Pattern Recognition Letters 201 (2026) 29–36

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Generalization performance distributions along learning curves

O. Taylan Turan [a,*], Marco Loog [a,b], David M.J. Tax [a]

[a] Pattern Recognition and Bioinformatics Laboratory, Delft University of Technology, Van Mourik Broekmanweg 6, Delft, 2628 XE, The Netherlands
[b] Institute for Computing and Information Sciences, Radboud University, Toernooiveld 212, Nijmegen, 6525 EC, The Netherlands

ARTICLE INFO

Editor: Prof. S. Sarkar

Keywords:
Learning curve
Generalization performance
Model selection
Hyper-parameter tuning
Quantiles

ABSTRACT

Learning curves show the expected performance with respect to training set size. This is often used to evaluate and compare models, tune hyper-parameters and determine how much data is needed for a specific performance. However, the distributional properties of performance are frequently overlooked on learning curves. Generally, only an average with standard error or standard deviation is used. In this paper, we analyze the distributions of generalization performance on the learning curves. We compile a high-fidelity learning curve database, both with respect to training set size and repetitions of the sampling for a fixed training set size. Our investigation reveals that generalization performance rarely follows a Gaussian distribution for classical classifiers, regardless of dataset balance, loss function, sampling method, or hyper-parameter tuning along learning curves. Furthermore, we show that the choice of statistical summary, mean versus measures like quantiles affect the top model rankings. Our findings highlight the importance of considering different statistical measures and use of non-parametric approaches when evaluating and selecting machine learning models with learning curves.

1. Introduction

In (supervised) machine learning, the goal is often to optimize the expected performance of a model. For optimizing the expected performance, samples are assumed to be drawn from some fixed distribution. However, in practice, we do not have access to this distribution. Instead, we rely on training and testing sets obtained from finite datasets, which introduces stochasticity. Factors such as, initialization or optimization procedures, further contribute to the variability of model performance.

To account for stochasticity, performance is most often summarized using the average [1]. While averages are useful in many problems, they are not always sufficient. If the performance distribution is highly skewed, exhibits heavy tails, or contains substantial outliers, the mean may fail to capture the underlying behavior. In such cases, more robust estimators such as quantiles can provide a clearer picture. For example, in high-risk settings, one may prefer a more conservative or, conversely, a more optimistic estimate than the mean. In finance, the Value-at-Risk measure is widely used to quantify potential losses at a specified probability level [2], and in medical applications such as drug discovery, higher quantiles can be used to evaluate promising candidates [3].

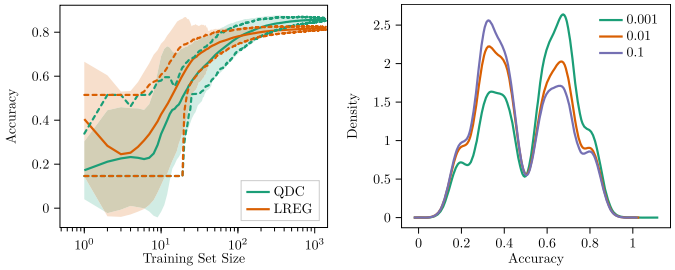
Average performance is used for obtaining learning curves, model selection and hyper-parameter tuning. Although generally non-parametric tests are available, often parametric ones are utilized without being able to test the assumptions of it. For instance, a set of performance estimates are often used with (paired) t-tests to determine if the expected performance of models are significantly different with limited number of samples. This prevents accurately testing if actually the individual model performance or their differences follow a Gaussian distribution or not [4]. This assumption of Gaussianity is often justified by viewing each prediction as a Bernoulli trial, leading to a Binomial distribution over errors. For large and fixed test sets, the Binomial distribution is assumed to converge asymptotically to a Gaussian [5]. However, when we use finite datasets, this assumption may not hold.

To highlight that generalization performance along learning curves might be complex, we show an example on a OpenML-46597 classification problem in Fig. 1(a). Here, the generalization accuracy of Quadratic Discriminant classifier (QDC) and Logistic Regression classifier (LREG) without regularization are plotted against training-set size (n), with the solid lines representing the average performance, and the dotted lines representing 5% and 95% quantiles. The shaded area indicates the two standard deviations away from the average performance. If we look at average performance, QDC is preferred over LREG for a training set size around $n > 100$. When the 5% quantile is used for comparison QDC has higher accuracy for $n > 200$. However, if we choose the 90% quantile, the threshold where one chooses QDC over LREG earlier $n > 80$.

This issue does not only pertain to the learning curves, but can also affect other sub-sampling based generalization performance estimation strategies. Fig. 1(b) shows the accuracy distribution of a multi-layer perceptron (5 hidden layers with 16 neurons) resulting from 20-fold cross-validation repeated 1000 times on a OpenML-1046 classification

* Corresponding author.
E-mail address: o.t.turan@tudelft.nl (O.T. Turan).

https://doi.org/10.1016/j.patrec.2026.01.003
Received 4 September 2025; Received in revised form 18 November 2025; Accepted 1 January 2026
Available online 3 January 2026



(a) Learning curves of Logistic Regression and Quadratic Discriminant models for OpenML-46597.Experiments are repeated 1000 times

(b) Multi-layer Perceptron with 20-fold cross-validation repeated 1000 times on OpenML-1046 for various learning rates.

Fig. 1. Performance Distributions for learning curve and cross-validation.

task. We can clearly observe a bimodal distribution for generalization performance, one below 0.5 accuracy level and one above. This means that some models resulting from the cross-validation procedure perform worse than random chance and some not. Across different learning rates, the dominant mode of the performance distribution can change. When conducting cross-validation with a limited computational budget, one may miss some of these modes and end up selecting a suboptimal learning rate.

