Unveiling the robustness of machine learning families

IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 035040 https://doi.org/10.1088/2632-2153/ad62ab

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
22 December 2023

REVISED
1 May 2024

ACCEPTED FOR PUBLICATION
12 July 2024

PUBLISHED
8 August 2024

PAPER

Unveiling the robustness of machine learning families

R Fabra-Boluda*, C Ferri, M J Ramírez-Quintana and F Martínez-Plumed

Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Valencia, Spain
* Author to whom any correspondence should be addressed.

E-mail: rafabbo@dsic.upv.es, cferri@dsic.upv.es, mramirez@dsic.upv.es and fmartinez@dsic.upv.es

Keywords: robustness, noise, instance difficulty, supervised learning, item response theory

Abstract

The evaluation of machine learning systems has typically been limited to performance measures on clean and curated datasets, which may not accurately reflect their robustness in real-world situations where data distribution can vary from learning to deployment, and where truthfully predict some instances could be more difficult than others. Therefore, a key aspect in understanding robustness is *instance difficulty*, which refers to the level of unexpectedness of system failure on a specific instance. We present a framework that evaluates the robustness of different ML models using item response theory-based estimates of instance difficulty for supervised tasks. This framework evaluates performance deviations by applying perturbation methods that simulate noise and variability in deployment conditions. Our findings result in the development of a comprehensive taxonomy of ML techniques, based on both the robustness of the models and the difficulty of the instances, providing a deeper understanding of the strengths and limitations of specific families of ML models. This study is a significant step towards exposing vulnerabilities of particular families of ML models.

1. Introduction

The proliferation of machine learning (ML) systems has transformed various fields, including medicine, finance, social media, and autonomous transport, integrating into our daily lives and shaping decision-making processes. With the growing influence of these systems, it is imperative to have reliable and robust ML systems that can function correctly under different conditions and inputs [1]. Robustness, in this context, refers to the ability of a ML system consistently maintain its predictions despite variations or perturbations [2].

Traditional evaluations of ML robustness have predominantly focused on resistance to adversarial examples—deliberately manipulated inputs designed to trick models into making incorrect predictions. These studies often involve the introduction of noise during training and testing phases to test the model's defences against such attacks [1]. These examples are commonly known as adversarial examples, and most research in the area focuses on measuring robustness against adversarial examples [3, 4]. However, while this focus is important, it largely overshadows another critical aspect: prediction consistency: understanding a model's stability and resilience to input variations, independent of initial training labels. This approach is motivated by the importance of consistency as a factor in assessing model robustness: a model that returns the same output for an input regardless of slight perturbations or noise potentially demonstrates a high degree of robustness against uncertainties or adversarial examples in deployment environments.

We also add an extra dimension in our analysis: the instance difficulty. Robustness is a multifaceted concept influenced by various factors [5], including the difficulty of the instance (intrinsic or extrinsic). Understanding where and why a model fails is critical to preventing unforeseen failures and improving robustness. To the best of our knowledge, this factor has not been considered as a criterion for evaluating model robustness. In this study, we investigate the uniformity of performance across different levels of instance difficulty and the sensitivity of instances to changes in the label predicted by a classifier under perturbations (such as adding noise to the input attributes). Typically, instances with higher difficulty levels are more sensitive under small perturbations.

A way of estimating the difficulty of instances is to calculate the average error of a set of systems per instance as a proxy for difficulty [6]. However, there are risks in using a population of systems, such as instability of difficulty metrics in the presence of a nonconforming system (failing on easy instances and succeeding on hard ones). To address these limitations, we utilise item response theory (IRT) [7] to infer instance difficulty from a matrix of instances and systems, giving more weight to conforming systems. IRT provides a scaled difficulty metric that follows a normal distribution and can be directly compared to a system's ability [7]. Thus, we aim to provide a comprehensive analysis of model robustness, taking into account instance difficulty and perturbation effects.

In this paper, we present a comprehensive evaluation framework to analyse the robustness of multiple ML families in a systematic and empirical manner. We factor in instance difficulty as a key consideration in our evaluation. In addition, we perform hierarchical clustering to categorise families based on their robustness. Our evaluation framework is versatile, incorporating datasets from diverse domains, a broad spectrum of ML techniques, and a randomised noise-inducing instance perturbation function. This enables its adaptability to various needs by adjusting the datasets, models, and perturbation function accordingly. To measure the robustness of a model, we compare its performance on original and perturbed test sets, taking into account the difficulty of the instances. The final measure of robustness is determined by the agreement modulo the instance difficulty. By performing hierarchical clustering, we categorise the ML families based on their robustness, providing a taxonomy that raises awareness of the vulnerabilities of different families of ML models. The main findings and contributions of this paper are:

- We highlight the importance of instance difficulty in ML robustness by demonstrating its impact on model performance and the need to consider it in robustness assessments.
- We provide a versatile evaluation framework designed to assess the robustness of ML systems across various domains and techniques, focusing on model stability and resilience to input variations.
- We unlock and deepen the relationship between the robustness of ML models and the difficulty of instances subjected to differing levels of noise.
- We introduce a taxonomy of ML families according to their ability to handle input variations and overall robustness, providing valuable insight into the strengths and weaknesses of different ML models.
- We provide a nuanced guide for practitioners on the judicious use of models in different real-world scenarios, taking into account the interplay between noise, instance difficulty and dataset complexity.

The structure of the paper is as follows: section 2 reviews the relevant literature on evaluating ML robustness in noisy conditions, estimating instance difficulty, and behavioural taxonomies of ML techniques. Section 3 outlines our methodology for analysing model robustness. Section 4 describes the experimental setup, while section 5 presents the results of our experiments. The insights and implications of our research are discussed in section 6. Finally, the paper concludes in section 7, where we summarise our contributions and suggest directions for future work.

2. Background

In this section, we revisit key concepts regarding model robustness, instance difficulty based on IRT, and behavioural taxonomies of ML techniques.

2.1. Robustness in a noisy framework

Robustness is a defining characteristic of ML systems, used to ensure the system behaves as desired when faced with changes in data [8]. One common method of simulating these changes is adding noise to the data, as real-world data often contain noise [9].

There are two types of noise [10]: class noise, which may be due to the presence of contradictory examples or instances with wrong classes, and attribute noise, that can be due to erroneous attribute values, missing or unknown attribute values, incomplete attribute or 'do not care' values. In this paper we focus on erroneus values, disregarding other sources of noise.

Working in noisy environments can degrade classifier performance [11]. An alternative to clean training data is to consider noise during training. Some approaches consider that the noise distribution is known in advance [12], but most recent approaches propose to use perturbed training data by adding random artificial noise to attributes and class labels to simulate erroneus values [13–15]. Noise introduction involves changing the values according to a source distribution, such as a Gaussian distribution or uniform random distribution for numerical attributes, or randomly changing categorical attributes [9, 16–18].

Injection of altered attribute instances into the training set can be employed for training more robust systems [19, 20] and for improving the robustness of neural networks [21], and models against adversarial

attacks [22, 23]. It is also commonly used for data augmentation in order to train more accurate classifiers [16, 24]. These noise-based techniques can obtain robust models that are relevant in some areas such as, for instance, speech emotion recognition systems [25–27]. On the other hand, label noise is used to simulate mislabelled instances or other forms of data corruption. It has been used to evaluate the robustness of computer vision models [28] and can also improve model robustness by reducing overfitting errors [29]. Perturbation techniques has been proved it can be employed for test model robustness by adding noise to the test instances. This is often used in adversarial ML, where the goal is to create adversarial examples that are indistinguishable from the original instances, but result in a change in the model's output.

The standard method for assessing robustness is to compare the performance of the model in the presence of noise with its performance in the absence of noise, regardless of whether the noise is present in the training or test data. The standard classification metrics such as accuracy and F-measure are used to measure the loss of performance after introducing noise in training. The equalised loss of accuracy metric is also used to measure the noise robustness of a classifier [11]. Other techniques for assessing robustness include mixed integer programming [30], abstract interpretation [31], and symbolic execution [32–34].

In this study, we focus on the consistency of model predictions relative to perturbed inputs. Consistency is an important factor in assessing model robustness: a model that returns the same output for an input regardless of slight perturbations or noise potentially demonstrates a high degree of robustness against uncertainties or adversarial examples in deployment environments. We also add an extra dimension in our analysis: the instance difficulty, a factor previously overlooked in evaluating model robustness. We use the kappa statistic as evaluation metric to assess the agreement between a model's predictions on original vs. perturbed data sets.

2.2. Instance difficulty

The handling of difficult instances during the development of AI systems is crucial, especially for trained models. Such instances, often associated with noise, outliers or decision bounds, can lead to overfitting or lack of convergence. While various approaches have been proposed to prevent overfitting by identifying anomalies or mislabelled instances, they do not clearly define what characterises these instances. In [35], instance hardness metrics were used to characterise the degree of difficulty of each input sample based on the empirical definition of classification behaviour. In computer vision and natural language processing (NLP) related tasks, however, the analysis is limited to global image properties (e.g. salience, memorability, photo quality, tone, colour, texture) [36–38] or lexical readability and richness [39, 40] in the case of NLP.

Tackling these difficult instances is a problem that is often addressed with domain-specific methods. In particular, IRT [7], traditionally a branch of psychometrics, has emerged as an innovative method applied to AI and ML [6, 41–44]. IRT encompasses a collection of models that link the responses of individuals to items (such as test questions) to the latent abilities of those individuals. Primarily used in educational assessment and psychometric evaluations, IRT seeks to quantify an examinee's ability through a battery of questions.

We focus on the dichotomous models where the response can be either correct or incorrect. Let's define U_{ji} as the binary response of individual j to item i, where $U_{ji} = 1$ represents a correct response, and $U_{ji} = 0$ indicates an incorrect one. The ability of the individual, denoted by θ_j, reflects their proficiency in the construct being measured. If an individual's ability matches the difficulty level of an item, there is an even chance of a correct answer. As a person's ability deviates from the item's difficulty, the probability of a correct response adjusts accordingly–increasing with a higher ability above the difficulty level, and decreasing with a lower ability.

In an IRT model, each item has its associated item characteristic curve (ICC) (see figure 1(left)) , which illustrates how the probability of a correct response varies with the test-taker's ability. For instance, in the two-parameter (2PL) IRT model, the ICC and the associated probability of getting the item correct are defined by a logistic function:

$$P\left(U_{ji} = 1 \mid \theta_j\right) = \frac{1}{1 + \exp\left(-a_i\left(\theta_j - b_i\right)\right)}. \tag{1}$$

The shape of an ICC is determined by the item's difficulty (b_i) –the parameter that dictates where on the ability scale the probability of a correct answer is 50%. If a respondent's ability θ_j equals the item difficulty b_i, there is an equal chance of answering correctly, depicted by the midpoint of the ICC.

Items are also differentiated by their discrimination parameter (a_i), which indicates the slope of the ICC at the difficulty point. High discrimination values mean even small differences in ability can translate into significant variations in the potential for correct responses. An item with $a_i = 1.0$ discriminates effectively, making slight adjustments in ability result in meaningful changes in the probability of a correct answer. The simplest IRT models, known as 1PL, assume a constant discrimination parameter of 1 for all items; only the

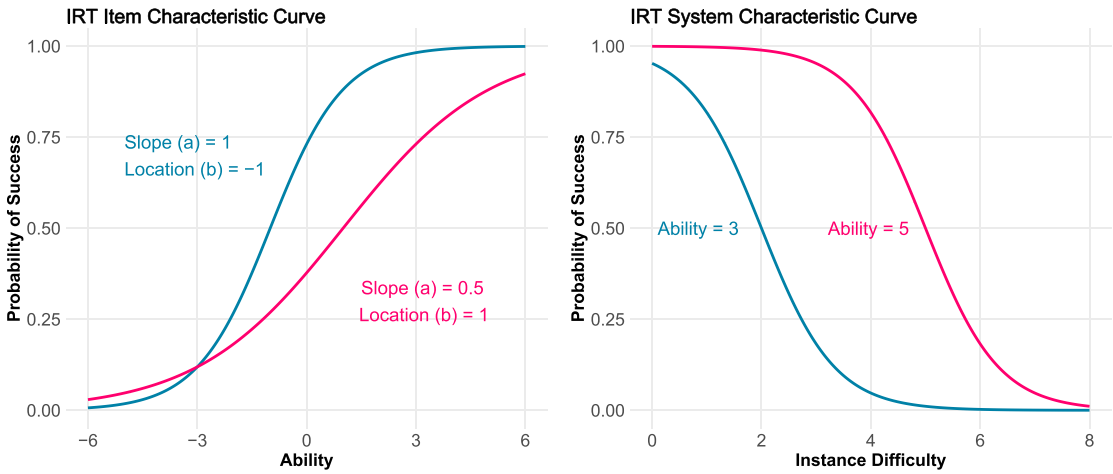


Figure 1. Left: example of a 2PL IRT ICC curve, with slope $a = 2$ in red and location parameter $b = 3$ in blue. Right: example of SCC curves with different abilities.

respondent's ability (θ_j) and the item's difficulty (b_i) need to be inferred. More advanced models, like the 2PL, estimate both discrimination and difficulty parameters, affording a better fit in some instances but also posing a risk of overfitting.

As all IRT models assume one single parameter for the respondent, their dual plots (known originally as person characteristic curves, here renamed as system characteristic curves (SCC)), also follow a logistic function (see figure 1(right)). Respondents who tend to correctly answer the most difficult items will be assigned to high values of ability. Difficult items in turn are those correctly answered only by the most proficient respondents. From this understanding and some common assumptions (ability and difficulty following some particular normal distributions), the latent variables can be inferred from a table of item-respondent pairs U_{ji}. Some two-step iterative variants of maximum-likelihood estimation (MLE), such as Birnbaum's method [45], can be used to infer all the IRT parameters.

IRT difficulty is characterised by being *system-independent* and *domain-generic* unlike the other metrics described above [46]. It also has some advantages over the use of mean performance as a metric of difficulty in terms of distribution, stability and predictability, as has been explored in the IRT literature [46].

2.3. Behaviour-based ML families

A classic approach to categorising ML techniques is to define families based on their formulation and learning strategy (e.g. neural networks, decision trees) [47–49]. However, this taxonomy does not take into account the intrinsic behaviour of a model (as measured by output agreement), especially in sparse domains where limited training data is available.

To comprehensively evaluate the robustness of ML models, it is necessary to analyse a diverse set of models under different parameters. In [50], the authors proposed a taxonomy of ML techniques for classification, grouping them based on the degree of behavioural agreement, i.e. differences in how they distribute the output class labels across the feature space. Both dense and sparse regions (where training data is scarce or absent) were considered using Cohen's kappa statistic [51] to effectively evaluate the difference between techniques. The use of Cohen's kappa is a valuable tool in the evaluation of the robustness of ML models. This statistic provides a means of comparing the behaviour of two models by assessing their predictions relative to each other, as opposed to the ground-truth. This approach allows for a deeper understanding of the models' behaviour, rather than simply their performance on specific tasks. Moreover, the use of Cohen's kappa is advantageous due to its robustness against data imbalance, making it a more reliable metric for comparing models. In dense regions, where training data is abundant, differences between models may be difficult to detect. However, their responses diverge in areas with sparse or no training data. In such low-density regions, the differences between ML models become more pronounced as each method extrapolates differently based on the limited examples available. Figure 2 demonstrates this phenomenon. The leftmost plot displays the original training data from a bivariate dataset used to train multiple ML models using diverse techniques. While these models show consistent behaviour in data-rich areas (for instance, around the point (0, 1)), their outputs are unpredictable in data-sparse regions (like point (1, 0.5)), which are also more vulnerable [52, 53]. Furthermore, sparse data can have a significant adverse impact on classifier performance, typically undermining predictive accuracy [50, 54].

