring trustworthy, safe, and reliable AI systems. When managed responsibly, it not only supports compliance and continuous improvement but also builds competitive strength and reinforces confidence across a broad spectrum of stakeholders.

5 CONCLUSION

We have discussed several trustworthiness characteristics important in the evaluation of AI systems, particularly those that are to be adopted in sensitive, societal applications. Assessing such systems is a necessity for the following reasons: (i) the assumption that such systems are usually powerful and can therefore invoke actions with wide-reaching impact, and (ii) the sensitivity of the corresponding high-stakes application.

We have also touched on fundamental characteristics that should be satisfied in nearly every AI system, such as evaluating the validity of the input data (on which the systems base their learning process). We have also shed light on characteristics which should be particu-

larly inspected in AI applications with a sensitive nature such as healthcare and self-driving cars. Such characteristics include bias, explainability, uncertainty quantification and the ability of the respective system to adapt to variations and/or ongoing changes in the learning environment. Some of the latter characteristics have rather recently been introduced to the literature, and are therefore less established than the performance-based characteristics. For example, unlike accuracy and performance, which can be evaluated via universal, well-defined, and agreed-upon quantitative metrics, explainability is a notion which strongly depends on the receiving end of the process. This includes the human using and/or owning the system, the context of the problem, and the main reasons why an automated solution to such a problem is adopted in the first place. Finally we have also discussed how synthetic data generation can be utilised to evaluate some characteristics related to interactions between attributes of the data, and the potential impact of such interactions on trustworthiness.

Our report complements and extends standards and guidance documents on testing and evaluation of AI systems. We provide detailed guidance which elaborates on standards documents such as [1] and [2], while at the same time widening the scope of trustworthiness characteristics in comparison with other more detailed guidance documents such as [9], [45] and [16].

In addition to acknowledging the importance of the standard requirements for ML models such as accuracy and being able to generalise to unseen data, we recommend dedicating further attention to domain-specific requirements. For example, the designer of an ML model that is expected to be eventually used in rather different environments to the one it has been developed in should ensure that the model possesses adaptation capabilities such that it can adaptively function within the new environments. As another example, an ML model that is to be used in automating decisions of admitting students to a particular school should be rigorously checked to ensure that the corresponding admission decisions are fair and are not biased towards a particular race or gender. Focussing on mitigating domain-specific risks is an important factor for establishing trustworthy ML models.

In order to document the results of these assessments in a standardised way, we recommend the adoption of a structured format which begins with specifying the overall purpose of the system and of the evaluation. This should be accompanied by defining the scope as well as the functionalities that are being evaluated. Afterwards, the results of the quantitative and qualitative metrics should also be provided. Finally, general conclusions and recommendations of the main flaws that the system should potentially address in the near future should also be highlighted.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the UK Department for Science, Innovation and Technology, which funded this work as part of the AI Standards Hub programme. We would also like to thank Peter Harris for his insightful review and thoughtful comments.

A AN EXAMPLE OF SYNTHETICALLY GENERATED RESULTS

In the following, we provide an example of a scenario where the generated data can help us understand the nature of a potential bias issue within the real data. The dataset we are working on here is a customer churn dataset. This is a dataset which addresses the banking industry issue of customer retention. More precisely, it aims to predict which customers persist in using the services of a particular bank, and which customers depart from using the service and move on to another bank [40]. This is a crucial issue for banks since it affects revenue, reputation, etc. Hence, being able to predict churn efficiently can help the respective bank develop strategies which are particularly tailored towards those who are more likely to discontinue using the bank services in the near future such that the likelihood they would remain can increase.

This is a binary classification problem where the label referred to as "Exited" can either have a value of 1 (denoting a customer who is leaving) or 0 (denoting a customer who remains). The data contains several input features depicting information about the current customers of the bank [40]. Some of these features are uncorrelated with the churn label like the customer ID and the customer's surname. On the other hand, several features have a clearly defined impact on the churn label. Examples include the credit score variable where a customer with a higher credit score is less likely to leave the bank, the tenure variable where customers who have been more loyal (i.e. have been using the bank services for a larger number of years) are less likely to leave the bank. Other influential indicators of the churn label include the balance variable (the higher the balance the less likely it is that the respective customer would leave), the credit card variable (customers with a credit card are less likely to leave), estimated salary (higher salaries clearly indicate less likelihood of leaving), and whether or not the customer is an active member (active members are less likely to eventually leave). There is however a need to further explore the impact of a few other features, namely the customer's gender, on the churn label.