In this paper, we systematically analyze the distributions of generalization performance obtained for learning curves across a wide range of models, datasets, sampling strategies, and evaluation metrics. To this end, we create a high-fidelity learning curve database (lcdb++)[1] We refer to it as high-fidelity since the biases introduced by predefined sampling grids or the limited number of repetitions that affect other databases [6,7]. We examine the extent to which the common Gaussianity assumption holds for this database and explore how factors such as dataset balance, model type, loss functions, multi-class extensions, hyper-parameter tuning, and sampling strategies influence this. Finally, we assess the practical impact of these deviations by comparing model selection outcomes when using traditional statistics (mean and standard deviation) versus alternative measures such as quantiles.

2. Experimental setup

We investigate the generalization performance along learning curves, which show generalization performance with respect to training set size. In this section, we formalize our definition of a learning curve and introduce how we constructed our learning curve database.

2.1. Generalization performance along learning curves

Let us denote the input and output spaces as $\mathbb{X}$ and $\mathbb{Y}$, respectively. A learning algorithm $\mathcal{A}$ takes a training set $\mathcal{D}_n := (x_i, y_i)_{i=1}^n$, which is sampled from a dataset containing i.i.d. samples $\mathcal{D}_N := (x_i, y_i)_{i=1}^N$ from unknown distribution $\mathcal{P}(x, y)$ over $\mathbb{X} \times \mathbb{Y}$. Providing this algorithm with training data result in a hypothesis h from a class $\mathbb{H}$: $h_n := \mathcal{A}(\mathcal{D}_n) \in \mathbb{H}$. Note that, producing a hypothesis can involve hyper-parameter tuning as well. Then, the prediction of a learner can be represented as $\hat{y} = h_n(x) \in \mathbb{Y}$. The performance of the learner is measured by a loss function $\mathcal{L}(y, \hat{y})$. The expected performance $\mathcal{R}$ of a hypothesis h over the true distribution $\mathcal{P}(x, y)$ is given by:

$$\mathcal{R}(h_n) = \int \mathcal{L}(y, \hat{y}) \mathcal{P}(x, y) dx dy \tag{1}$$

The true performance in Eq. (1) is attainable only when we have access to $\mathcal{P}(x, y)$ and when the integral is tractable. In practice, the risk is usually estimated through cross-validation, bootstrapping, or other related methods [8]. A learning curve can then be obtained by increasing the training set size n, repeatedly sampling training sets $\mathcal{D}_n$ from a finite dataset $\mathcal{D}_N$, and computing the average risk

$$\bar{\mathcal{R}}_n = \mathop{\mathbb{E}}_{\mathcal{D}_n \sim \mathcal{D}_N} \mathcal{R}(\mathcal{A}(\mathcal{D}_n)). \tag{2}$$

2.2. High-fidelity learning curve database

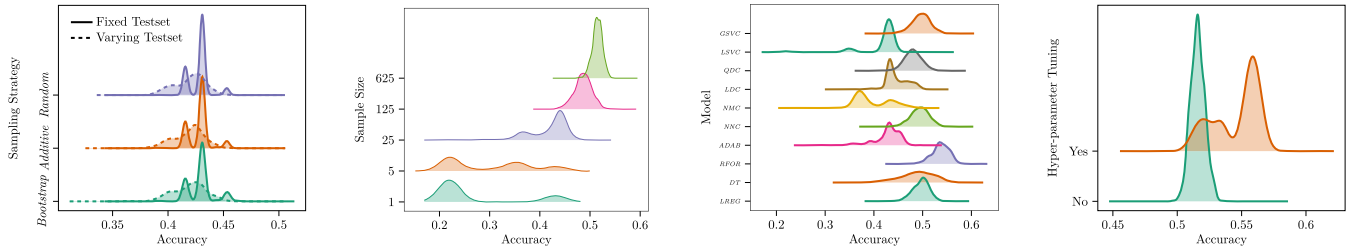
When constructing learning curves, two key decisions must be made: the selection of training set sizes used to evaluate generalization performance, and the number of repetitions performed for each size. Existing learning curve databases [6,7] typically employ only 25 repetitions per training set size. However, this number is insufficient to obtain reliable distributions of generalization performance, which is the focus of our study. To address this, we repeat our experiments 1000 times. This approach allows us not only to analyze average performance but also to estimate additional statistical measures, such as quantiles, with greater accuracy. Furthermore, previous databases rely on a sparse grid of training set sizes, whereas we consider every possible training set size for each dataset. In addition, unlike prior work, we account for different sampling and splitting strategies as well as hyperparameter optimization.

The database contains the generalization performance of Logistic Regression (LREG), Linear and Gaussian Kernel Support Vector Machine (LSVC, GSVC), Linear Discriminant and Quadratic Discriminant (LDC, QDC), Nearest Mean (NMC), Decision Tree (DT), Random Forest (RFOR), AdaBoost (ADAB) classifiers. Since some of these methods do not support multi-class classification we use one-vs-rest for generalization performance estimates. In our experiments we investigated 10 datasets from OpenML database [9]. We utilize the four widely used performance measures for classification tasks. Area Under the receiver-operating characteristic curve (AUC), Brier Loss (BRI), Accuracy (ACC) and Cross-Entropy Loss (CE) . Note that AUC is only for binary classification, hence we use one-vs-rest strategy to generalize two-class classifiers to multi-class classifiers [10]. Every training set size in the range $n \in [1, 0.7 * N]$ for fixed models and $n \in [10, 0.7 * N]$ for hyper-parameter tuned learning curves is used. Moreover, for every training set size we repeat the experiments 1000 times.