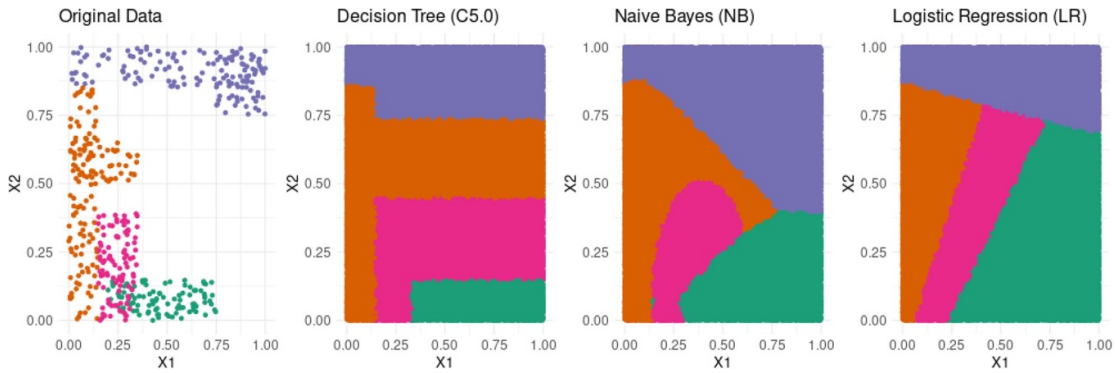


Figure 2. Illustration of the behavior of three different models trained over the same four-class classification dataset (shown on the leftmost plot). The pictures show the different predictions for particular class regions (known as decision boundaries) in dense and sparse areas.

A_1 : {MLP, AVNNet, LMT}
A_2 : {PMR, LR, PDA, SVM (Linear)}
A_3 : {FDA, MARS}
A_4 : {PLS}
A_5 : {RDA, HDRDA}
A_6 : {5-NN, 3-NN, Ibk, GPS}
A_7 : {LVQ}
A_8 : {SVM (Poly), SVM (RBF)}
A_9 : {GBM, CI_RF, BagFDA}
A_{10} : {RPART, Bagging}
A_{11} : {JRip}
A_{12} : {Cl_T}
A_{13} : {J48, C5.0}
A_{14} : {PART}
A_{15} : {RegRF, RF, ParRF}
A_{16} : {RFRules}
A_{17} : {NB, SDA}
A_{18} : {RBF}

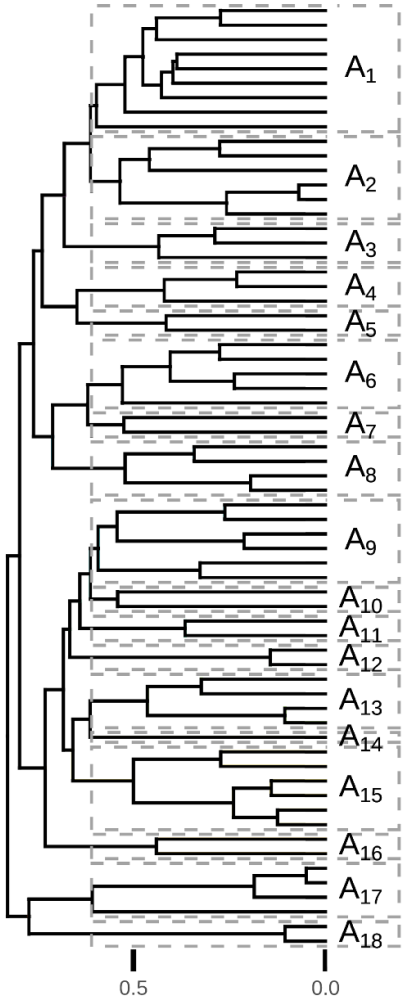


Figure 3. Dendrogram representing the hierarchical clustering (18 groups, A_i) of ML models.

The study compared 65 ML models, including variations of hyperparameters, using the pairwise comparison method and averaging the results over 75 datasets. The authors used hierarchical clustering to group the models into families based on their behaviour, resulting in 18 model families (see figure 3). The kappa statistic was used to objectively quantify the differences between two models or model families.

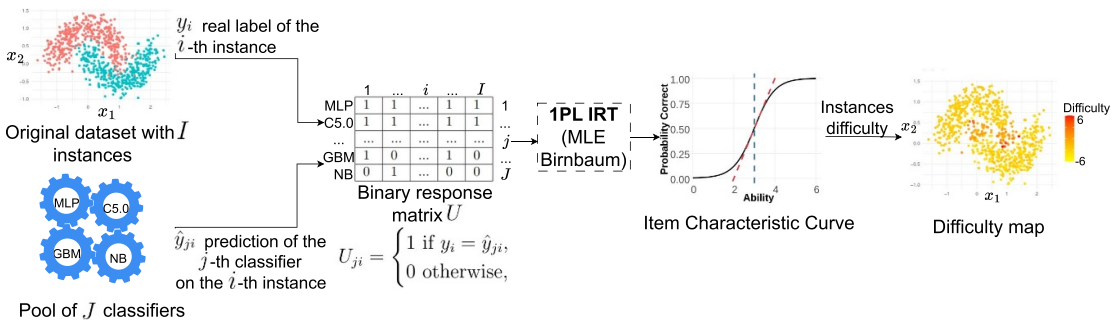


Figure 4. Graphical outline of the procedure for computing instance difficulty based on model responses using the IRT framework.

3. Robustness evaluation methodology

In this section, we describe a methodology for evaluating the robustness of ML to noise and instance difficulty. Using an extensive collection of model responses, we estimate instance difficulty using a robust evaluation framework informed by IRT and visualised by SCCs. We ensure representative noise introduction that reflects diverse real-world disturbances by perturbing datasets in a controlled manner. To identify patterns of resilience, we construct robustness taxonomies by clustering models based on their consistency of performance despite noise and varying levels of difficulty. This integrated approach aims to provide a nuanced understanding of model behaviour, facilitating informed decisions in the deployment of resilient AI systems.

3.1. Estimation of difficulty

The estimation of instance difficulty requires a preliminary check to ensure that each benchmark has a sufficient number of model scores (i.e. responses per instance, here referred to as 'items') [55]. In addition, these responses should come from a variety of model architectures and technologies to provide a robust assessment of difficulty. Figure 4 shows a schematic of the procedure applied to estimate instance difficulty, highlighting the process of aggregating model responses from different architectures to form the response matrix U. We collect responses for instances that the models have not encountered during training, typically using the test folds. This ensures that our performance scores truly reflect the models' ability to handle new data. The collection of model performances forms a $J \times I$ matrix , denoted U, containing all binary responses U_{ji}. From this we can obtain the ICCs, which graphically represent the probability of a correct model prediction for instances as a function of their estimated difficulty and the model's ability. These curves are useful for understanding the behaviour of different models in relation to the complexity of different instances. By visually analysing the ICCs, we can assess the discriminative power of items and the consistency of model performance as difficulty varies across a range of instances.

To effectively compute the instance difficulty, we adopt the IRT framework as recommended by [46], which use 1PL IRT models for simplicity where we set the discrimination parameter $a_i = 1.0$ for all items. Consequently, our focus shifts to inferring only two parameters: the ability θ_j of the models and the difficulty b_i of the instances.

In order to provide more transparency into our approach, in appendix A we present a detailed description of our procedure for estimating the difficulty of instances using synthetic data.

3.2. Introduction of noise

To evaluate model behaviour in different noise scenarios, including adversarial attacks, we need a versatile and universal noise generation method. Our approach will consider noise levels as a reflection of different contexts. Random noise will be introduced using standard probability distributions, as demonstrated in previous research [56]. This will help to standardise the process of noise introduction across different settings. We will handle the perturbation of instances by modifying attribute values within an appropriate range. For numerical attributes we use a Gaussian distribution to create values related to the original distribution. Using Gaussian distribution provides more flexibility in controlling the characteristics of the noise (through its parameters) compared to other artificial noise distributions such us the uniform noise. Additionally, Gaussian noise has been used on certain real world problems, such as sensor data processing [24, 57], image denoising [58], speech recognition [59] or differential privacy [60]. For nominal attributes, we employ a method that involves recalculating the probabilities for each category within an attribute to

simulate errors in data encoding or collection that might occur during real-world data handling. Thus, the noise injection is done as follows:

- **Numerical attributes**: Let ν be the level of noise to be injected into a numerical attribute at, and σ the standard deviation of all values of at. Then, a value x in at is modified as $x' \sim N(x, \sigma \cdot \nu)$, i.e. we follow a normal distribution using x as mean and σ multiplied by the noise level ν as standard deviation.
- **Nominal attributes**: Let $\{at_1, \ldots, at_m\}$ be the set of the m possible values of a nominal attribute at, and p the vector that represents the empirical distribution of at, that is, $p = (p_{at_1}, \ldots, p_{at_m})$, where, p_i is the frequency of value i. Consider we have an instance of value $x = at_j$ in at, we represent as the vector $t = (t_{at_1}, \ldots, t_{at_m})$ with $t_{at_i} = 0 \; \forall i \in \{1..m\}, i \neq j$, and $t_{at_j} = 1$. To insert a noise level ν, we calculate $\alpha = 1 - e^{(-\nu)}$, and then compute a new vector of probabilities $p' = \alpha \cdot p + (1 - \alpha) \cdot t$. The value of α, which ranges from 0 to 1, depends on the ν parameter. It balances the influence of the original distribution and the instance value on the updated probability vector p' that selects the noise value. Finally, we use p' in order to sample the new value x' of the attribute.

To measure the impact of noise on the models' performance, we will compare the predictions from the original test set with those from the noisy sets using the Kappa statistic. This comparison aims to quantify how noise influences model reliability and accuracy.

3.3. Measuring model robustness to noise and difficulty

To assess the robustness of ML models from different families, we apply all models to classify the same benchmark test set with progressively increasing the percentage of noisy instances. The degree to which a model is affected by noise provides an insight into its robustness. This can be assessed quantitatively using Cohen's kappa statistic [51]. The Cohen's kappa statistic is formulated as follows. Let $\hat{y}_1 = \{y_{11}, y_{12}, \ldots, y_{1N}\}$ and $\hat{y}_2 = \{y_{21}, y_{22}, \ldots, y_{2N}\}$ be the predictions of two models m_1 and m_2, on a test set of N instances. The Cohen's Kappa metric is defined as:

$$\kappa(\hat{y}_1, \hat{y}_2) = 1 - \frac{1 - p_0(\hat{y}_1, \hat{y}_2)}{1 - p_e(\hat{y}_1, \hat{y}_2)} \tag{2}$$

where $p_0(\hat{y}_1, \hat{y}_2)$ can be defined as the relation of the number of coincidences between the predictions of the two models and the size of the test set:

$$p_0(\hat{y}_1, \hat{y}_2) = \frac{1}{N} \sum_{i}^{N} (\delta(\hat{y}_{1i}, \hat{y}_{2i})) \tag{3}$$

where δ is the Kronecker's delta. If we define n_{ic} as the number of times that the model i predicted the class $c \in C$, we can formulate $p_e(\hat{y}_1, \hat{y}_2)$ as:

$$p_e(\hat{y}_1, \hat{y}_2) = \frac{1}{N^2} \sum_{c}^{C} n_{1c} \cdot n_{2c}. \tag{4}$$

We use the Cohen's Kappa statistic for measuring the agreement between a model's predictions on the original test set and those on its noisy counterpart. For this purpose, let us consider $\mathcal{T}$ as the Universe of all possible data sets that can be derived from different inputs. Given a test set $T \in \mathcal{T}$, we define a perturbation function $\phi : \mathcal{T} \to \mathcal{T}$ that applies noise to a data set, resulting in a perturbed test set $T' = \phi(T)$. If we have two models, m_1 and m_2, each trained on the same data, and they make predictions $\hat{y}_m = m(T)$ on the original test and $\hat{y}'_m = m(T')$ on the perturbed test, model m_1 is considered more robust than m_2 if the kappa consistency between its predictions on T and T' is greater than that of m_2, i.e. $\kappa(\hat{y}_{m_1}, \hat{y}'_{m_1}) > \kappa(\hat{y}_{m_2}, \hat{y}'_{m_2})$.

It is important to clarify that our goal is not simply to evaluate the overall performance of a model, but rather to understand how its behaviour is affected by different intensities of noise applied to instances of varying difficulty. For this reason, we disregard the actual class label in this context, recognising that introducing noise into an instance's attributes may alter its true class.

Figure 5 illustrates the complete methodology to estimate the SCCs of a given dataset d. First, we estimate the instances difficulty as explained in section 3.1. Then, we group the instances difficulties into bins of equal size $\mathcal{B} = \{B_1, B_2, \ldots, B_K : B_i > B_j, i > j\}$, where B_i represents the average difficulty of the instances within that bin. The original dataset d is used to learn multiple classification models $\mathcal{M} = \{m_1, m_2, \ldots, m_M\}$ under a k-fold cross validation setting. The function ϕ injects different grades of noise $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_N : \lambda_i > \lambda_j, i > j\}$ in the test set $T \subset d$. Take the function $\hat{y}(m, T, B, \lambda)$ as the set of predicted labels from a model $m \in \mathcal{M}$ applied to data test T after noise has been added at a rate of $\lambda \in \Lambda$

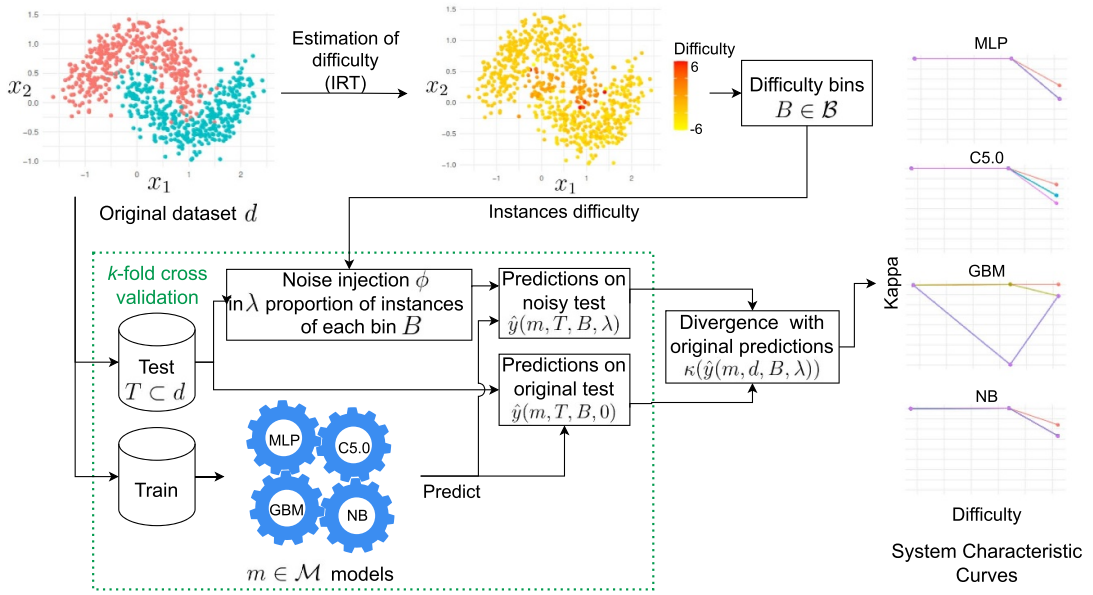


Figure 5. Description of the methodology to estimate the SCCs.

within the difficulty bin B. Meanwhile, $\kappa(\hat{y}(m,d,B,0),\hat{y}(m,d,B,\lambda))$ shows the kappa score contrasting the predictions given by model m on the original dataset d in difficulty bin B with the predictions made with the λ-noisy instances. For brevity, we denote this measure simply as $\kappa(\hat{y}(m,d,B,\lambda))$. To visualise this measure, we employ the SCCs. To construct an SCC, we plot the mean difficulty of each bin $B \in \mathcal{B}$ on the x-axis against the performance metric kappa on the y-axis (see figure 5(right)). For illustrative purpose, we chose to use three bins in this example. The number of bins represents a practical compromise between the granularity of detail and the simplicity required for effective interpretation.

3.4. Robustness-based taxonomies

As mentioned earlier, a SCC visualises how a model's responses change as it processes examples of varying difficulty, with the addition of noise interference. However, it falls short of revealing the overall patterns of a model's behaviour relative to others. To fill this gap, we aim to create robustness taxonomies for ML models. These taxonomies categorise models based on their persistence in maintaining consistent predictions despite the introduction of noise and the complexity of the instances they evaluate. The task of clustering models in this way is a complex one, compounded by the different characteristics inherent in each dataset. Notwithstanding these complexities, we have developed three methods to synthesise the collective behaviour of multiple models.