In the churn dataset, some correlation was noticed between the gender attribute and the churn label, where female customers are more often predicted to leave the bank service [40]. More precisely, out of those who end up leaving the service, 44% are men, whereas 56% are women. On the other hand, out of those who remain with the bank service, 57% are men, and 43% are women. In order to test whether this imbalance is something that can be straightforwardly fixed, or else whether there is some correlation that, at least as far as this dataset is concerned, exists between the attribute and the label, we have adopted a generative modelling approach to generate synthetic data.

In Table 1, we display examples of the results of synthetic data generated based on a conditional generative adversarial network [60]. This is a generative model that generates data conditioned on specific attribute values. We have generated data conditioned on both values of the churn label, as well as conditioned on several other values of other attributes. The results have confirmed a similar trend regarding the correlation between the gender attribute and the churn label, where women are more likely to leave the bank service than men. Similar to the ratios noted above for the real data, for the synthetic data: Out of those who end up leaving the service, 43% are men, whereas 57% are women, and out of those who remain with the bank service, 56% are men, and 44% are women. As such, this demonstrates that this is not an issue where imbalance can be straightforwardly fixed. The correlation between both attributes would either require collecting more real data (possibly over other banks and/or different periods of time) to further check this imbalance, or else a more dedicated analysis to understand the reasons behind this correlation.

Table 1: A sample displaying records of the churn data that we have synthetically generated based on conditional generative adversarial networks. Out of those who end up leaving the service, 43% are men, whereas 57% are women, and out of those who remain with the bank service, 56% are men, and 44% are women. This demonstrates a similar pattern to the real data where there is a a correlation between the gender and label attributes.

CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
654	Germany	Female	44	9	146450.9	3	1	1	141247.04	1
799	France	Female	33	9	120390.8	1	1	0	153445.09	1
735	Spain	Female	38	8	13386.72	2	1	0	88385.41	1
741	Spain	Male	38	2	133491.9	1	1	0	41344.2	1
782	France	Male	34	5	0	2	1	1	199992.48	0
774	France	Male	38	4	1740.7	1	1	0	25910.98	0
657	Spain	Male	42	10	0	2	1	0	124481.64	0
701	Spain	Female	41	6	159614.2	1	0	1	11.58	0
663	France	Female	39	8	2821.45	1	0	1	11.58	0
685	Germany	Female	43	2	122090	1	1	0	82470.2	1
658	Germany	Female	47	5	83687.52	2	0	1	13477.65	1
846	France	Male	35	9	0	1	0	0	88351.31	0
785	France	Male	40	9	765.35	2	1	0	11.58	0
768	France	Male	33	4	0	2	1	0	85740.6	0
746	France	Male	45	4	0	2	0	0	163270.21	0
827	France	Male	54	9	57289.76	1	0	1	127520.8	0
691	France	Male	26	4	0	2	1	1	75175.61	0

REFERENCES

[1] ISO//IEC TR 24028. “Overview of trustworthiness in artificial intelligence”. In: (2020).

[2] ISO//IEC TR 29119-11. “Guidelines on the testing of AI-based systems”. In: (2020).

[3] T. Adel, S. Bilson, M. Levene, and A. Thompson. “Trustworthy artificial intelligence in the context of metrology”. In: *Producing Artificial Intelligent Systems: The roles of Benchmarking, Standardisation and Certification*. Ed. by I. Ferreira. Computational Intelligence. Cham, Switzerland: Springer Nature, 2024.

[4] T. Adel and C. de Campos. “Learning Bayesian networks with incomplete data by augmentation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31 (2017).

[5] T. Adel, T. Cohen, M. Caan, and M. Welling. “3D scattering transforms for disease classification in neuroimaging”. In: *Neuroimage: Clinical* 14 (2017), pp. 506–517.

[6] T. Adel, I. Valera, Z. Ghahramani, and A. Weller. “One-network adversarial fairness”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (2019).

[7] A. Agarwal et al. “A reductions approach to fair classification”. In: *International Conference on Machine Learning (ICML)* (2018).

[8] H. Alibrahim and S. Ludwig. “Hyperparameter optimization: Comparing genetic algorithm against grid search and Bayesian optimization”. In: *IEEE Congress on Evolutionary Computation (CEC)* (2021).

[9] M. Alsalem et al. “Evaluation of trustworthy artificial intelligent healthcare applications using multi-criteria decision-making approach”. In: *Expert Systems with Applications* 246 (2024).

[10] S. Basodi, S. Tan, W. Song, and Y. Pan. “Data integrity attack detection in smart grid: A deep learning approach”. In: *International Journal of Security and Networks* (2020).