To investigate the effect of splitting and sampling schemes in learning curve generation, we consider the most common approaches. One way to obtain a learning curve is by using all available data, while another is by separating a test set at the beginning [8]. If the whole dataset is used, the size of the test set decreases as the number of training samples increases. We refer to this approach as *Varying Testset*. In contrast, when the dataset is split at the start, we call it *Fixed Testset*. To obtain multiple training datasets for a given training size, several sampling strategies may be employed, such as input density preserving sampling, cross-validation, random sampling, or the bootstrap method [11]. In our study, we focus on the simplest settings to examine the effect of sample replacement and sample dependence. Specifically, we use *Random* sampling, where samples are drawn without replacement; *Bootstrap* sampling, where samples are drawn with replacement; and *Additive* sampling, where random samples are drawn without replacement and incrementally added to the training dataset. A visual summary of the possible samplings is shown in Fig. A.4, and information about the OpenML datasets used can be found in Table A.8.

To maintain a manageable computational load, we limited hyper-parameter tuning to a single parameter for models with multiple hyper-parameters, specifically ADAB (maximum iterations), DT (minimum leaf size), and RFOR (number of trees). For each hyper-parameter, we conducted a grid search over 20 values. Continuous hyper-parameters were sampled logarithmically between 10^{-8} and 10^{-1}, while discrete hyper-parameters were sampled uniformly between 1 and 100. All experiments were performed using our in-house learning curve generation tool [12].

[1] Our database can be found at https://surfdrive.surf.nl/files/index.php/s/eJCTZ5n9rG72k9w.



(a) Different sampling strategies at sample size 500 for ADAB.

(b) Different sample sizes for QDC $n = (1, 5, 25, 125, 625)$.

(c) Different models at sample size 100.

(d) Hyper-parameter tuning QDC at sample size 900.

Fig. 2. Investigating the performance distribution in various settings for OpenML-1063 dataset with fixed testset.

Table 1
Percentage of non-normal performance distributions for each loss measure and dataset.

	11	23	37	53	61	1063	44037	46597	46733	46847
Accuracy	98.16 ± 6.28%	99.75 ± 0.46%	97.63 ± 6.68%	94.85 ± 13.75%	99.38 ± 2.47%	93.77 ± 9.98%	91.95 ± 16.79%	88.10 ± 19.30%	97.76 ± 9.73%	94.66 ± 15.14%
AUC	92.27 ± 12.85%	97.14 ± 5.96%	93.57 ± 21.84%	82.15 ± 28.38%	91.51 ± 13.61%	88.82 ± 15.25%	87.70 ± 21.64%	86.47 ± 16.13%	97.54 ± 4.82%	86.63 ± 23.61%

Table 2
Percentage of non-normal performance distributions for each loss measure and model.

	ADAB	DT	GSVC	LDC	LREG	LSVC	NMC	NNC	QDC	RFOR
Accuracy	99.58 ± 2.21%	97.90 ± 5.12%	97.25 ± 8.14%	99.14 ± 2.33%	95.99 ± 8.43%	92.87 ± 18.94%	97.91 ± 6.04%	92.79 ± 13.59%	85.64 ± 22.16%	98.07 ± 5.53%
AUC	94.55 ± 5.12%	93.56 ± 7.98%	96.09 ± 10.37%	88.83 ± 14.34%	94.99 ± 9.73%	97.28 ± 8.82%	89.70 ± 22.09%	85.07 ± 19.30%	69.43 ± 35.53%	95.58 ± 8.84%

3. Results

In this section, we investigate if the generalization performance distributions follow a Gaussian distribution and the effect of it. This investigation is conducted from various aspects, including model family, performance metric, and sampling strategy. Samples of generalization performance distributions from our learning curve database are presented in Fig. 2, along with density estimations of the observed generalization performance for several cases. Results in this section pertain only to Accuracy and AUC. Furthermore, unless pairwise comparisons are not mentioned the whole database is used, for the paired investigations 3 missing dataset results are excluded. The corresponding results for other loss measures are presented in Appendix D.

The effect of sampling strategies is visualized in Fig. 2(a), using a fixed learner ADAB and OpenML-1063 dataset. While the overall variation between sampling methods are not substantial, it is apparent that the fixing the test set size can shift the current modes and add other modes to the generalization performance. Since we do not see a significant difference between the sampling strategies, we use the *Additive* sampling strategy in the rest of the Fig. 2 with the *Fixed Testset* as that is the mostly used setting in supervised machine learning applications.

Fig. 2(b) shows slices of training set sizes versus generalization performance for a fixed QDC model for OpenML-1063 dataset, and the accuracy performance metric. As seen in the figure, when the training set size is small ($n = 1$) performance distribution has two distinct modes. As the sample size increases, the distribution of generalization performance changes significantly. So, the generalization performance distribution changes with respect to training set size.

For the same dataset and training set size, different models have completely different performance distributions. In Fig. 2(c), we plot performance distributions for a fixed sample size ($n = 100$) and a fixed dataset across all models. It is evident that not only does the average of the performance distributions vary, but the shapes of the distributions also differ. LSVC displays a heavy-tailed performance distribution, NMC has bi-modal distribution, while DT has a more uniform spread. In contrast, the ADAB exhibits a multiple modes that are close to each other.