We start with our first approach, the differences across noise (DAN) method. This technique computes the average change in kappa across $\mathcal{D} = \{d_1, d_2, \ldots, d_L\}$ a set of datasets, as we step from one noise component λ_j to the next λ_{j-1}, all arranged in the matrix $\mathbf{A}_{M,K,N}$. The description of this calculation follows:

$$\mathbf{A}_{m,B_i,\lambda_j} = \frac{1}{L}\sum_{d=1}^{L}\left(\kappa\left(\hat{y}\left(m,d_d,B_i,\lambda_j\right)\right) - \kappa\left(\hat{y}\left(m,d_d,B_i,\lambda_{j-1}\right)\right)\right)_{1\leqslant i\leqslant K,\ 1<j\leqslant N}. \tag{5}$$

Our second method, differences across difficulty (DAD), averages the differences in kappa between consecutive bins, for each level of noise. In this case, the matrix $\mathbf{B}_{M,K,N}$ is computed as follows:

$$\mathbf{B}_{m,B_i,\lambda_j} = \frac{1}{L}\sum_{d=1}^{L}\left(\kappa\left(\hat{y}\left(m,d_d,B_i,\lambda_j\right)\right) - \kappa\left(\hat{y}\left(m,d_d,B_{i-1},\lambda_j\right)\right)\right)_{1<i\leqslant K,\ 1\leqslant j\leqslant N}. \tag{6}$$

Our last method, differences across noise and difficulty (DAND), is a hybrid approach that combines the results of the two previous methods:

$$\mathbf{C} = (\mathbf{A} \mid \mathbf{B}). \tag{7}$$

The matrices we have created serve as profiles for each model, capturing how its predictions change across different levels of noise when tackling instances of varying difficulty. These matrices allow us to

construct a pairwise dissimilarity matrix, which averages the discrepancies in model responses across the datasets, iteratively adjusted for different noise levels and instance complexity. Computing this matrix gives us the tools to examine how models vary in their robustness; in other words, we can quantify the relative distances between models in terms of their noise tolerance.

Once we have constructed the dissimilarity matrix, it allows us to cluster models based on the similarity of their responses under these conditions. By applying hierarchical clustering analysis, we can further refine these categories into a taxonomy that organises models into a hierarchy of nested groups. In particular, this clustering approach ensures that models with closely aligned performance across a spectrum of instance difficulty are grouped together. This method of clustering not only provides us with a systematic and empirical framework for understanding model similarities within families of ML algorithms, but also brings to light the models that stand out in terms of robustness.

3.5. Average κ loss analysis

The hierarchical clustering provides a taxonomy that groups the models with similar resilience against difficulty under different noise conditions. However, the taxonomy does not show intrinsic characteristics of the robustness of the groups, thus, we cannot assess to which extent the different groups are resilient against difficulty, noise injection, or both. To cover this flaw, we developed the average κ loss analysis, which accounts for the average loss in κ in function of the noise injection and instances difficulty.

Definition 1 (κ Gradients). Let $B_{\text{easy}} = \min(\mathcal{B})$ be the easiest difficulty bin, $B_{\text{hard}} = \max(\mathcal{B})$ be the hardest difficulty bin, $\lambda_{\max} = \max(\Lambda)$ be the maximum amount of noise introduced in the dataset d, and $\lambda_{\min}$ be the minimum amount of noise injected in the dataset d.

Given the model m and the dataset d, we define the easy κ gradient, $\Delta^{m,d}_{\text{easy}}$, as the difference in the easiest bin between the κ score at the maximum presence of noise and the κ score in the minimum presence of noise

$$\Delta^{m,d}_{\text{easy}} = \kappa\left(\hat{y}\left(m, d, B_{\text{easy}}, \lambda_{\min}\right)\right) - \kappa\left(\hat{y}\left(m, d, B_{\text{easy}}, \lambda_{\max}\right)\right). \tag{8}$$

Similarly, the hard κ gradient, $\Delta^{m,d}_{hard}$, is defined as

$$\Delta^{m,d}_{\text{hard}} = \kappa\left(\hat{y}\left(m, d, B_{\text{hard}}, \lambda_{\min}\right)\right) - \kappa\left(\hat{y}\left(m, d, B_{\text{hard}}, \lambda_{\max}\right)\right). \tag{9}$$

Finally we define the following loss functions:

Definition 2 (Average κ Losses). Given the model m, a set of datasets $\mathcal{D} = \{d_1, d_2, \ldots, d_L\}$, and the κ gradients $\Delta^{m,d}_{\text{easy}}$ and $\Delta^{m,d}_{\text{hard}}$, the average κ loss in difficulty, $\bar{L}^m_{\text{diff}}$, is defined as

$$\bar{L}^m_{\text{diff}} = \frac{1}{L}\sum_{d=1}^{L} |\Delta^{m,d}_{\text{easy}} - \Delta^{m,d}_{\text{hard}}| \tag{10}$$

and the average κ loss by noise, $\bar{L}^m_{\text{noise}}$, is defined as

$$\bar{L}^m_{\text{noise}} = \frac{1}{L}\sum_{d=1}^{L} \frac{\Delta^{m,d}_{\text{easy}} + \Delta^{m,d}_{\text{hard}}}{2}. \tag{11}$$

Note that $\bar{L}^m_{\text{diff}}$ takes into account the *kappa* loss between the hardest and easiest bins, while $\bar{L}^m_{\text{noise}}$ takes into account the *kappa* loss due to noise, as it calculates the mean of the losses in both extreme bins. We can visualize the result of these equations with a scatter plot, so we can visualize both equations, each one in function of the other. This allows us to have a combined view of the robustness against different instances difficulties under the presence of noise.

4. Experimental setting

We conducted our experiments by utilising the `R` language and the `caret` package [61]. All models were trained from scratch, without the use of any pre-trained models. The IRT 1PL models were estimated using the `MIRT` `R` package [62] following the experimental setting in [46]), and the predictions of a diverse range of models were obtained through the `OpenML` API [63]. The experiments involved up to 2000 evaluations per dataset. For the sake of clarity and balance in our visual representation, we choose to use five difficulty bins. Our experiments generate noisy test datasets with a predefined noise level $\nu = 0.2$. We will adjust the proportion of instances altered by noise, parameterized by λ, across various bins. These proportions will range from $\lambda = 0$ (where the original test set is unaltered) to $\lambda = 0.5$ (where the half of the test set is perturbed). The process to estimate the SCCs used a 5-fold cross-validation framework.

Table 1. List of benchmarks for the experiments, sorted by size, characterised by the number of instances, attributes, classes and complexity. Class distribution per benchmark is shown in figure B1.

Dataset	# Instances	# Attributes	# Classes	Complexity value	Complexity
Nursery	129 60	9	5	0	Simple
Wall-robot-navigation	5456	5	4	0	
Artificial-characters	102 18	8	10	0.01	
Page-blocks	5473	11	5	0.01	
GesturePhaseSegmentationProcessed	9873	33	5	0.02	
Letter	200 00	17	26	0.02	
Waveform-5000	5000	41	3	0.02	
Spambase	4601	58	2	0.03	
Satimage	6430	37	6	0.03	
Mfeat-morphological	2000	7	10	0.03	
Analcatdata_dmft	797	5	6	0.04	
First-order-theorem-proving	6118	52	6	0.05	
Gas-drift	139 10	129	6	0.06	
Segment	2310	20	7	0.06	
Yeast	1484	9	10	0.06	
Texture	5500	41	11	0.08	Complex
Optdigits	5620	65	10	0.12	
Vowel	990	13	11	0.14	
Mfeat-zernike	2000	48	10	0.24	
Mfeat-karhunen	2000	65	10	0.32	
Mfeat-fourier	2000	77	10	0.38	
Gina_prior2	3468	785	10	2.26	
CNAE-9	1080	857	9	7.14	

4.1. Data and classifiers

Estimating IRT difficulty requires datasets with results from multiple models for each instance. However, obtaining instance-wise results, meaning a $J \times I$ matrix displaying the performance of each system $j \in \{1..J\}$ for each instance $i \in \{1..I\}$, can be challenging as it is not easy to find experiments that are not reported in an aggregated format. `OpenML` [64] is a valuable resource for this, as it allows sharing of data sets and results in detail, including curated datasets like `OpenML-CC18`, from which we selected a set of 23 benchmarks for supervised learning (detailed in table 1 and figure B1 in appendix B) that met the criteria of a sufficient number of items I and models J. The aim was also to achieve a diverse set of benchmarks, including datasets with different number of instances (up to 200 00 in our selection), attributes (up to 857), as well as cover diverse domains, including handwriting recognition, chemical sensor measurements, spam detection, and yeast gene identification.

For implementing the calculus of instance difficulty using IRT, we use the `MIRT` [62] R package, which allows us to use the Birnbaum method for estimation. An advantage of IRT-based packages such as `MIRT` is the inclusion of goodness-of-fit indicators. These indicators assess whether the observed data fit well with the expected results under the statistical IRT model. If the statistics of an item do not fit well, this may indicate that the IRT model does not adequately represent the data-generating process, potentially leading to biased parameter estimates. However, in our case, no estimated models were discarded due to poor item fit statistics or inconsistencies, indicating confidence in the validity of our inferences.

Our experiments have shown that certain characteristics of datasets, including the total number of instances (#ins), the number of attributes (#att), and the diversity of classes (#clas), play a critical role in influencing the performance of the model. In order to quantify the complexity of a dataset, we have developed a metric that encapsulates these aspects. This complexity metric is formulated as the product of the number of attributes and the number of classes divided by the number of instances, as shown in equation (12).

$$\text{Complexity} = \frac{\#\text{att} \cdot \#\text{clas}}{\#\text{ins}}. \tag{12}$$

According to this measure, a dataset is considered more complex if it contains a higher number of classes and attributes, especially when combined with a lower number of instances [65]. We use this metric to classify datasets into categories such as ‘simple’ and ‘complex’. Specifically, we classify a dataset as ‘complex’ if its complexity metric exceeds 0.08 (see table 1). This threshold was chosen based on our observation that complexity escalates significantly beyond this point.

Table 2. List of the 18 models employed for the experiments, along with the hyperparameters used (* indicates that the default hyperparameters have been used).

Technique (ID)	Parameters
C5.0 (C5.0)	`trials = 1, winnow = False`*
Conditional inference tree (CI_T)	`mincriterion = 0.05`
Flexible discriminant analysis (FDA)	`degree = 1, nprune = 17`
Stochastic gradient boosting machine (GBM)	`interaction.depth = 2, n.trees = 50`
Propositional rule learner (JRip)	`NumOpt = 2, NumFolds = 3, MinWeights = 2`*
K-nearest neighbor (3NN)	`K = 3`
Learning vector quantization (LVQ)	`size = 50, K = 3`
MultiLayer perceptron (MLP)	`1 hidden layer, 7 neurons`
Multinomial logistic regression (MLR)	`summ = 0, censored = FALSE`*
Naive bayes (NB)	`laplace = 0, usekernel = FALSE`*
Rule-based classifier (PART)	`threshold = 0.25, pruned = 'yes'`*
Radial basis function network (RBF)	`size = 5`*
Regularised discriminant analysis (RDA)	`gamma = NA, lambda = NA`
Random forest (RF)	`mtry = 64`
Classification and regression trees (RPART)	`cp = 0.01`*
Partial least squares (PLS)	`ncomp = 3`
Support vector machines (SVM)	`Poly, degree = 2`
Random forest rule-based model (RFRules)	`mtry = 64`

In terms of ML models, we used a set of 18 ML models from different ML families (from [50]), as detailed in table 2. These families were derived from a large pool of 65 models that were evaluated on a variety of datasets and grouped into families based on their performance, as described in section 2.3. To represent each family, we selected a single model (the centre or centroid of each cluster) to be representative of its family. This approach allowed us to have a diverse set of models, providing a comprehensive understanding of the robustness of different ML families. Unless otherwise specified, we use the default hyperparameters for the models trained. Different hyperparameters were selected after initial exploratory experiments (e.g. to correct execution and performance problems for different dataset and model combinations).

4.2. Experimental questions

The purpose of posing different research questions is to gain insight into the relationship between the robustness of ML models and the difficulty of instances, which have been altered by varying levels of noise. For this, we set 4 experimental questions.

- **Q1**: What is the distribution of the IRT-difficulty metric across different benchmarks? This question seeks to understand how the estimated difficulty of instances varies across different benchmarks, using the IRT-difficulty metric.
- **Q2**: Are there noticeable differences in the robustness of different models, considering the difficulty of instances? This question aims to determine if there is a relationship between the difficulty of instances and the robustness of different ML models.
- **Q3**: Can we group ML models based on their robustness, taking into account the difficulty of instances? This question seeks to identify if it is possible to categorise models into groups based on their robustness, considering the influence of instance difficulty. The results of this question could provide valuable insights into the design and selection of ML models for specific tasks and domains.
- **Q4**: How does dataset complexity influence the robustness of ML to noise and instance difficulty? This question seeks to analyse the interplay between dataset complexity (from simpler datasets with few features or classes to more complex ones with a high number of features or classes), noise levels, and instance difficulty.

5. Results

5.1. Difficulty distribution per benchmark

The IRT difficulty parameters are usually characterised by a normal distribution with a standard deviation of 1, and different locations for each dataset. The acceptable range for the difficulty parameter is determined by the goal of the test and the group of individuals being tested. For example, in educational assessments, values around 1 are common, while in health evaluations, values around 4 are more typical. In the case of ML benchmarks, values around -6 to 6 are generally accepted , as demonstrated in [6, 46]. To avoid extreme values, any instances with difficulties outside the range of $[-6, 6]$ were discarded in our experiments, which

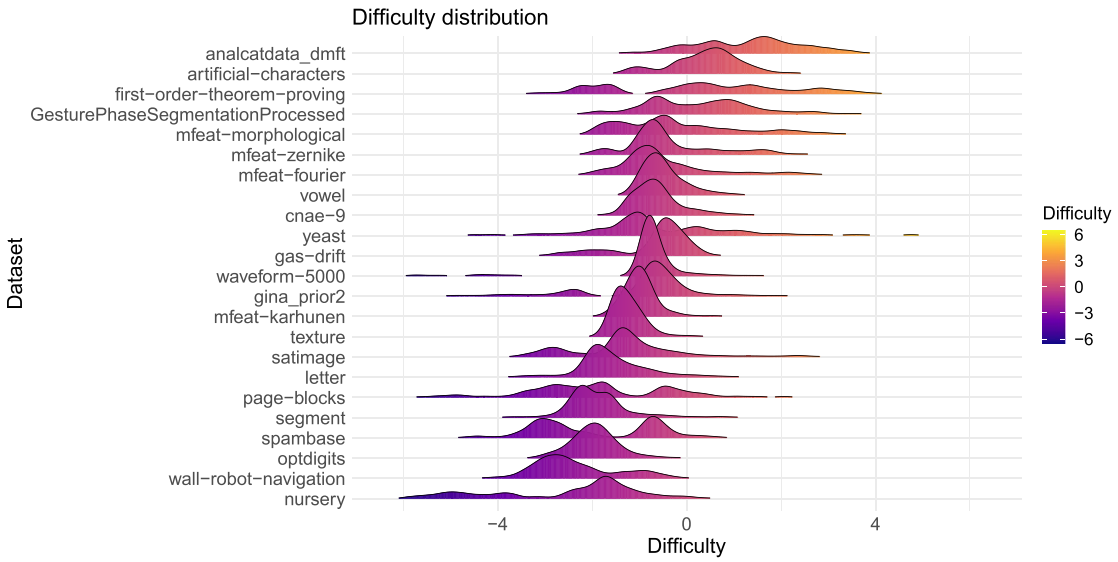


Figure 6. IRT-difficulty distribution per dataset. Benchmarks sorted by average difficulty.

affected less than 0.1% of instances in all benchmarks. The IRT difficulty distribution for each benchmark is presented in figure 6 and shows a standard deviation of around 1 for most cases.