[11] G. Batista, R. Prati, and M. Monard. “A study of the behavior of several methods for balancing machine learning training data”. In: *ACM SIGKDD Explorations Newsletter* (2004), pp. 20–29.

[12] J. Bergstra and Y. Bengio. “Random search for hyper-parameter optimization”. In: *Journal of Machine Learning Research (JMLR)* 13 (2012), pp. 281–305.

[13] T. Brennan, W. Dieterich, and B. Ehret. “Evaluating the predictive validity of the COMPAS risk and needs assessment system”. In: *Criminal Justice and Behavior* 36 (2009), pp. 21–40.

[14] K. Cabello-Solorzano, I. Ortigosa de Araujo M. Pena, L. Correia, and A. Tallon-Ballesteros. “The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis”. In: *International Conference on Soft Computing Models in Industrial and Environmental Applications* (2023), pp. 344–353.

[15] R. Caruana, N. Karampatziakis, and A. Yessenalina. “An empirical evaluation of supervised learning in high dimensions”. In: *International Conference on Machine Learning (ICML)* (2008), pp. 96–103.

[16] J. Chandrasekaran et al. “Test & evaluation best practices for machine learning-enabled systems”. In: *Machine Learning Archive* arXiv:2310.06800 [stat.ML] (2023).

[17] A. Chouldechova. “Fair prediction with disparate impact: A study of bias in recidivism prediction instruments”. In: *Big Data* 2 (2017).

[18] D. Devi, S. Biswas, and B. Purkayastha. “A Review on solution to class imbalance problem: Undersampling approaches”. In: *International Conference on Computational Performance Evaluation (ComPE)* (2020), pp. 626–631.

[19] Y. Ding, J. Liu, J. Xiong, and Y. Shi. “Revisiting the evaluation of uncertainty estimation and its application to explore model complexity-uncertainty trade-off”. In: *Computer Vision and Pattern Recognition (CVPR) Workshops* (2020).
[20] F. Doshi-Velez and B. Kim. “Towards a rigorous science of interpretable machine learning”. In: *Machine Learning Archive* arXiv:1702.08608 [stat.ML] (2017).
[21] T. Emmanuel et al. “A survey on missing data in machine learning”. In: *Journal of Big data* (2021), pp. 1–37.
[22] H. Escalante. “A comparison of outlier detection algorithms for machine learning”. In: *International Conference on Communications in Computing* (2005).
[23] A. Farahani, S. Voghoei, K. Rasheed, and H. Arabnia. “A brief review of domain adaptation”. In: *Advances in Data Science and Information Engineering* (2021), pp. 877–894.
[24] M. Feldman et al. “Certifying and removing disparate impact”. In: *ACM SIGKDD international conference on knowledge discovery and data mining* (2015), pp. 259–268.
[25] I. Goodfellow et al. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 27 (2014).
[26] A. Gosain and S. Sardana. “Handling class imbalance problem using oversampling techniques: A review”. In: *IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (2017), pp. 79–85.
[27] Margherita Grandini, Enrico Bagli, and Giorgio Visani. “Metrics for multi-class classification: an overview”. In: *arXiv preprint arXiv:2008.05756* (2020).
[28] J. He, R. Mao, Z. Shao, and F. Zhu. “Incremental learning in online scenario”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 13926–13935.
[29] S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International Conference on Machine Learning (ICML)* (2015).
[30] Z. Jiang et al. “Generalized demographic parity for group fairness”. In: *International Conference on Learning Representations (ICLR)* (2022).
[31] M. Kang and J. Tian. “Machine learning: Data pre-processing”. In: *Prognostics and health management of electronics: Fundamentals, machine learning, and the internet of things* (2018), pp. 111–130.
[32] A. Khandani, A. Kim, and A. Lo. “Consumer credit-risk models via machine-learning algorithms”. In: *Journal of Banking & Finance (JBF)* 34 (2010), pp. 2767–2787.
[33] M. Levene et al. “A life cycle for trustworthy and safe artificial intelligence systems”. In: *NPL Report MS 57* (2024).
[34] Y. Luo et al. “Balancing accuracy and interpretability of machine learning approaches for radiation treatment outcomes modeling”. In: *BJR Open* 1 (2019).
[35] D. Marijan. “Comparative study of machine learning test case prioritization for continuous integration testing”. In: *Software Quality Journal* 31 (2023), pp. 1415–1438.
[36] T. Mitchell. *Machine learning*. McGraw Hill, 1997.
[37] G. Naidu, T. Zuva, and E. Sibanda. “A review of evaluation metrics in machine learning algorithms”. In: *Artificial Intelligence Application in Networks and Systems* (2023).
[38] Q. Nguyen et al. “Influence of data splitting on performance of machine learning models in prediction of shear strength of soil”. In: *Mathematical Problems in Engineering* (2021).
[39] OECD. *Catalogue of Tools Metrics for Trustworthy AI*. https://oecd.ai/en/catalogue/metrics. Accessed June 2025. 2025.