Tuning the hyper-parameters for the same sample size of one model (QDC in this case) can also alter the generalization performance distribution compared to fixed model. As shown in Fig. 2(d), although the average of the performance improves with tuning, it creates another mode that is close to the not tuned version performance, indicating that tuning can still lead to poorly performing models in some cases.

As illustrated in Fig. 2, the distribution of generalization performance can vary significantly depending on how learning curves are obtained, and may deviate from Gaussianity. In the following section, we qualitatively investigate how frequently this deviation occurs using our learning curve database, derived from various real-world datasets.

3.1. Deviations from the Gaussian distribution

Across all loss functions and datasets, we investigate, for each training set size, whether the distribution of generalization performance is Gaussian along learning curves. Approximately 94% of the generalization performance distributions are found to be non-Gaussian according to the Shapiro-Wilk test [13,14] with significance level of $\alpha = 0.05$. To further validate these results, we also conducted D'Agostino and Pearson's normality test [15] at the same significance level, which identified 89% of the distributions in our dataset as non-Gaussian. The close agreement between the two tests reinforces that generalization performance distributions rarely follow a Gaussian form in our learning curve database for classical learners. For the remainder of this paper, all normality analyses are performed using the Shapiro-Wilk test.

In Table 1, we present the percentages of non-Gaussian generalization performance distributions, along with corresponding standard deviations. For nearly all datasets, the accuracy metric consistently shows the lowest level of Gaussianity, with only around 5% of distributions detected as Gaussian. This shows that it is rare to find performance distributions that are Gaussian on our database of generalization performance distributions.

Among all datasets, OpenML-23 exhibits the lowest proportion of Gaussian distributions, approximately 2%. This dataset is imbalanced, containing a different number of samples across all available classes. In contrast, OpenML-46597, which is balanced (with the same number of samples per class), exhibits the highest percentage of Gaussian cases.

To better investigate the effect of dataset imbalance, Table 3 also reports our statistics separately for balanced and imbalanced datasets.

Table 3
Effect of dataset imbalance to the Gaussianity.

	Imbalanced	Balanced
Accuracy	96.95%	93.57%
AUC	92.66%	86.96%

Table 4
Percentage of non-normal and loss measure for binary and multi-class cases of LREG.

	Binary	Multi-class
Accuracy	96.99%	94.78%
AUC	93.21%	96.43%

Table 5
Percentage of non-normal test splits for each dataset for Accuracy and a fixed testset.

	DT	*LDC*	*LREG*	*RFOR*
Not Tuned	97.67±6.03%	99.80± 0.57%	97.89 ±5.47%	97.82±5.47%
Tuned	98.73±4.84%	99.50 ± 1.67%	98.72 ±4.35%	100.0 ±0.00%

Our analysis indicates that data imbalance has only a marginal overall effect for the accuracy metric. However, for *AUC* metric there is around 6% increase in Gaussian distributed performance. A similar result is observed for Brier Loss, see Appendix D. This indicates that performance metrics can be affected by data imbalance.

Different models lead to different performance distributions. An example of this is shown in Fig. 2(c). To assess whether certain models are more prone to producing non-Gaussian generalization performance distributions, Table 2 reports the percentage of non-Gaussian cases observed for each model across different loss functions. Our selection covers a diverse range of model families, including ensemble methods, gradient-based learners, and models with analytical solutions. Among all models, ADAB shows the lowest propensity for Gaussian generalization performance, with approximately 1% of its accuracy distributions classified as Gaussian. This effect is slightly reduced when using the AUC metric. By contrast, the QDC model yields 30% Gaussian distributions for the AUC.

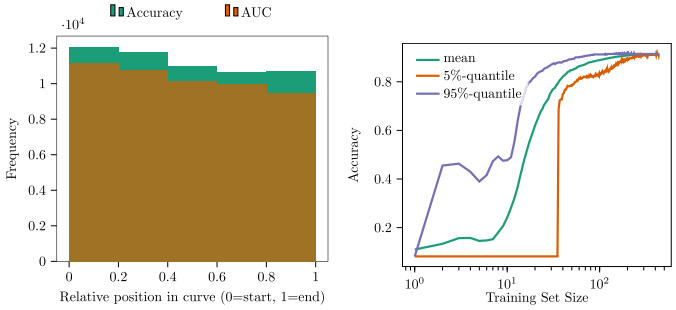
There are several modeling decisions involved in solving supervised classification problems. One key decision arises when adapting a binary classifier to handle multi-class classification tasks. In our setup, we use the one-vs-rest strategy to enable this extension. To evaluate the effect of this approach, we focus on the Logistic Regression model applied to both binary and multi-class problems, as shown in Table 4. Our results indicate that this modeling has different effects for performance metrics. The accuracy metric shows minimal sensitivity to the transition from binary to multi-class classification with a slight increase in normal distributions. In contrast, the AUC metric exhibits a decrease. This is expected since it also requires adaptation through the one-vs-rest scheme in multi-class settings.

Hyper-parameter tuning increases the non-Gaussian performance distributions. In Table 5, we examine the effects of hyperparameter tuning. Four different models are considered are, DT, LDC, LREG, and RFOR, which represent a diverse set of modeling approaches. In our database learning curves for hyper-parameter-tuned models are shorter than their untuned counterparts since their starting point is training set size $n = 10$ instead of $n = 1$. Hence, we restrict the analysis of untuned models to the same sample sizes used in the tuned setting. The evaluation metric is accuracy for consistency. For all selected models, hyper-parameter tuning results in slight increase for non-Gaussianity of the generalization performance. In the case of the RFOR model, performance is entirely non-Gaussian.