Regarding the location of the IRT difficulty parameters (**Q1**), it is observed that the analcatdata_dmft benchmark contains more challenging instances, with a mean difficulty of 1.33 ± 1.12. On the other hand, the nursery dataset contains less difficult instances, with a mean difficulty of -2.50 ± 1.56. While the difficulty distributions across different benchmarks often exhibit a bell-shaped pattern, deviations from a perfect normal distribution can be observed, indicating varying levels of skewness and kurtosis. Some benchmarks exhibit a skewed distribution (e.g. artificial-characters, mfeat-morphological), while others have a multimodal distribution (e.g. analcatdata_dmft, spambase, satimage), indicating that the instances are grouped into different levels of difficulty. This variation in the distribution could be attributed to the diverse range of systems used for difficulty estimation, as reported in [46].

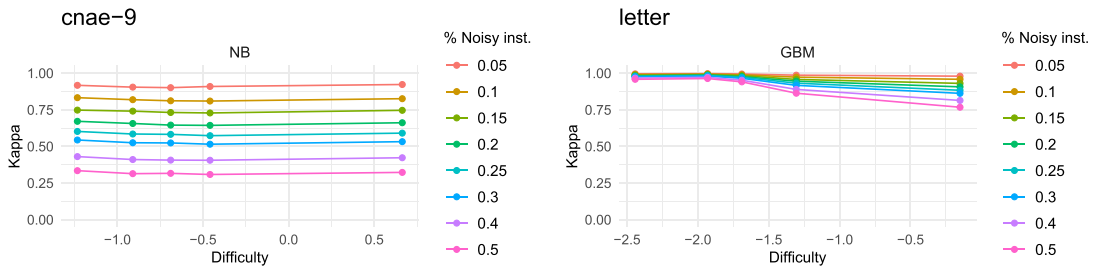
In general, it is important to consider the location and distribution of IRT difficulty parameters when designing and interpreting the results of ML benchmarks. The diversity in the distribution of IRT difficulty parameters can affect the generation and results of the tests and the evaluation of the models (see, e.g. those approaches related to adversarial data collection [66, 67], which focus on modifying the most difficult instances in ML benchmarks). On the other hand, the standard deviation of 1 in most cases suggests that the parameters are relatively well calibrated and consistent across benchmarks. However, the presence of outliers and variations in the distributions highlights the importance of considering the characteristics of each benchmark when interpreting the results.

5.2. Model robustness based on noise and difficulty

Here we aim to determine the relationship between the difficulty of instances and the robustness of different ML models in terms of their behaviour (**Q2**). To do this, we evaluate the performance of each technique listed in table 2 by comparing its predictions on the original test sets for each dataset in table 1 with its predictions on noisy test sets using the kappa metric. A model that is highly robust against both noise and difficulty would exhibit a kappa of 1, meaning its predictions remain unchanged regardless of the level of noise or difficulty. Conversely, as more noise is introduced to increasingly difficult instances, the behaviour of the model may change, resulting in a decrease in the kappa values shown in the SCCs.

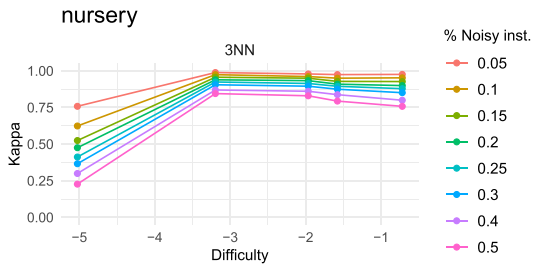
Figure 7 shows four representative examples of SCCs (kappa values plotted against difficulty bins), with the average difficulty per bin shown on the x-axis, to illustrate the general behaviour observed when analysing the performance of different models under the influence of noise for instances of varying difficulty. However, there are other models that show different (and less common) behaviour. A more comprehensive analysis of the SCCs obtained for all the models and datasets in tables 1 and 2 is provided in the appendix C.

In general, for all SCC, kappa equals 1 when the test set is not perturbed ($\lambda = 0$), as the output labels of the different trained models are compared with themselves. As we increase the number of perturbed instances (maintaining the same proportion for each bin of difficulty), differences in the behaviour of the analysed techniques become apparent. Moving to the different performances, the model behaviour depicted

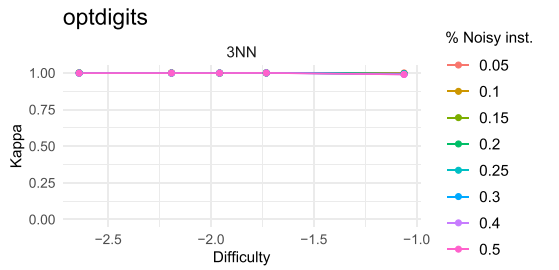


(a) Robust model regardless of difficulty but not against noise.

(b) Robust model against noise for easy instances, but not for difficult instances



(c) Robust model regardless of difficulty for medium and high difficulty bins, but not for easy instances.



(d) Robust model regardless of difficulty and noise.

Figure 7. Examples of model behaviour (SCCs), given by the kappa vs difficulty for different models and benchmarks, varying the proportion of noisy instances (λ) for each difficulty bin.

in figure 7(a) shows lines that are almost horizontally parallel (i.e. barely affected by difficulty) since performance remains nearly constant for all difficulty bins. However, the curves move closer to the x-axis (decreasing kappa values) as more noise is introduced. This suggests that the models following this pattern are robust to difficulty but weak to noise.

In contrast, models that exhibit behaviour similar to that shown in figure 7(b) are more susceptible to the difficulty of instances. This is the most common behaviour found in our analysis: as more noise is added to the most difficult instances, the performance of the model becomes increasingly degraded (decreasing kappa values), while the effect of noise on easy instances is less pronounced. This implies that this type of model is robust to noise in easy instances, but weaker in the face of noise in more difficult bins. Our research shows that the introduction of noise generally leads to misclassification of instances in the hardest bins in favour of a single class. The more noise that is introduced, the more instances are misclassified into that class, resulting in a greater decrease in the kappa metric. This phenomenon may also indicate that the more difficult instances are close to the decision boundary or in regions where classes overlap, making the behaviour of most techniques in these regions more unpredictable than in less difficult regions.

Figure 7(c) shows the opposite behaviour. The predictions become less susceptible to noise in the more difficult bins, while the easier bins see a significant decrease in Kappa. Upon closer examination, this is attributed to the class distributions in the simpler bins (see appendix B, figure B2). These bins often consist of mostly instances of a single class, but they can be misclassified as noise increases, leading to a decrease in kappa. The model's tendency to focus on a single class in these bins results in over-specialisation and sensitivity to even minor changes in predictions.

It is worth mentioning that due to the variety of models and datasets used, there may be other, less common behaviours (or variations of the behaviours described above) that can be observed. For example, there are models that are minimally or not at all affected by noise and/or difficulty, as shown in figure 7(d). This underlines the significance of also conducting individualised analysis of the robustness of models, taking into account the further properties of the models and datasets in question.

In general, we have seen that our analysis is affected by several factors, not only the amount of injected noise and the difficulty of the instances, but also the intrinsic properties of the dataset, such as the class distribution and the number of instances and attributes. To further refine our analysis and gain a deeper understanding, in the following section we will introduce a third dimension to our analysis: the complexity of the benchmark in terms of the number of instances, attributes and classes. As a result, we will examine

Table 3. The Silhouette Width (*SilD*) and Average between Cluster (*AvgBC*) distance are calculated for each method (DAN, DAD and DAND) with a varying number of clusters, ranging from 2 to 10. Bold values indicate the highest value for each metric among the three methods for each cluster count.

		Number of clusters								
Metrics	Method (Differences across)	2	3	4	5	6	7	8	9	10
AvgBC	Noise (DAN)	0.09	0.08	0.07	0.07	0.07	0.06	0.06	0.06	0.06
	Difficulty (DAD)	0.15	0.14	0.13	0.13	0.13	0.13	0.12	0.12	0.12
	Noise and Difficulty (DAND)	**0.16**	**0.15**	**0.15**	**0.15**	**0.15**	**0.14**	**0.14**	**0.14**	**0.14**
SilW	Noise (DAN)	**0.47**	**0.44**	**0.40**	0.29	0.23	0.22	0.22	0.21	0.20
	Difficulty (DAD)	0.42	0.34	0.27	0.26	0.28	0.30	0.26	0.22	0.21
	Noise and Difficulty (DAND)	0.35	0.27	0.30	**0.32**	**0.31**	**0.30**	**0.30**	**0.32**	**0.27**

three scenarios: (1) all 23 datasets, (2) simple datasets, and (3) complex datasets. The complexity of the datasets was established following the equation (12), and it is shown in table 1.

5.3. Robustness-based behaviour taxonomy

Building on **Q3**, we seek first to confirm the validity of the three approaches we proposed for aggregating System Characteristic Curves (SCCs), namely DAN, DAD, and DAND, as detailed in section 3.4. To evaluate the effectiveness of these aggregation methods, we use two well-established cluster quality metrics: the Average Between Clusters (*AvgBC*) distance [68], which evaluates the separation between clusters, and the Silhouette Width (*SilW*) [69], considers both the inter-cluster distance and the intra-cluster tightness. These metrics help to determine the distinctiveness and coherence of the clusters formed by our methods.

As shown in table 3, the DAND method tends to delineate more distinct clusters, a sign of superior clustering quality. In particular, at around five clusters, the DAND method—taking into account variances in both noise and difficulty—outperforms the other methods in maintaining separability and coherence of clusters. We observe that fewer clusters result in more separation but less compactness, as indicated by higher *AvgBC* but lower *SilW* values. Using the elbow method [70], we also inferred that the optimal number of clusters is around five. Figure 8(left) presents the hierarchical clustering taxonomy that considers all datasets.

Due to space constraints, while all SCCs for both models and datasets are included in the appendix C, we have chosen to focus on presenting the behaviour of a prototype model for each cluster. These prototypes, which are considered to be the most representative of their respective groups, encapsulate the range of model behaviours under evaluation. The SCCs of these prototypes are shown in figures 9–11, corresponding to scenarios with all datasets, simpler datasets and more complex datasets respectively. This selective presentation makes it easier to understand the cluster-specific model behaviour.

The joint analysis of the taxonomy and the SCCs of each prototype conveys the following findings:

- Group A_1 with the PART model as its prototype, contains models that show a notable sensitivity to noise (see figure 9(a) for its SCCs). The SCCs of these models tend to be parallel and horizontal, suggesting that the difficulty of the instances has a muted effect on their performance relative to noise. Models in this group, which often use decision rules, decision trees or nearest neighbours, reflect findings in the literature that emphasise their vulnerability to noisy data. This finding is critical when applying these models in real-world scenarios where data may be contaminated with various types of noise.
- Group A_2 is characterised by models that are highly sensitive to noise, as evidenced by the increasing separation of their SCCs with increasing noise levels (figure 9(b)). The Naive Bayes (NB) model stands out as the prototype of this group, suggesting that its simplifying assumptions may contribute to a less damped response to noise. NB has been found to exhibit a distinct and unique behaviour during our experimental phase. This peculiar behaviour suggests that NB models may require additional noise handling techniques, or should be carefully considered when selecting them for applications where data quality cannot be guaranteed.
- Group A_3 is led by the Learning Vector Quantization LVQ model, which is more robust to noise but more sensitive in the face of complex instances (figure 9(c)). The similarity of its SCCs across different noise intensities suggests an inherent robustness to noise, but as the difficulty of the instances increases, performance varies. This robustness to noise can be exploited in scenarios where data corruption is a concern, but additional strategies may be required to deal with more difficult instances.
- Group A_4 represents models that are more easily perturbed by noise than by the intrinsic complexity of the instances (figure 9(d)). The prototypes in this group, such as the Conditional Inference Trees (CI_T), show a marked separation in their SCCs following the introduction of noise, suggesting that noise has a more

