The Impact of Imbalanced Training Data on Learning Curve Prior-Fitted Networks

Bozhidar Kostov[1]

Supervisor(s):Tom Viering[1]**, Cheng Yan**[1]**, Sayak Mukherjee**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Bozhidar Kostov
Final project course: CSE3000 Research Project
Thesis committee: Tom Viering, Cheng Yan, Sayak Mukherjee, Matthijs Spaan

Abstract

Learning curves represent the relationship between the amount of training data and the error rate in machine learning. An important use case for learning curves is extrapolating them in order to predict how much data is needed to achieve a certain performance. One way to do such extrapolations is using Deep Learning with a Prior-Fitted Network(PFN). This paper explores how training the PFN on an imbalanced dataset, i.e. containing learning curves from two or more machine learning models with a skewed distribution, affects the performance of the network. Research into imbalanced learning has shown that machine learning models can favor the more prevalent classes or data. Therefore, it is worthwhile to explore whether such trends can occur for the neural networks that we train for learning curve extrapolation. Our experiments focused on analyzing different imbalance scenarios and comparing them. Our results show that mixing learning curves from different learners can improve extrapolation performance in some cases, but the effect strongly depends on the learner characteristics and training proportions.

1 Introduction

When training a machine learning model we might expect a certain performance - this is usually dependent on the amount of data we use during training. However, collecting data can be difficult, expensive and/or time-consuming. Therefore, it can be beneficial to know the relationship between performance and the amount of training data. The plot of this relation is called a learning curve. More specifically, it can be useful if we have the learning curve until some point to be able to extrapolate and predict how it would look like if more data is used. This means that we can predict how much data is needed for a machine learning model to achieve a particular performance.

Traditionally the extrapolation of learning curves has been done using parametric formulas [6]. It is expected that more training data leads to better performance. However, learning curves can be "ill-behaved", i.e. the performance of the model does not always improve with more data. Learning Curve Prior-Fitted Networks (LC-PFNs) [7] are used for learning curve extrapolation as well. They are trained directly on a dataset containing learning curves. The learning curves the LC-PFN is trained on depend on both the learner and dataset from which the curves are derived from. Therefore, this leads to three testing scenarios: Unseen Data - we evaluate the model on learning curves that come from the same learner but use different datasets, Unseen Learner - the learning curves used for evaluation use the same data but on a different learner, Unseen Data Unseen Learner - a combination of the two other testing scenarios. Our research focused on the evaluation of the performance of the LC-PFN when the training data used are sampled from two different learners and it is imbalanced, i.e. the curves from one learner are more represented than the curves from the other. For testing purposes we focused only on the Unseen Data scenario because the other two introduce a domain shift [8] , which also has an effect on performance. This paper will address the following questions:

RQ1: How does training LC-PFNs on imbalanced datasets compare to training them on data from a single learner in terms of extrapolation performance?

RQ2: What trends emerge in LC-PFN performance as the proportion of training data from each learner changes in mixed training sets?

The outline of the report is as follows. Section 2 discusses related work to our research - formal definition of learning curves, LCDB, LC-PFNs and imbalanced regression. Section 3 presents the experimental setup - what models are trained, how their performance is evaluated and the metrics used. In Section 4 a summary of the results of our experiments can be found and their implication. Moreover, the ethical aspects and reproducibility of our work are presented in Section 5. Finally, Section 6 outlines the future work that can be done on the project and gives a conclusion in which key concepts and takeaways are summarized.

2 Related Work

This paper extends recent research in learning curve extrapolation by building upon the Learning Curve Prior-Fitted Networks framework [7]. We provide background on learning curves in machine learning and review the LC-PFN approach to curve extrapolation, which forms the foundation for our research. Moreover, this section discusses the Learning Curve Database (LCDB) [4; 9] and its uses. Finally, imbalanced regression is also presented as a concept.

2.1 Learning Curves

Learning curves are fundamental tools in machine learning that visualize the relationship between model performance and training set size. There are two types of learning curves: Epoch-wise learning curves, that plot the performance over multiple training runs over the same data, and Sample-size learning curves. Sample-size curves plot the performance of the model as a function of the size of the training set. Both types of curve are used in machine learning research and practice. Epoch-wise curves are useful for hyperparameter tuning and algorithm selection [6]. Sample-size curves are more beneficial for project planning or resource allocation because they give the relationship between amount of data and performance of model. On Figure 1 an example of a learning curve is shown. It can be seen that the error of the machine learning model decreases as it is trained on more data, [6] defines such curves as "well-behaved".