Normality of paired performance differences. Table 6 reports the proportion of non-normal performance difference distributions across all model combinations. We observe that when fixed models are evaluated on varying test sets, the paired differences between models are more likely to follow Gaussian distributions, around 20% of the cases, compared to learning curves obtained from tuned models on fixed test sets, where this holds in only about 1% of the cases. Notably, neither model tuning alone nor the choice between fixed and varying test sets shows such a pronounced increase in the normality of paired performance difference distributions.

Table 6
Percentage of non-normal paired generalization performance differences for all model combinations.

	Accuracy	AUC
Not Tuned + Varying Testset	84.85%	81.25%
Tuned + Fixed Testset	99.24%	98.40%



(a) The frequency of performance distributions that are Gaussian, as a function of the training set size.

(b) Quantile learning curves for the Quadratic Discriminant Classifier on the OpenML-11 dataset.

Fig. 3. Plots of normality frequency and quantile–mean differences along learning curves.

Smaller training set sizes exhibit less normally distributed generalization performance. Thus far, we have established that generalization performance distributions along learning curves are predominantly non-Gaussian, regardless of the assumptions made during the construction of the curves. To illustrate the spatial dependence of this phenomenon with respect to sample size, we normalize all learning curves obtained, with Additive sampling and Fixed Testset, to the range $[0, 1]$ with respect to the training set size and report the frequency of observed non-Gaussian distributions across this normalized domain. As depicted in Fig. 3(a), the general trend indicates that the likelihood of observing Gaussian generalization distributions increases with larger training sample sizes. On our preliminary inspection we found an exception for the AUC metric compared to others, where the initial segments appear to exhibit more Gaussian behavior. This, however, is largely due to the lack of variation, where the metric often remains constant at 0.5. In such cases, the Shapiro-Wilk test becomes invalid due to zero variance. Only 4.2% of the generalization performance distributions in our database exhibit this behavior, which we nevertheless regard as Gaussian.

3.2. Gaussianity assumption gone wrong

In the previous section, we demonstrated that the assumption of Gaussianity is frequently violated along learning curves. In this section, we investigate the consequences of this violation. Specifically, we analyze the learning curves of all models across all datasets for the *Additive* sampling strategy with Fixed Testset and examine whether the rankings of the top-3 models change.

Our baseline is the commonly used approach of selecting models based on the mean and standard deviation, which implicitly assumes Gaussian performance distributions. We compare this with selection based on alternative statistical measures: the 0.975-quantile, the median (0.5-quantile), and the 0.025-quantile. To ensure a consistent

Table 7
Top-3 model Probability of observing a change in the top 3 models for Accuracy and AUC Measures with Additive Sampling where the test set is separate and hyper-parameter tuning is done. Excluding the 3 datasets that had missing learning curves.

	Accuracy	AUC
$\mu + 2\sigma$ *vs 0.975-Quantile*	0.94	0.90
μ *vs Median*	0.42	0.44
$\mu - 2\sigma$ *vs 0.025-Quantile*	0.40	0.44

comparison between quantiles and the mean, we evaluate the 0.975/0.025-quantiles relative to the mean using the interval $\mu \pm 2\sigma$, where μ and σ are the mean and standard deviation of the performance distribution, respectively in Table 7. From the table, we observe that selection based on the mean often resembles selection based on lower quantiles, with the probability of a different model ordering around 0.40 for accuracy and 0.44 for AUC. Comparing the mean with the median shows a slightly higher probability of reordering, while the 0.975-quantile produces the most conflicts, with a 0.94 probability that the top-3 models differ in ranking or composition. This indicates that model ordering changes significantly across different quantiles with respect to the mean, especially when the best-performing models are of interest.

4. Discussion and conclusion

In this work, we investigate the distribution of classifier performance across multiple datasets, evaluation measures, sampling strategies, and training set sizes. We find that, in most cases, performance distributions deviate from a Gaussian distribution. In particular, smaller training sets often produce distributions with long tails and multiple modes, and hyper-parameter tuning amplifies the non-Gaussian behavior. This effect is consistent across different test splits and sampling strategies, although the choice of sampling method itself has only a minor impact on Gaussianity.

In addition to the normality tests presented in Section 3.1, skewness [16] and kurtosis [17] tests at a significance level of $\alpha = 0.05$, show that approximately 84% of the skewness values and 75% of the kurtosis values in our database deviate from those of a Gaussian distribution. Analyzing the median skewness for the final training set sizes in our learning curve database suggests that, for all metrics, generalization performances are left-skewed. In terms of kurtosis, the median values indicate that the AUC and Accuracy metrics tend to exhibit leptokurtic behavior, whereas the Brier and Cross-Entropy losses generally display platykurtic characteristics. These results are shown in Appendix B.

The results reported in [18,19] suggest that the normality assumption is well justified for many sources of variation in hyper-parameter optimization for deep learning applications, which are not covered in our study. Nevertheless, not all generalization performances in these studies conform to a Gaussian distribution also. Moreover, their experimental setup differs from ours: their sources of variation do not include training set sizes. Our spatial analysis along learning curves further reveals that smaller training sets are more likely to yield non-Gaussian distributions, whereas larger sets tend to produce distributions closer to Gaussian behavior, partially explaining the patterns observed in prior work. High-fidelity learning curve generation for deep learning models is needed to fully verify these results.

We show the deviations from Gaussianity have practical consequences for model selection. When comparing models using the mean and standard deviation versus robust statistics such as quantiles and medians, substantial discrepancies arise. For instance, ranking models by the 0.95 quantile instead of the mean changes the top three models in 90% of the cases, highlighting that relying solely on mean performance can be misleading.