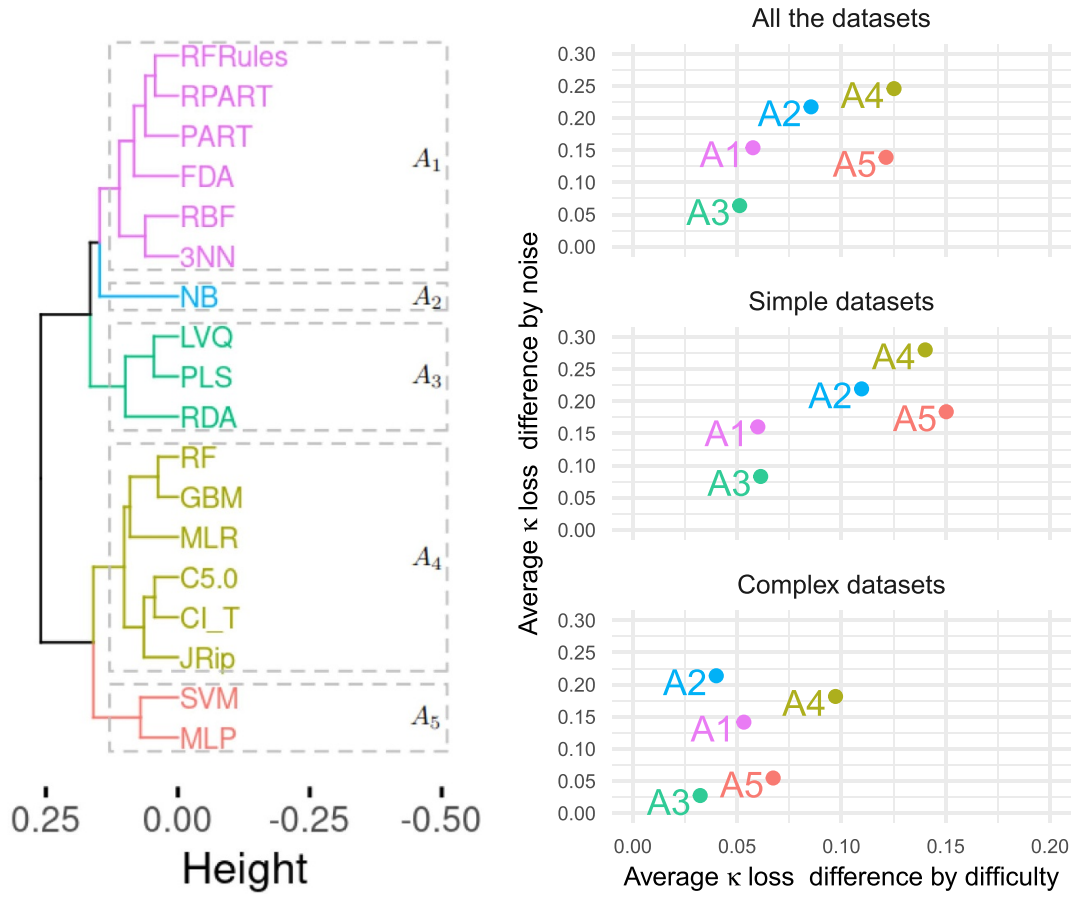


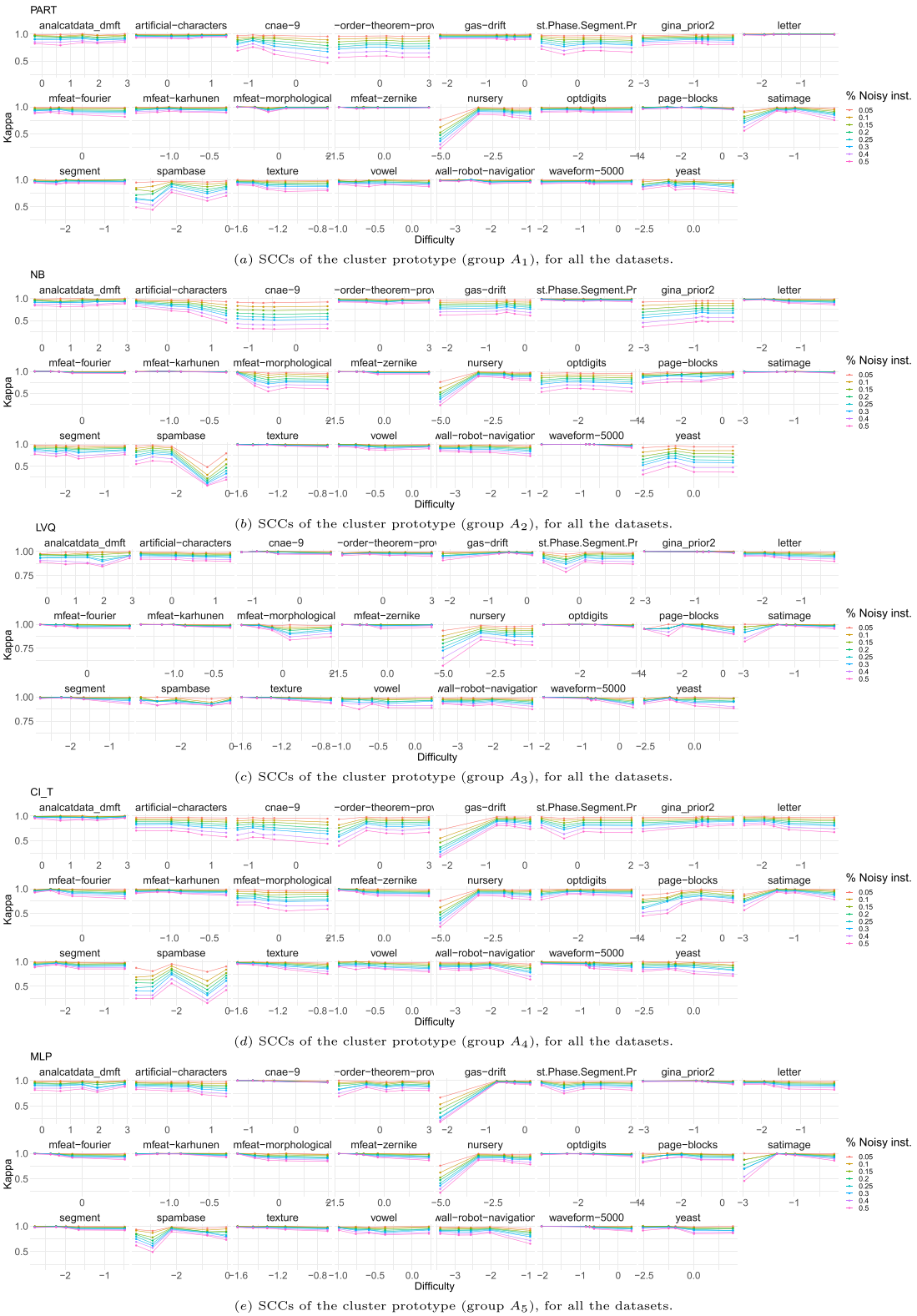
Figure 8. Robustness taxonomy and the representation of robustness against difficulty and noise of each cluster. (Left) Robustness taxonomy by aggregating across levels of noise and difficulty bins. (Right) Average loss κ difference considering noise and difficulty. We show the representation considering all the datasets, the simple datasets, and the complex datasets.

pronounced effect on these models than complexity. This finding is important when selecting models for situations where noise is present and reliability is critical, such as in medical or financial applications.

- Group A_5, is exemplified by the multi-layer perceptron MLP model. The susceptibility of this group seems to be more influenced by the difficulty of the instances, especially the most difficult ones, while being less influenced by noise (figure 9(e)). This category of models, which includes neural networks, has been extensively studied for its robustness, often in the context of image processing and adversarial attacks. Our results suggest that their resilience extends to tabular data affected by random noise, positioning them as a viable option when robustness to noise is a priority.

5.4. Dataset complexity and model robustness

An intriguing outcome from our experiments is the relative robustness exhibited by models in the face of complex datasets (**Q4**). This observation suggests that the complexity of a dataset may buffer the impact of noise and difficulty, potentially due to factors such as a higher number of classes. This effect is discussed further in figure 8(right), which showcases the average κ loss across difficulty levels and noise intensities. This figure provides a visual summary of the robustness or vulnerability of each model cluster under varying conditions. The first plot encapsulates all datasets, providing an overarching view of each cluster's behaviour when exposed to noise and difficulty. Subsequent plots segment the analysis into datasets categorised by complexity: the middle plot covers datasets with fewer distinguishing features (simple), and the bottom plot focuses on those with a greater number of features or classes (complex). Within these plots: (a) a cluster's proximity to the zero mark on the x-axis indicates robustness to instance difficulty; these models maintain their predictive consistency regardless of how challenging the data is; and (b) the y-axis measures the sensitivity of each cluster to the introduction of noise; a higher value corresponds to a greater divergence in model performance when noise is introduced.



(a) SCCs of the cluster prototype (group A_1), for all the datasets.

(b) SCCs of the cluster prototype (group A_2), for all the datasets.

(c) SCCs of the cluster prototype (group A_3), for all the datasets.

(d) SCCs of the cluster prototype (group A_4), for all the datasets.

(e) SCCs of the cluster prototype (group A_5), for all the datasets.

Figure 9. SCCs of each cluster prototype, including all the datasets.

It is noteworthy that almost all clusters show more robust behaviour with complex data sets, as reflected by a lower average kappa loss in the face of both noise and difficulty. This pattern suggests an adaptive resilience that complex datasets seem to induce in the models, possibly because their rich feature sets provide a buffer against noise and instance difficulty. In contrast, for simpler datasets, the observed effects of noise and instance difficulty are more pronounced. One reason for this phenomenon could be the tendency for bins in simpler datasets to specialise in fewer classes (see figure B2 in the appendix B). The relative rarity of

(a) SCCs of the cluster prototype (group A_1), for simple datasets.

(b) SCCs of the cluster prototype (group A_2), for simple datasets.

(c) SCCs of the cluster prototype (group A_3), for simple datasets.

(d) SCCs of the cluster prototype (group A_4), for simple datasets.

(e) SCCs of the cluster prototype (group A_5), for simple datasets.

Figure 10. SCCs of each cluster prototype, including only the simple datasets.

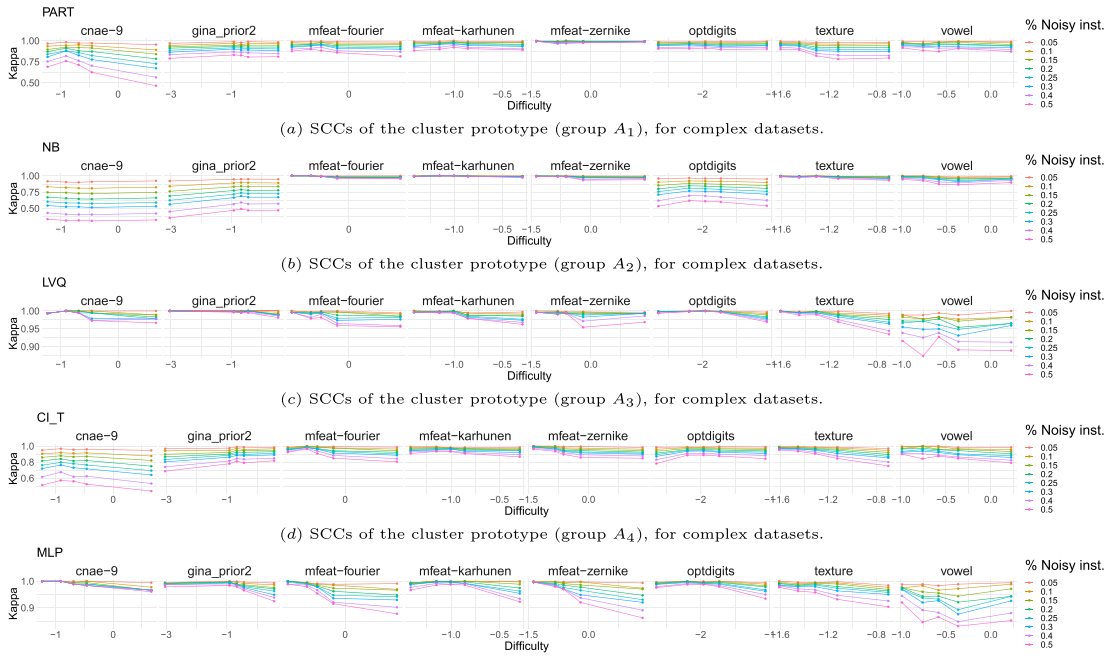
classes in such datasets increases the likelihood that certain bins will become dominated by one class, leading to a pronounced variance in kappa. Example datasets such as nursery or first-order theorem-proving, where the simplest instances tend to cluster in a single class, illustrate this specialisation. This specialisation effect can be seen as a marked drop in kappa within the easiest bins of their respective SCCs (reflected in figure 10).

In this regard, we find that cluster A_1 remains consistent and shows negligible variation in its behaviour whether applied to simple or complex datasets, while cluster A_2 is resilient to noise but is more challenged by the difficulty of instances in simpler datasets. The cluster A_3 models are the least disturbed by noise and are adept at handling difficult instances, especially evident in simpler datasets where they rival the consistency of A_1 but surpass it in noise immunity. In contrast, cluster A_4 contains the most vulnerable models, heavily influenced by both noise and difficulty in simpler datasets, although their robustness increases amidst the intricacies of complex datasets. Cluster A_5 follows an interesting arc; its models struggle with difficulty in simple landscapes, but adapt to maintain a stronger defence against both difficulty and noise when navigating the multifaceted terrain of complex datasets.

In essence, figure 8(right) not only highlights the importance of considering dataset complexity when assessing model robustness, but also reveals intricate patterns in how different models respond to the challenges posed by noise and instance difficulty.

6. Implications of model robustness taxonomy

Our robustness-based taxonomy provides invaluable insights into the resilience of ML models under the stress of noise and fluctuating instance difficulty. Given the diverse nature of real-world data, these insights provide a nuanced guide for practitioners in the selection and deployment of models across domains.



(a) SCCs of the cluster prototype (group A_1), for complex datasets.

(b) SCCs of the cluster prototype (group A_2), for complex datasets.

(c) SCCs of the cluster prototype (group A_3), for complex datasets.

(d) SCCs of the cluster prototype (group A_4), for complex datasets.

(e) SCCs of the cluster prototype (group A_5), for complex datasets.

Figure 11. SCCs of each cluster prototype, including only complex datasets.

Our study reveals a cohort of models within cluster A_1 that exhibit a robust nature, maintaining consistent performance regardless of dataset complexity. This predictability is essential in high-reliability situations where the cost of failure is high, such as healthcare diagnostics or autonomous vehicle navigation. The assurance of consistent performance facilitates the extensive validation process that might otherwise be required across different data types.

Cluster A_2 explores models that are good at dealing with noise, but reveal weaknesses in addressing the fundamental challenges posed by the data. This finding underscores the importance of fully understanding the characteristics of the dataset—not just the presence of noise—when selecting models for specific tasks. For developers and data scientists, this distinction requires heightened attention to the complexities and nuances of dataset components, to ensure that models are tailored to deftly handle the inherent difficulties of the data.

The exemplary robustness of models in cluster A_3 to both noise and instance difficulty signals a compelling choice for applications with simple but noisy data. This resilience makes them suitable candidates for use in industrial settings where the data may not vary significantly, but is potentially corrupted by noise, allowing for accurate and stable model performance.

A fascinating nuance uncovered by cluster A_4 suggests that more elaborate datasets, characterised by higher dimensionality and richer features, can sometimes enhance a model's robustness by reducing its susceptibility to noise and difficulty. This finding can inform the development and deployment strategies of ML models for complex systems, suggesting that complicated data can indeed be an ally in improving performance reliability.

The models aggregated in cluster A_5 demonstrate an adaptive robustness that is able to significantly improve on complex datasets, while potentially struggling with simpler datasets. This ability to dynamically adjust robustness in response to the complexity of the data landscape highlights the potential of ML models to exploit complex data features, offsetting the perturbations caused by noise and difficult instances.

The taxonomy also brings to the fore the impact of class distribution on robustness metrics, particularly in simpler datasets where bins may specialise in single classes. This observation highlights the need for careful consideration of class distribution in model evaluation, training, and validation processes to achieve robust performance. Taken together, the extended discussions derived from our refined taxonomy provide data-driven advances in the fundamental understanding of ML model behaviour under different operational conditions. For practitioners, these discussions serve as a strategic compass for navigating ML model selection and deployment, ensuring that robustness is prudently matched to the unique requirements of each application. This attention to robustness, when effectively incorporated, paves the way for the creation of reliable and trustworthy AI systems that are tailored to thrive amidst the complexities of real-world data.

7. Conclusions and future work

Our evaluation framework and taxonomy provides a comprehensive approach for examining the robustness of ML models in the presence of noisy instances, offering a systematic and thorough evaluation environment for practitioners to assess the resilience of models under various conditions. By taking into account the difficulty of instances, our research provides critical information so that practitioners can better understand the strengths and weaknesses of models and make informed decisions about selecting robust, fit-for-purpose models that can withstand the noise of real-world scenarios.

We have shown that the SCCs can help to identify the most appropriate models based on their robustness to different levels of difficulty. In cases where the difficulty values of instances in a test or validation set are unknown, there are several straightforward methods for estimating them, such as averaging the difficulty values of the most similar examples in the original training set, or training a difficulty estimator (as in [46]). This can be done on small sets or even on individual instances, allowing the best model to be determined for each instance. By using difficulty predictions, practitioners can limit the use of models to only those instances that are considered easy and for which the models are robust. This approach provides a starting point for further research into the robustness of ML models, and sheds light on the limitations and strengths of different models and families.

The SCCs also show that it is common for the kappa value in simple bins, where the majority of instances belong to the same class, to drop significantly when appropriate noise is introduced. This information can be used to develop countermeasures against adversarial attacks. If a new instance to be classified falls into this difficulty bin and is classified by the model as being in a different class from the expected class for that bin, it may have been deliberately altered to become adversarial.

We plan to extend our evaluation framework to include benchmarks from additional domains, such as vision and text, and a broader range of perturbation functions. This will provide deeper insights into the robustness of different ML models and improve the diversity and generalisability of our approach. Specifically, we will investigate scenarios such as object detection in autonomous vehicles, where sensor inaccuracies, environmental factors and equipment wear introduce complex noise. We may also use noise injection techniques, such as modifying sensory data or adding artificial noise to imagery and radar data, to mimic specific real-world conditions. Additionally, we will examine NLP models by challenging them with text perturbations (e.g. character or word-level perturbations [71]), to test their ability to maintain context understanding under distorted input conditions. We intend to use existing datasets that naturally contain noise variations, and may also use noise injection techniques to simulate real-world conditions by modifying sensory data or adding artificial noise to imagery, radar and textual data. In addition, this extension may require the creation of difficulty estimators [46] and specialised perturbation functions [72] to generate noisy inputs so that we can assess the robustness of systems to different levels of challenge.

In addition, we plan to explore alternative settings of our framework to gain further insight into the behaviour of ML models. For example, we may investigate the robustness of models when only the most relevant attributes are perturbed, or by using alternative noise injection methods. All in all, our ongoing efforts promise to uncover a wealth of information about the robustness of ML models and provide cutting-edge insights for practitioners in the field.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/rfabra/ml-robustness-difficulty [73].

Acknowledgments

We thank the anonymous reviewers for their comments. This work was funded by the Norwegian Research Council Grant 329745 Machine Teaching for Explainable AI, the MIT-Spain—INDITEX Sustainability Seed Fund under Project COST-OMIZE, CIPROM/2022/6 (FASSLOW) funded by Generalitat Valenciana, the EC H2020-EU Grant Agreement No. 952215 (TAILOR), and Spanish Grant PID2021-122830OB-C42 (SFERA) funded by MCIN/AEI/10.13039/501100011033 and 'ERDF A way of making Europe'. RFB is supported by predoctoral Grant PRE2019-090892.