[40] A. Paradita, N. Agustiana, P. Rukmana, and P. Nelsa. “Comparative analysis of naive Bayes and K-nearest neighbors algorithms for customer churn prediction: A Kaggle dataset case study”. In: *International Conference on Information Science and Technology Innovation (ICoSTEC)* (2024).

[41] S. Pargaonkar. “Machine learning algorithms for automated software testing: A comprehensive review of current trends and challenges”. In: *The Algorithmic Odyssey – A Comprehensive Guide to AI Research* (2021), pp. 189–204.

[42] F. Pecorelli, D. Di Nucci, C. De Roover, and A. De Lucia. “On the role of data balancing for machine learning-based code smell detection”. In: *ACM SIGSOFT International Workshop on Machine Learning Techniques for Software Quality Evaluation* (2019), pp. 19–24.

[43] Maximilian Pintz, Joachim Sicking, Maximilian Poretschkin, and Maram Akila. “A survey on uncertainty toolkits for deep learning”. In: *arXiv preprint arXiv:2205.01040* (2022).

[44] S. Pothuganti. “Review on over-fitting and under-fitting problems in machine learning and solutions”. In: *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* (2018), pp. 3692–3695.

[45] O. Rainio, J. Teuho, and R. Klen. “Evaluation metrics and statistical tests for machine learning”. In: *Nature Scientific Reports* 14 (2024).

[46] C. Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. In: *Nature Machine Intelligence* 1 (2019), pp. 206–215.

[47] P. Sawant. “Test case prioritization for regression testing using machine learning”. In: *IEEE International Conference on Artificial Intelligence Testing (AITest)* (2024).

[48] P. Schratz et al. “Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data”. In: *Ecological Modelling* 406 (2019), pp. 109–120.

[49] S. Siddiqui et al. “Survey on machine learning biases and mitigation techniques”. In: *Digital* 4 (2023), pp. 1–68.

[50] T. Takase, S. Oyama, and M. Kurihara. “Effective neural network training with adaptive learning rate based on training loss”. In: *Neural Networks* (2018), pp. 68–78.

[51] K. Varshney. *Trustworthy machine learning and artificial intelligence*. The ACM Magazine for Students, 2019.

[52] C. Wan et al. “Automated testing of software that uses machine learning apis”. In: *International Conference on Software Engineering* (2022), pp. 212–224.

[53] Cheng Wang. “Calibration in deep learning: A survey of the state-of-the-art”. In: *arXiv preprint arXiv:2308.01222* (2023).

[54] H. Wang, B. Ustun, and F. Calmon. “On the direction of discrimination: An information-theoretic analysis of disparate impact in machine learning”. In: *IEEE International Symposium on Information Theory (ISIT)* (2018).

[55] S. Wang et al. “Automatic unit test generation for machine learning libraries: How far are we?” In: *International Conference on Software Engineering (ICSE)* (2021), pp. 1548–1560.

[56] T. Wang et al. “A comprehensive trustworthy data collection approach in sensor-cloud systems”. In: *IEEE Transactions on Big Data* 8 (2018), pp. 140–151.

[57] K. Weiss, T. Khoshgoftaar, and D. Wang. “A survey of transfer learning”. In: *Journal of Big Data* 3 (2016), pp. 1–40.

[58] M. Winter. “The ISTQB Certified Tester, AI Testing (CT-AI)”. In: *Softwaretechnik-Trends Band* (2022).

[59] J. Wu et al. “Hyperparameter optimization for machine learning models based on Bayesian optimization”. In: *Journal of Electronic Science and Technology* 17 (2019), pp. 26–40.

[60] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. “Modeling tabular data using conditional GAN”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 32 (2019).

[61] W. Ying, Y. Zhang, J. Huang, and Q. Yang. “Transfer learning via learning to transfer”. In: *International Conference on Machine Learning (ICML)* (2018), pp. 5085–5094.

[62] R. Zemel et al. “Learning fair representations”. In: *International Conference on Machine Learning (ICML)* (2013), pp. 325–333.

[63] L. Zintgraf, T. Cohen, T. Adel, and M. Welling. “Visualizing deep neural network decisions: Prediction difference analysis”. In: *International Conference on Learning Representations (ICLR)* (2017).