Average learning curves can be used for model selection [20] faster and equally reliable as cross-validation. Under the appropriate assumptions, one can even find strong results regarding the expected minimum error achievable for a given training set size [21]. However, we argue that quantile-based learning curves can provide additional insights into training set size requirements and model robustness. For example, in Fig. 3(b), the average performance of the Quadratic Discriminant Classifier on the OpenML-11 dataset increases steadily with more training data, yet the lower quantile remains flat, indicating that poorly performing runs do not improve until the training set size reaches $n = 40$. Conversely, while the average performance appears to plateau around 100 samples, the lower quantile begins to improve, revealing delayed benefits for underperforming runs. These patterns demonstrate that averages can obscure critical information about reliability and robustness.

The failure of the Gaussianity assumption has further implications. In particular, it violates assumptions underlying (paired) t-tests commonly used for algorithms trained with random sub-sampling. Although we did not use the exact sampling strategies from [22,23], similar assumptions may be violated there as well, given that we observed no significant differences between sampling methods.

For robust empirical studies, we recommend using quantiles of generalization performance distributions using the statistical frameworks presented in [24,25] or investigating stochastic dominance as proposed in [26]. While generating well-sampled performance distributions for accurate quantile estimation is computationally demanding [27], such rigor is essential for benchmarking and model comparison to fulfill their intended scientific purpose [28,29], particularly in data-scarce or high-uncertainty settings.

CRediT authorship contribution statement

O. Taylan Turan: Writing – review & editing, Writing – original draft, Visualization, Software, Investigation, Formal analysis, Data curation, Conceptualization; **Marco Loog:** Writing – review & editing, Supervision, Investigation; **David M.J. Tax:** Writing – review & editing, Supervision, Investigation, Conceptualization.

Data availability

Data/code is publicly available and mentioned in the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Learning curve database

A.1. Dataset information and sampling strategies

As mentioned in Section 2 we have selected 10 datasets from OpenML [9]. The rationale for selecting the datasets is directly tied to the effects we aim to investigate, primarily the problem type (binary versus multi-class classification) and the degree of class imbalance. Given that our learning curve database evaluates performance across varying training set sizes, incorporates 1000 repetitions per experiment, and includes hyperparameter optimization, there is a significant computational demand. To ensure feasibility, we have selected datasets of small to medium size. In constructing our collection, we carefully ensured that all key features relevant to our study objectives are represented. The classification datasets used in this work are presented with their corresponding OpenML-ID is given in Table A.8.

Our database consists of multiple sampling and splitting strategies these are illustrated in Fig. A.4.

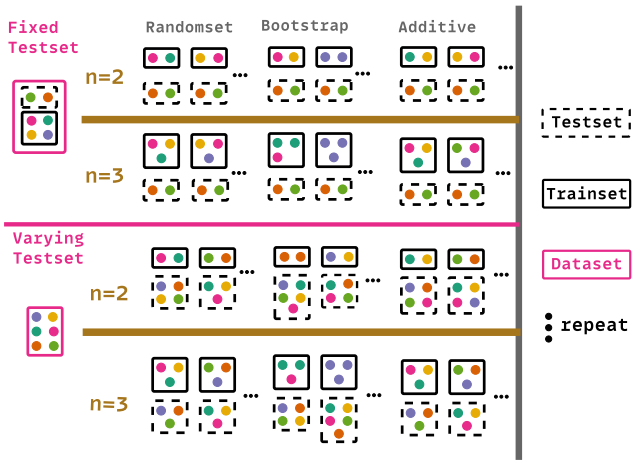


Fig. A.4. Summary of sampling strategies for learning curve generation (*Additive,Bootstrap,Random Selections and with and without split*) used in this work. The dataset for training and testing for 2 repetitions is given for sample sizes 2 and 3 for both varying and fixed test set. At the top (above the pink line) a fixed test set is used, in the bottom a varying test set is used. The dashed lines encapsulate the testing set, and the solid lines encapsulate the training set used for the learning curve creation. The colored balls represent the data points in the dataset. Each ball has a unique color, hence seeing the same colored data point multiple times indicate sampling with replacement. Ellipsis represent the repetition of the experiment with different samplings for the same sample size.

Table A.8
Dataset information used in creating the Learning Curve Database.

ID	Samples	Features	Classes	Balanced
11	628	4	3	✗
23	1473	9	3	✗
37	768	8	2	✗
53	270	13	2	✓
61	150	4	3	✓
1063	523	21	2	✗
44037	4970	12	2	✓
46597	2111	16	7	✓
46847	383	16	2	✗
46733	520	16	2	✗

	Description
11	Balance Scale Weight & Distance
23	Contraceptive Method Choice
37	Pima Indians Diabetes
53	Statlog (Heart)
61	Iris
1063	KC2 Software Defect Prediction
44037	Tabular Benchmark
46597	Estimation of Obesity Level
46847	Differentiated Thyroid Cancer Recurrence
46733	Early Stage Diabetes Risk Prediction

A.2. Missing learning curves

In our current experiments 94 learning curves are missing out of 4800 expected learning curves. These are due to timed out experiments which are not finalized because of time constraints. The missing experiments are summarized in Table A.9. From these one can see that the missing learning curves mostly come from Gaussian Kernel Support Vector Classifier for large datasets with hyper-parameter tuning.

Table A.9
Summary of missing learning curves.

Category	Group	# of Missing
Model	ADAB	2
	GSVC	64
	LREG	6
	NNC	16
	RFOR	6
Dataset ID	1063	24
	44037	20
	46597	50
Hyper-parameter	Not Tuned	18
	Tuned	76

Appendix B. Skewness and excess kurtosis

The kurtosis and skewness values at the final training set size of each learning curve in our database are presented in Table B.10. The final points were selected because they exhibit the smallest deviations from Gaussianity compared to earlier stages of the learning curves.