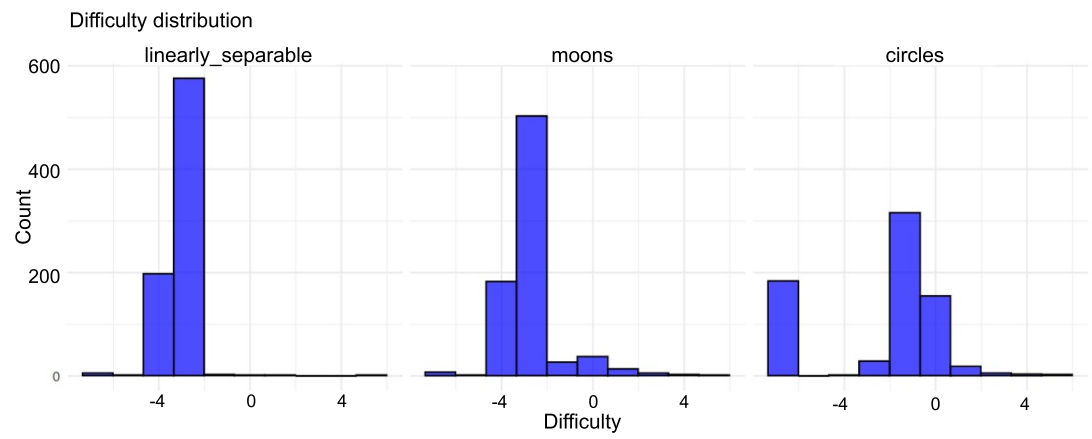


Figure A1. Difficulty distribution of the benchmarks.

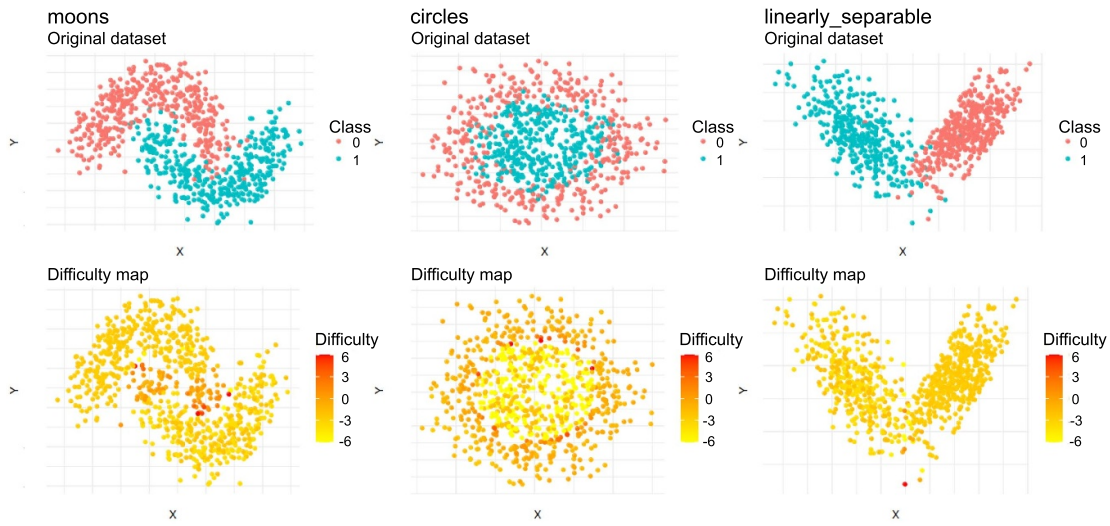


Figure A2. Difficulty distribution at instance level for each benchmark.

Appendix A. Experiments on synthetic data

We illustrate the Robustness Evaluation Methodology described in section 3 through the use of synthetic datasets to demonstrate the capabilities of our framework. More concretely, we include the estimation of difficulty and the construction of the SCCs for different grades of noise injected in the data. We use three common synthetic datasets characterised as *moons*, *circles* and *linearly_separable*, showing that even in data with limited pattern diversity, difficulty distributions often assume a normal distribution, as shown in figure A1. Figure A2 further visualises this by mapping individual instances, with difficulty indicated by colour gradients, showing that the most difficult instances tend to cluster near their respective class boundaries.

Going beyond difficulty estimation, we employ 18 models spanning different ML families, which are described in section 2.3. We deliberately introduce varying degrees of noise into the test set and assess the impact on model performance using the SCCs. Figure A3 shows the SCCs for the synthetic datasets, which illustrate the different responses of models to both instance difficulty and injected noise, highlighting the complex dynamics at play that affect model robustness.

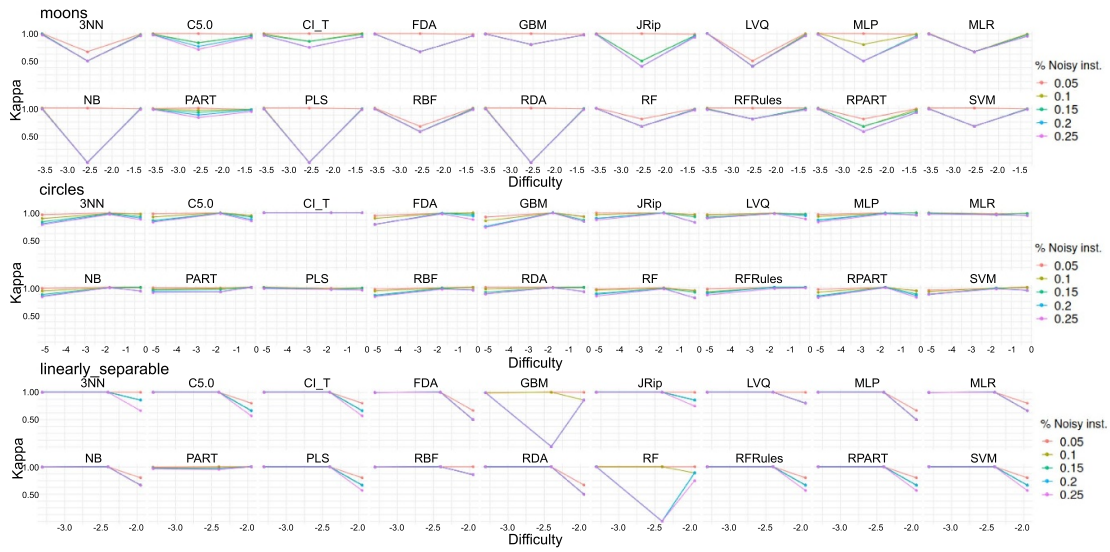


Figure A3. SCC for the benchmarks.

Appendix B. Original class distribution per dataset and difficulty bin

The figure B1 shows the original class distribution for each dataset. In addition, the figure B2 shows the class distribution for each bin of difficulty for all datasets. These figures allow us to have an insight into the class imbalance that may occur for different levels of difficulty in the datasets, and how this may affect the performance of the models.

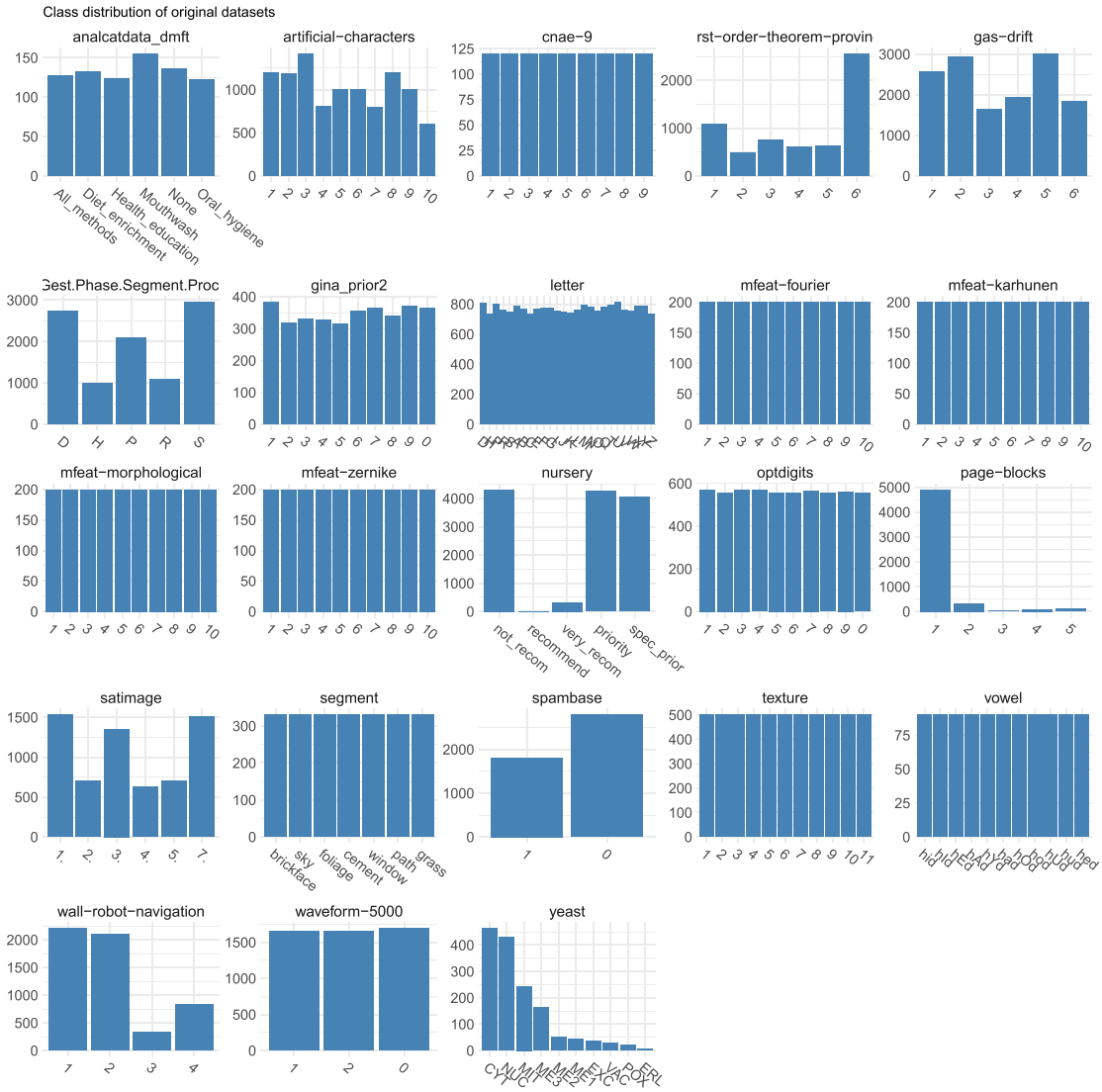


Figure B1. Class distribution of the benchmarks (real label).

(a) Class distribution of the benchmarks per difficulty bin (1/2).

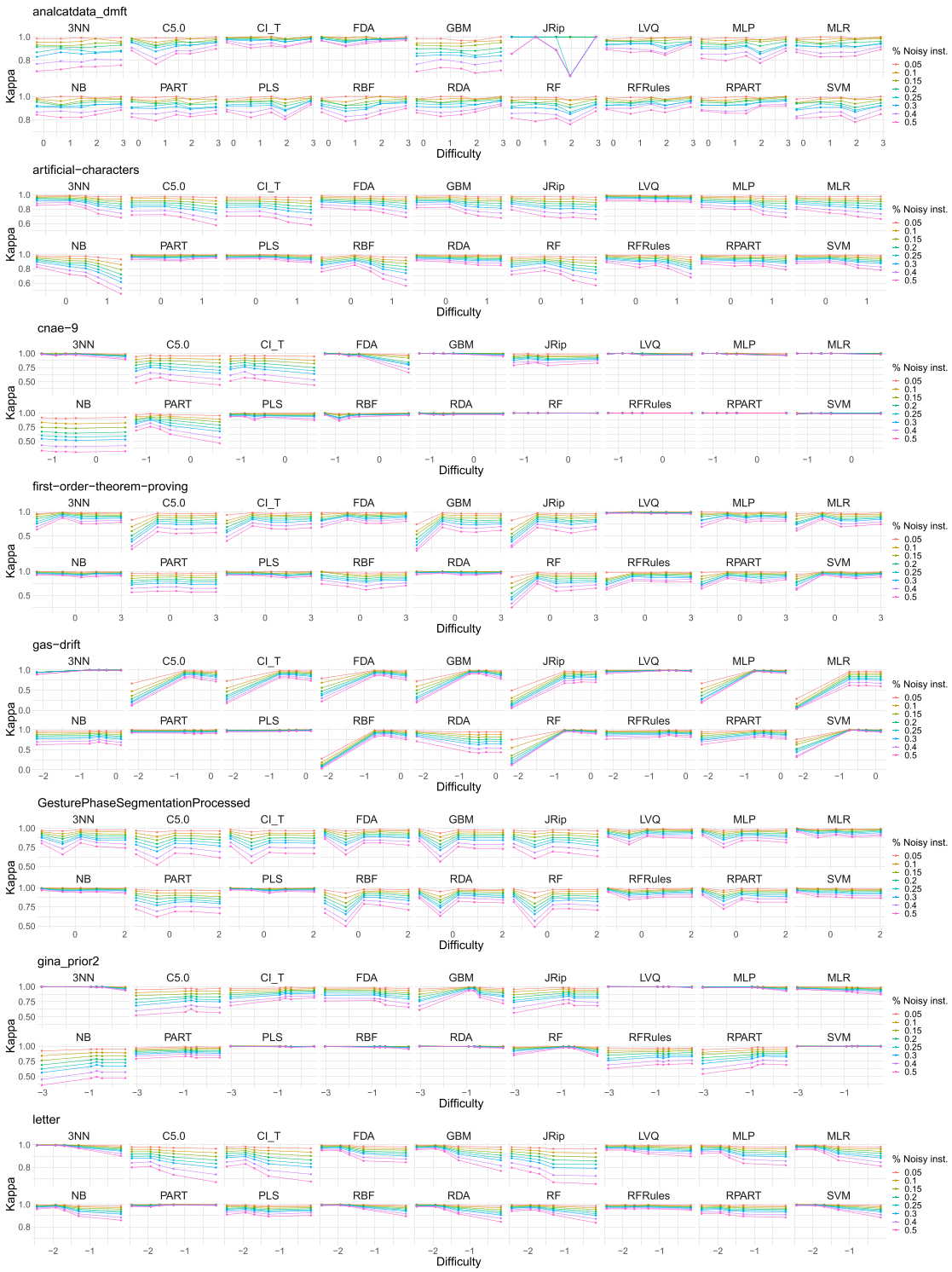
Figure B2. Class distribution of the benchmarks per difficulty bin.

(b) Class distribution of the benchmarks per difficulty bin (2/2).

Figure B2. (Continued.)

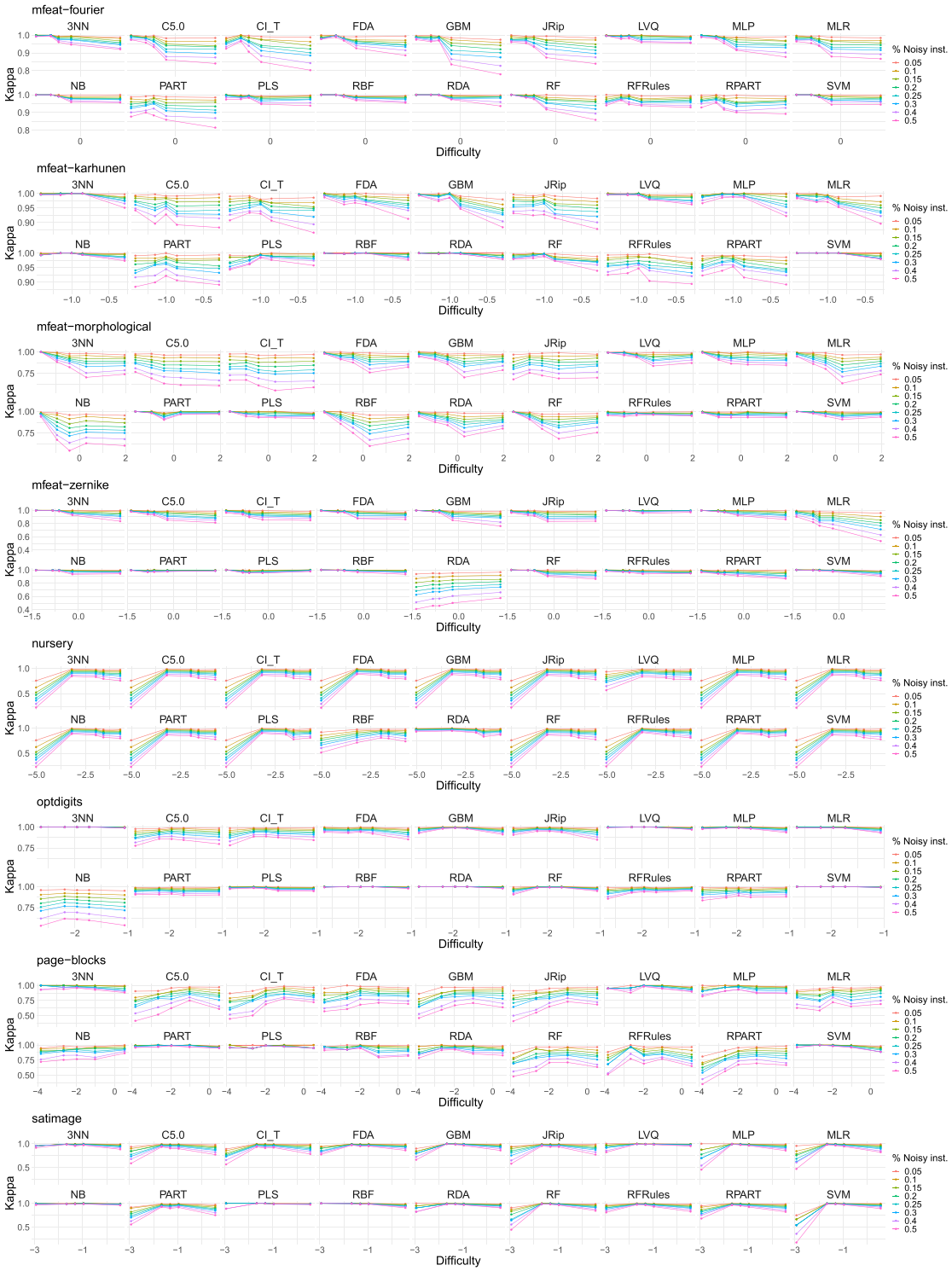
Appendix C. System characteristic curves of all models and datasets

The SCCs allow us to visually compare the models and gain insight into their strengths and weaknesses in presence of noise and instance difficulty. Our analysis of the SCCs in the figures C1–C3 shows that models do not always perform similarly on different datasets. Some models that perform well on one dataset may not perform as well on another. This highlights the impact that other factors, such as the number of instances, features, and classes can have on robustness. These factors are regarded by the complexity measure that we explained in section 5.4.



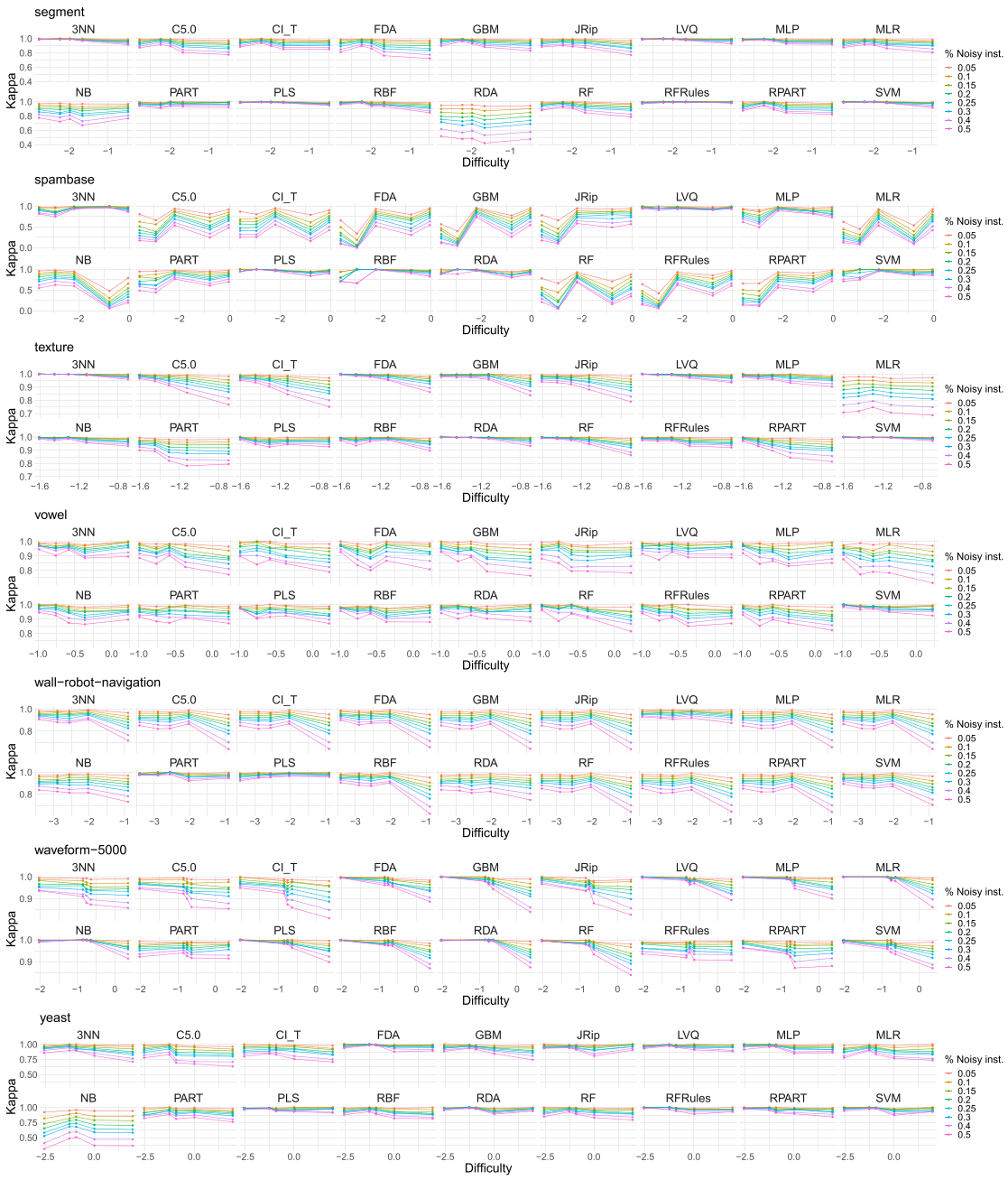
(a) The SCCs obtained from all the datasets and models (1/3).

Figure C1. The SCCs obtained from all the datasets and models.



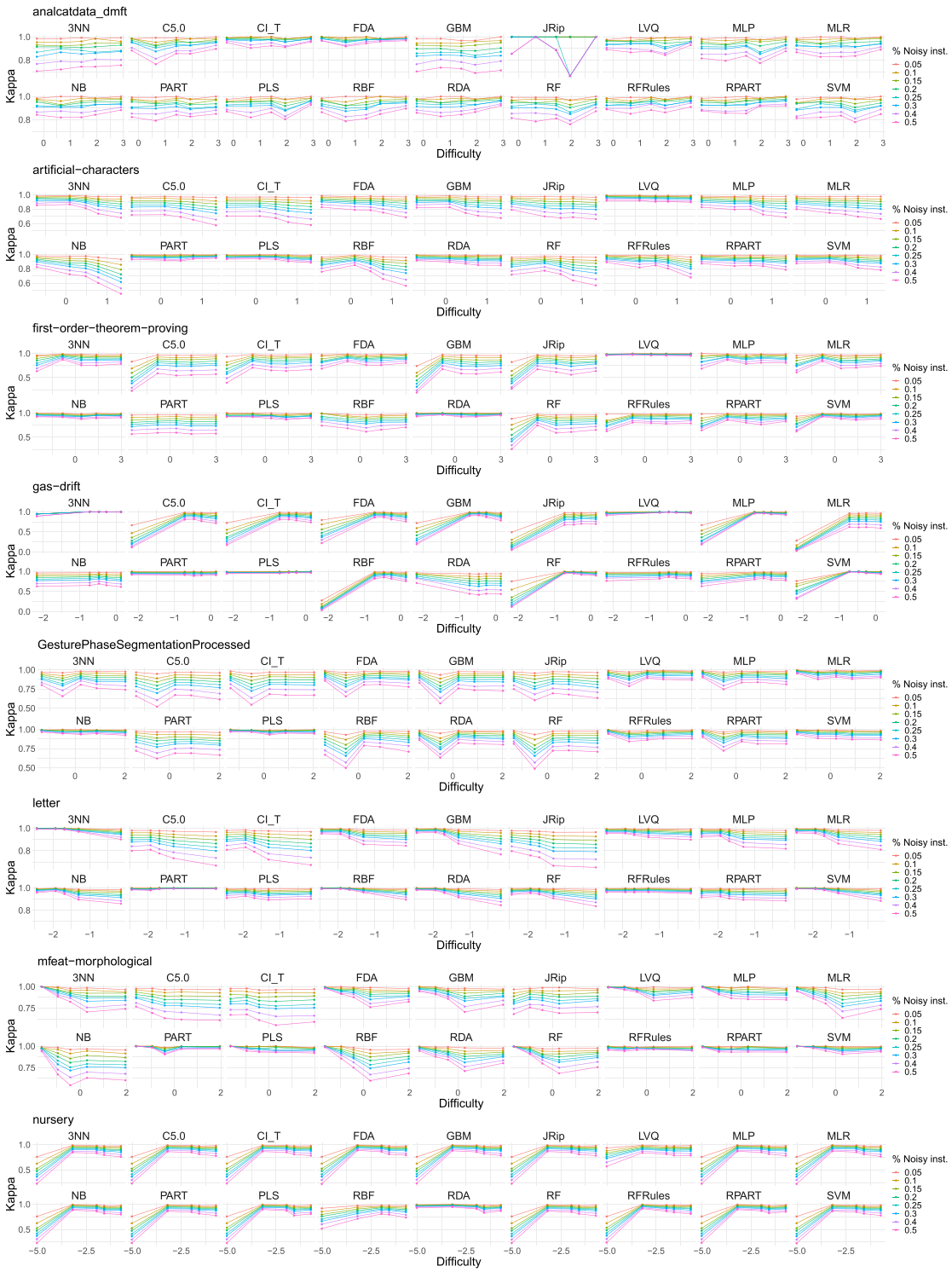
(b) The SCCs obtained from all the datasets and models (2/3).

Figure C1. (Continued.)



(c) The SCC obtained from all the datasets and models (3/3).

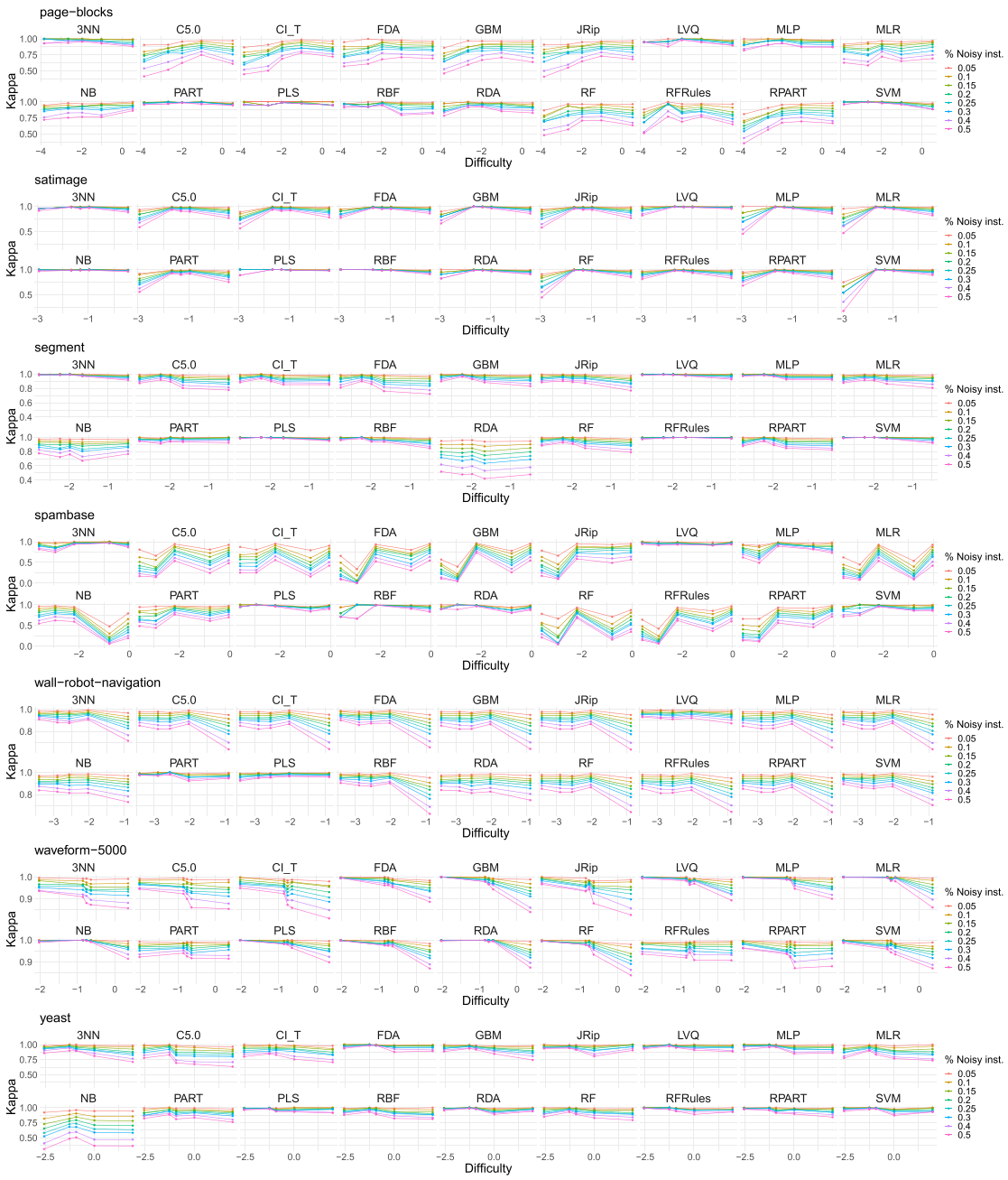
Figure C1. (Continued.)



(a) The SCC obtained from the simple datasets (1/2).

Figure C2. The SCCs obtained from the simple datasets.

The figures C2 and C3 show the results of the SCCs for simple and complex datasets, respectively. In general, it is observed that simple datasets are more sensitive to noise and, thus, models perform worse as more instances are noisy perturbed. This observation also suggests that for simple datasets, models tend to overfit the data. The noise tend be more influential in specific difficulty bins, i.e. the same amount of noise



(b) The SCC obtained from the simple datasets (2/2).

Figure C2. (Continued.)

causes higher variations in some particular difficulty bins, while the others are less affected by noise. This suggest that the interaction between noise and instances difficulty is complex to disentangle. The methods explained in sections 3.4 and 3.5 attempt to shed light on this challenge.