Table B.10
Median values of skewness and kurtosis for each metric at the last observation point.

Metric	Skewness	Excess-Kurtosis
Accuracy	-0.12	0.19
AUC	-2.40	8.23
Cross-Entropy	-0.61	-0.72
Brier	-0.99	-0.72

Appendix C. Effect of sampling strategies

For the same set of learners used in Section 3.2, we examine the effect of different sampling strategies without using a separate test split, as shown in Table C.11. The only clear difference appears for the Logistic Regression model combined with Bootstrap sampling, where we observe an increase in the proportion of performance distributions classified as Gaussian. However, beyond this case, we do not observe any clear or systematic relationship between the sampling methods (Additive, Bootstrapping, and Random Selection) and the Gaussianity of the resulting performance distributions, indicating that not all models are affected by the sampling strategy in the same way.

Table C.11
Average Percentage of non-normal results for selected models for investigating the effect of sampling type.

	Additive	Bootstrap	Random
DT	97.32 ± 7.17%	99.49 ± 2.00%	97.82 ± 5.78%
LDC	99.63 ± 1.53%	99.87 ± 0.27%	99.45 ± 1.52%
LREG	98.48 ± 4.50%	97.46 ± 6.36%	99.01 ± 3.51%
RFOR	99.06 ± 3.90%	98.87 ± 4.20%	98.99 ± 3.90%

Appendix D. Corresponding results for brier and cross-entropy losses results

This section presents the results corresponding to those reported in Section 3, focusing on Brier Loss and Cross-Entropy Loss. Table D.12 summarizes the impact of class imbalance and the number of classes on the Gaussianity of performance distributions. Table D.13 reports the Gaussianity across different datasets and models.

Fig. D.5 illustrates which portions of the learning curves exhibit a higher prevalence of non-Gaussian performance distributions for each loss function. These analyses provide insight into how dataset characteristics and model selection influence the normality of the observed learning curve distributions.

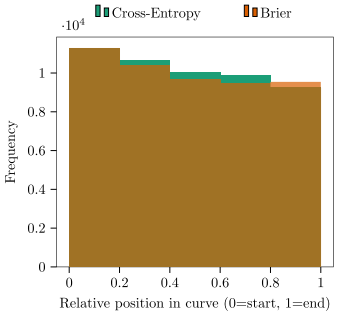


Fig. D.5. Quantifying which portion of the learning curves we see more non-Gaussian performance distributions for Brier and Cross-Entropy losses.

Table D.12
Effect of dataset imbalance and number of classes on Gaussianity for Brier and Cross-Entropy losses (transposed).

Condition	Brier	Cross-Entropy
Imbalanced	95.21%	96.97%
Balanced	90.84%	94.43%
Binary	92.60%	99.74%
Multi-class	95.22%	99.98%

Table D.13
Average Percentage of non-normal test splits for each loss measure across datasets and models.

Dataset/Model	Brier	Cross-Entropy
11	97.72±7.13%	98.99±4.17%
23	98.82±3.43%	99.46±3.28%
37	91.42±23.83%	93.32±22.00%
53	94.40±14.65%	95.39±14.16%
61	95.74±16.19%	95.13±18.01%
1063	90.51±15.15%	96.01±10.07%
44037	82.79±33.02%	91.86±17.80%
46597	91.79±13.63%	95.32±11.37%
46733	99.11±3.87%	99.13±3.42%
46847	93.13±15.69%	94.93±14.72%
ADAB	99.29±2.95%	99.80±1.51%
DT	93.89±13.76%	95.31±12.09%
GSVC	96.42±16.39%	94.97±18.92%
LDC	93.39±11.63%	94.51±10.73%
LREG	93.14±12.24%	99.49±2.90%
LSVC	91.59±27.71%	99.99±0.07%
NMC	86.60±28.35%	87.17±27.73%
NNC	95.07±11.91%	94.95±12.12%
QDC	83.79±24.31%	86.58±23.15%
RFOR	92.49±15.12%	98.33±4.34%

References

[1] C. Fröhlich, R.C. Williamson, Tailoring to the Tails: Risk Measures for Fine-Grained Tail Sensitivity, 2023, arXiv:2208.03066

[2] Z. Xiao, H. Guo, M.S. Lam, Quantile regression and value at risk, in: C.-F. Lee, J.C. Lee (Eds.), Handbook of Financial Econometrics and Statistics, Springer New York, New York, NY, 2015, pp. 1143–1167. https://doi.org/10.1007/978-1-4614-7750-1_41

[3] O. Watson, I. Cortes-Ciriano, A. Taylor, J. Watson, A decision-Theoretic approach to the evaluation of machine learning algorithms in computational drug discovery, Bioinformatics (Oxford, England) 35 (22) (2019) 4656–4663.

[4] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, Cambridge, Cambridge, 2011. https://doi.org/10.1017/CBO9780511921803

[5] J. Langford, Tutorial on practical prediction theory for classification, J. Mach. Learn. Res. 6 (10) (2005) 273–306. http://jmlr.org/papers/v6/langford05a.html.