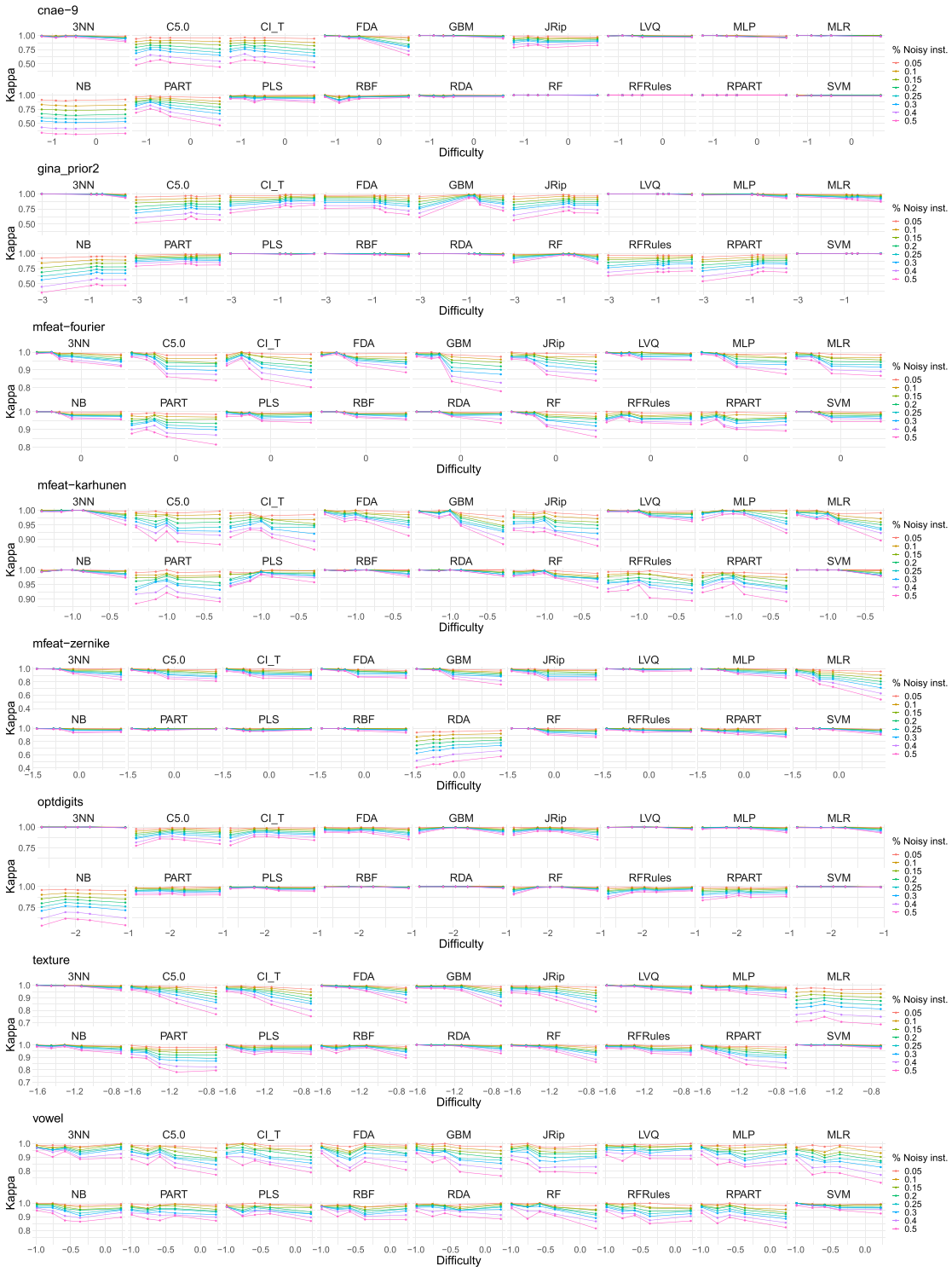


Figure C3. The SCC obtained from the complex datasets.

ORCID iDs

R Fabra-Boluda https://orcid.org/0000-0003-2954-2041
F Martínez-Plumed https://orcid.org/0000-0003-2902-6477

References

[1] Sarker I H 2021 Machine learning: algorithms, real-world applications and research directions *SN Comput. Sci.* **2** 160
[2] Li B, Qi P, Liu B, Di S, Liu J, Pei J, Yi J and Zhou B 2023 Trustworthy AI: from principles to practices *ACM Comput. Surv.* **55** 1–46

[3] Goodfellow I, McDaniel P and Papernot N 2018 Making machine learning robust against adversarial inputs *Commun. ACM* **61** 56–66
[4] Wang Y, Ma X, Bailey J, Yi J, Zhou B and Gu Q 2021 On the convergence and robustness of adversarial training *Int. Conf. on Machine Learning* vol 97 (PMLR) pp 6586–95
[5] Lian J, Freeman L, Hong Y and Deng X 2021 Robustness with respect to class imbalance in artificial intelligence classification algorithms *J. Qual. Technol.* **53** 505–25
[6] Martínez-Plumed F, Prudêncio R B C, Martínez-Usó A and Hernández-Orallo J 2019 Item response theory in AI: analysing machine learning classifiers at the instance level *Artif. Intell.* **271** 18–42
[7] Hambleton R K and Swaminathan H 2013 *Item Response Theory: Principles and Applications* (Springer)
[8] Zhang J M, Harman M, Ma L and Liu Y 2020 Machine learning testing: survey, landscapes and horizons *IEEE Trans. Softw. Eng.* **48** 1–36
[9] Ljunggren D and Ishii S 2021 A comparative analysis of robustness to noise in machine learning classifiers *Student Thesis* DiVA KTH Royal Institute of Technology
[10] Zhu X and Wu X 2004 Class noise vs. attribute noise: a quantitative study *Artif. Intell. Rev.* **22** 177–210
[11] Sáez J A, Luengo J and Herrera F 2016 Evaluating the classifier behavior with noisy data considering performance and robustness: the equalized loss of accuracy measure *Neurocomputing* **176** 26–35
[12] Wu X and Zhu X 2008 Mining with noise knowledge: error-aware data mining *IEEE Trans. Syst. Man Cybern.* A **38** 917–32
[13] Ripley B 2008 *Pattern Recognition And Neural Networks* vol 11 (Cambridge University Press)
[14] Sáez J A, Galar M, Luengo J and Herrera F 2014 Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition *Knowl. Inf. Syst.* **38** 179–206
[15] Ferri C, Hernández-Orallo J and Modroiu R 2009 An experimental comparison of performance measures for classification *Pattern Recognit. Lett.* **30** 27–38
[16] Moreno-Barea F J, Jerez J M and Franco L 2020 Improving classification accuracy using data augmentation on small data sets *Expert Syst. Appl.* **161** 113696
[17] Zhu X, Wu X and Chen Q 2003 Eliminating class noise in large datasets *Proc. 20th Int. Conf. on Machine Learning (ICML-03)* pp 920–7
[18] Teng C-M 1999 Correcting noisy data *Int. Conf. on Machine Learning* pp 239–48
[19] Zur R M, Jiang Y, Pesce L L and Drukker K 2009 Noise injection for training artificial neural networks: a comparison with weight decay and early stopping *Med. Phys.* **36** 4810–8
[20] Krizhevsky A, Sutskever I and Hinton G E 2017 Imagenet classification with deep convolutional neural networks *Commun. ACM* **60** 84–90
[21] Bastani O, Ioannou Y, Lampropoulos L, Vytiniotis D, Nori A and Criminisi A 2016 Measuring neural net robustness with constraints *Advances in Neural Information Processing Systems* vol 29
[22] Madaan D, Shin J and Ju Hwang S 2021 Learning to generate noise for multi-attack robustness *Int. Conf. on Machine Learning* (PMLR) pp 7279–89
[23] Li B, Chen C, Wang W and Carin L 2019 Certified adversarial robustness with additive noise *Advances in Neural Information Processing Systems* vol 32
[24] Arslan M, Guzel M, Demirci M and Ozdemir S 2019 SMOTE and Gaussian noise based sensor data augmentation *2019 4th Int. Conf. on Computer Science and Engineering (UBMK)* (IEEE) pp 1–5
[25] Latif S, Rana R and Qadir J 2018 Adversarial machine learning and speech emotion recognition: utilizing generative adversarial networks for robustness (arXiv:1811.11402)
[26] Yi L and Mak M-W 2020 Improving speech emotion recognition with adversarial data augmentation network *IEEE Trans. Neural Netw. Learn. Syst.* **33** 172–84
[27] Latif S, Asim M, Rana R, Khalifa S, Jurdak R and Schuller B W 2020 Augmenting generative adversarial networks for speech emotion recognition (arXiv:2005.08447)
[28] Leistner C, Saffari A, Roth P M and Bischof H 2009 On robustness of on-line boosting-a competitive study *2009 IEEE 12th Int. Conf. on Computer Vision Workshops, (ICCV Workshops)* (IEEE) pp 1362–9
[29] Zhang J M, Harman M, Guedj B, Barr E T and Shawe-Taylor J 2021 Perturbation validation: a new heuristic to validate machine learning models (arXiv:1905.10201)
[30] Tjeng V, Xiao K and Tedrake R 2017 Evaluating robustness of neural networks with mixed integer programming (arXiv:1711.07356)
[31] Gehr T, Mirman M, Drachsler-Cohen D, Tsankov P, Chaudhuri S and Vechev M 2018 AI: safety and robustness certification of neural networks with abstract interpretation *2018 IEEE Symp. on Security and Privacy (SP)* (IEEE) pp 3–18
[32] Gopinath D, Wang K, Zhang M, Pasareanu C S and Khurshid S 2018 Symbolic execution for deep neural networks (arXiv:1807.10439)
[33] Katz G, Barrett C, Dill D L, Julian K and Kochenderfer M J 2017 Reluplex: an efficient SMT solver for verifying deep neural networks *Int. Conf. on Computer Aided Verification* (Springer) pp 97–117
[34] Usman M, Noller Y, Păsăreanu C S, Sun Y and Gopinath D 2021 NeuroSPF: a tool for the symbolic analysis of neural networks *2021 IEEE/ACM 43rd Int. Conf. on Software Engineering: Companion Proc. (ICSE-Companion)* (IEEE) pp 25–28
[35] Smith M R, Martinez T and Giraud-Carrier C 2014 An instance level analysis of data complexity *Mach. Learn.* **95** 225–56
[36] Russakovsky O *et al* 2015 Imagenet large scale visual recognition challenge *Int. J. Comput. Vis.* **115** 211–52
[37] Liu D, Xiong Y, Pulli K and Shapiro L 2011 Estimating image segmentation difficulty *Int. Workshop on Machine Learning and Data Mining in Pattern Recognition* (Springer) pp 484–95
[38] Vijayanarasimhan S and Grauman K 2009 What's it going to cost you?: predicting effort vs. informativeness for multi-label image annotations *2009 IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE) pp 2262–9
[39] Richards B 1987 Type/token ratios: what do they really tell us? *J. Child Lang.* **14** 201–9
[40] Hoover D L 2003 Another perspective on vocabulary richness *Comput. Humanit.* **37** 151–78
[41] Martínez-Plumed F, Prudêncio R B C, Martínez-Usó A and Hernández-Orallo J 2016 Making sense of item response theory in machine learning *ECAI 2016—22nd European Conf. on Artificial Intelligence* pp 1140–8
[42] Martínez-Plumed F and Hernández-Orallo J 2020 Dual indicators to analyse AI benchmarks: difficulty, discrimination, ability and generality *IEEE Trans. Games* **12** 121–31
[43] Lalor J P 2020 Learning latent characteristics of data and models using item response theory *PhD Thesis* University of Massachusetts
[44] Chen Z and Ahn H 2020 Item response theory based ensemble in machine learning *Int. J. Autom. Comput.* **17** 621

[45] Birnbaum A 1968 Some latent trait models and their use in inferring an examinee's ability *Statistical Theories of Mental Test Scores* (Addison-Wesley)
[46] Martínez-Plumed F, Castellano-Falcón D, Monserrat C and Hernández-Orallo J 2022 When AI difficulty is easy: the explanatory power of predicting IRT difficulty *Proc. AAAI Conf. on Artificial Intelligence*
[47] Hernández Orallo J, Ferri Ramírez C and Ramírez Quintana M J 2004 *Introducción a la Minería de Datos* (Pearson Prentice Hall)
[48] Flach P 2012 *Machine Learning: the art and Science of Algorithms That Make Sense of Data* (Cambridge University Press)
[49] Fernández-Delgado M, Cernadas E, Barro S and Amorim D 2014 Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **15** 3133–81
[50] Fabra-Boluda R, Ferri C, Martínez-Plumed F, Hernández-Orallo J and Ramírez-Quintana M J 2020 Family and prejudice: a behavioural taxonomy of machine learning techniques *ECAI 2020* (IOS Press) pp 1135–42
[51] Landis R and Koch G 1977 An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers *Biometrics* **33** 363–74
[52] Wärnling O and Bissmark J 2017 The sparse data problem within classification algorithms: the effect of sparse data on the NaïVe Bayes algorithm *Student Thesis* DiVA
[53] Fabra-Boluda R, Ferri C, Hernández-Orallo J, Martínez-Plumed F and Ramírez-Quintana M J 2018 Modelling machine learning models *Philosophy and Theory of Artificial Intelligence 2017* (Springer) pp 175–86
[54] Fabra-Boluda R, Ferri C, Hernández-Orallo J, Martínez-Plumed F and Ramírez-Quintana M J 2018 Identifying the machine learning family from black-box models *Advances in Artificial Intelligence: 18th Conf. of the Spanish Association for Artificial Intelligence, CAEPIA 2018, (Granada, Spain, 23–26 October 2018), Proc. 18* (Springer) pp 55–65
[55] Wright B D and Stone M H 1979 *Best Test Design* (Mesa Press)
[56] Ferri C, Hernández-Orallo J, Martínez-Usó A and Ramírez-Quintana M J 2014 Identifying dominant models when the noise context is known *First Workshop on Generalization and Reuse of Machine Learning Models Over Multiple Contexts*
[57] Lin S-L 2021 Application of machine learning to a medium gaussian support vector machine in the diagnosis of motor bearing faults *Electronics* **10** 2266
[58] Jin X and Hirakawa K 2013 Approximations to camera sensor noise *Image Processing: Algorithms and Systems XI* vol 8655 (SPIE) pp 149–55
[59] Braun S, Neil D and Liu S-C 2017 A curriculum learning method for improved noise robustness in automatic speech recognition *2017 25th European Signal Processing Conf. (EUSIPCO)* (IEEE) pp 548–52
[60] Abadi M, Chu A, Goodfellow I, McMahan H B, Mironov I, Talwar K and Zhang Li 2016 Deep learning with differential privacy *Proc. 2016 ACM SIGSAC Conf. on Computer and Communications Security* pp 308–18
[61] Kuhn M 2008 Building predictive models in R using the caret package *J. Stat. Softw.* **28** 1–26
[62] Chalmers R P 2012 mirt: a multidimensional item response theory package for the R environment *J. Stat. Softw.* **48** 1–29
[63] van Rijn J N, Bischl B, Torgo L, Gao B, Umaashankar V, Fischer S, Winter P, Wiswedel B, Berthold M R and Vanschoren J 2013 OpenML: a collaborative science platform *Machine Learning and Knowledge Discovery in Databases* (Springer) pp 645–9
[64] Vanschoren J, Van Rijn J N, Bischl B and Torgo L 2014 OpenML: networked science in machine learning *ACM SIGKDD Explor. Newsl.* **15** 49–60
[65] Blanco-Vega R, Hernández-Orallo J and Ramírez-Quintana M J 2004 Analysing the trade-off between comprehensibility and accuracy in mimetic models *Int. Conf. on Discovery Science* (Springer) pp 338–46
[66] Kaushik D, Kiela D, Lipton Z C and Yih W-tau 2021 On the efficacy of adversarial data collection for question answering: results from a large-scale randomized study (arXiv:2106.00872)
[67] Wallace E, Williams A, Jia R and Kiela D 2022 Analyzing dynamic adversarial training data in the limit *Findings of the Association for Computational Linguistics* (Association for Computational Linguistics) pp 202–17
[68] Mirkin B 2011 Choosing the number of clusters *Wiley Interdiscip. Rev.-Data Min. Knowl. Discovery* **1** 252–60
[69] Rousseeuw P J 1987 Silhouettes: a graphical aid to the interpretation and validation of cluster analysis *J. Comput. Appl. Math.* **20** 53–65
[70] Thorndike R L 1953 Who belongs in the family? *Psychometrika* **18** 267–76
[71] Wang X *et al* 2021 Textflint: unified multilingual robustness evaluation toolkit for natural language processing *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int. Joint Conf. on Natural Language Processing: System Demonstrations* pp 347–55
[72] Petit B S M F F K J, Stottelaar B, Feiri M and Kargl F 2015 Remote attacks on automated vehicles sensors: experiments on camera and LiDAR *Black Hat Europe*
[73] Fabra-Boluda R 2023 *ML Robustness Difficulty* (available at: https://github.com/rfabra/ml-robustness-difficulty)