[6] F. Mohr, T.J. Viering, M. Loog, J.N. Van Rijn, et al., LCDB 1.0: An extensive learning curves database for classification tasks, in: M.-R. Amini, S. Canu, A. Fischer, T. Guns, P. Kralj Novak, G. Tsoumakas (Eds.), Machine Learning and Knowledge Discovery in Databases, 13717, Springer Nature Switzerland, Cham, 2023, pp. 3–19. https://doi.org/10.1007/978-3-031-26419-1_1

[7] C. Yan, F. Mohr, T. Viering, LCDB 1.1: A Database Illustrating Learning Curves Are More Ill-Behaved Than Previously Thought, 2025, arXiv:2505.15657

[8] T. Viering, M. Loog, The Shape of Learning Curves: A Review, 2022, arXiv:2103.10948

[9] B. Bischl, G. Casalicchio, T. Das, M. Feurer, S. Fischer, P. Gijsbers, S. Mukherjee, A.C. Müller, L. Németh, L. Oala, L. Purucker, S. Ravi, J.N. rijn van Rijn, P. Singh, J. Vanschoren, J. velde van der Velde, M. Wever, OpenML: insights from 10 years and more than a thousand papers, Patterns 6 (7) (2025) 101317. https://doi.org/10.1016/j.patter.2025.101317

[10] C.M. Bishop, Pattern Recognition and Machine Learning, Information Science and Statistics, Springer, New York, New York, 2006.

[11] M. Budka, B. Gabrys, Correntropy-Based density-Preserving data sampling as an alternative to standard cross-Validation, in: The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE, Barcelona, Spain, 2010, pp. 1–8. https://doi.org/10.1109/IJCNN.2010.5596717

[12] O.T. Turan, Learning Curve Plus Plus, 2024, (https://github.com/taylanot/lcpp).

[13] S.S. Shapiro, M.B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (1965) 591–611. https://api.semanticscholar.org/CorpusID:124868013.

[14] J.P. Royston, Algorithm AS 181: The W Test for Normality, Appl. Stat. 31 (2) (1982) 176. arXiv:10.2307/2347986 https://doi.org/10.2307/2347986

[15] R. D'Agostino, E.S. Pearson, Tests for departure from normality, Biometrika 60 (3) (1973) 613–622. http://www.jstor.org/stable/2335012.

[16] R.B. D'Agostino, A. Belanger, A suggestion for using powerful and informative tests of normality, Am. Stat. 44 (4) (1990) 316–321. http://www.jstor.org/stable/2684359.

[17] F.J. Anscombe, W.J. Glynn, Distribution of the kurtosis statistic b2 for normal samples, Biometrika 70 (1) (1983) 227–234. http://www.jstor.org/stable/2335960.

[18] X. Bouthillier, P. Delaunay, M. Bronzi, A. Trofimov, B. Nichyporuk, J. Szeto, N. Mohammadi Sepahvand, E. Raff, K. Madan, V. Voleti, S. Ebrahimi Kahou, V. Michalski, T. Arbel, C. Pal, G. Varoquaux, P. Vincent, Accounting for variance in machine learning benchmarks, in: A. Smola, A. Dimakis, I. Stoica (Eds.), Proceedings of Machine Learning and Systems, 3, 2021, pp. 747–769. https://proceedings.mlsys.org/paper_files/paper/2021/file/0184b0cd3cfb185989f858a1d9f5c1eb-Paper.pdf.

[19] C. Lehmann, Y. Paromau, Quantifying Uncertainty and Variability in Machine Learning: Confidence Intervals for Quantiles in Performance Metric Distributions, 2025, https://doi.org/10.48550/arXiv.2501.16931

[20] F. Mohr, J.N. van Rijn, Fast and Informative Model Selection using Learning Curve Cross-Validation, CoRR abs/2111.13914 (2021). arXiv:2111.13914

[21] A. Salazar, L. Vergara, E. Vidal, et al., A proxy learning curve for the bayes classifier, Pattern Recognit. 136 (2023) 109240. https://doi.org/10.1016/j.patcog.2022.109240

[22] T.G. Dietterich, Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms, Neural. Comput. 10 (7) (1998) 1895–1923. https://doi.org/10.1162/089976698300017197

[23] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[24] X. Li, L. Tian, J. Wang, J.R. Muindi, et al., Comparison of quantiles for several normal populations, Comput. Statist. Data Analy. 56 (6) (2012) 2129–2138. https://doi.org/10.1016/j.csda.2012.01.002

[25] R.R. Wilcox, D.M. Erceg-Hurn, F. Clark, M. Carlson, et al., Comparing two independent groups via the lower and upper quantiles, J. Stat. Comput. Simul. 84 (7) (2014) 1543–1551. https://doi.org/10.1080/00949655.2012.754026

[26] R. Dror, S. Shlomov, R. Reichart, Deep dominance - How to properly compare deep neural models, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57Th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2773–2785. https://aclanthology.org/P19-1266/. https://doi.org/10.18653/v1/P19-1266

[27] E.N. Jonsson, J. Nyberg, A quantitative approach to the choice of number of samples for percentile estimation in bootstrap and visual predictive check analyses, CPT: Pharmacomet. Syst. Pharmacol. 11 (6) (2022) 673. https://doi.org/10.1002/psp4.12790

[28] M. Dehghani, Y. Tay, A.A. Gritsenko, Z. Zhao, N. Houlsby, F. Diaz, D. Metzler, O. Vinyals, The Benchmark Lottery, 2021, arXiv:2107.07002
[29] O.E. Gundersen, S. Shamsaliei, H.S. Kjærnli, H. Langseth, et al., On reporting robust and trustworthy conclusions from model comparison studies involving neural networks and randomness, in: Proceedings of the 2023 ACM Conference on Reproducibility and Replicability, ACM REP '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 37–61. https://doi.org/10.1145/3589806.3600044