However, [6] shows that not all learning curves follow such a pattern. These curves are defined as "ill-behaved". An example of a "ill-behaved" curve is shown on Figure 2. Certain machine learning models are more prone to have ill-behaving learning curves such as Quadratic Discriminant Analysis(QDA). Research shows that "ill-behaved" curves are more common than expected [9]. Our research focuses on Sample-size learning curves, more specifically doing

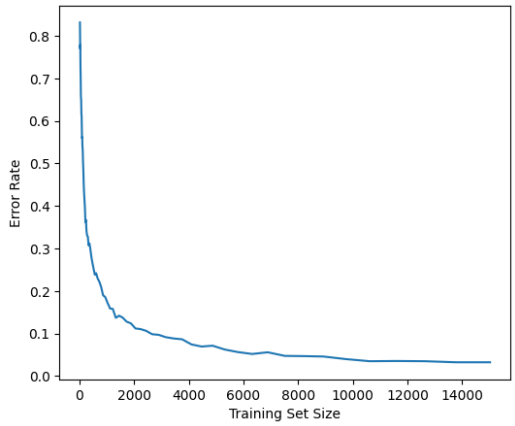

Figure 1: Example of a learning curve -

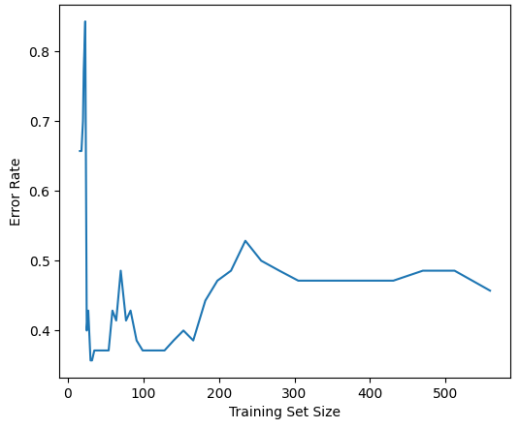

Figure 2: Example of a ill-behaved learning curve

extrapolation using a neural network.

2.2 Learning-Curve Prior Fitted Network

The first usage of prior-data fitted neural networks (PFNs) for epoch-wise learning curve extrapolation can be found in [1]. Learning Curve Prior-Fitted Networks(LC-PFNS) represent a bayesian approach to learning curve extrapolation that uses transformer architecture. Moreover, [7] extends the LC-PFN approach to also work for sample-wise learning curves. The PFN developed can be trained on two types of data driven priors. The first uses parametric curve fitting to generate synthetic data and the second trains directly on learning curves. Both approaches use the Learning Curve Database (LCDB) for training and evaluation. This approach enables LC-PFNs to provide not just a single curve as a prediction but also a confidence interval for the shape of the learning curve which can be beneficial especially in cases of ill-behavior. Figure 3 shows an example prediction of the LC-PFN: In this paper we used the sample-wise LC-PFN trained directly on data from LCDB.

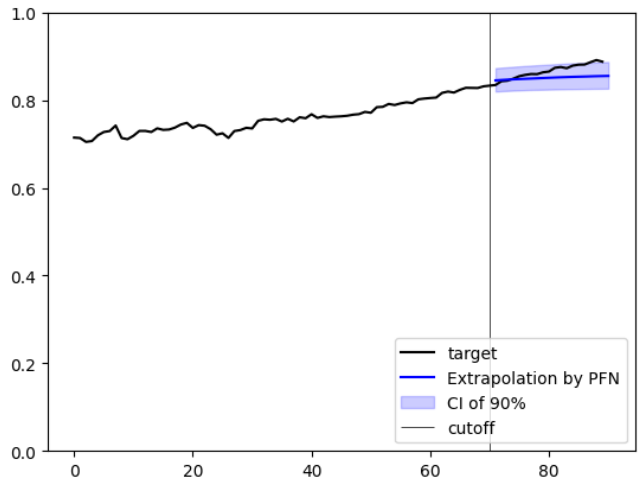

Figure 3: Prediction of the LC-PFN

2.3 Learning Curve Database

The Learning Curves Database (LCDB) 1.0 is an extensive collection of learning curves that provides empirical data for 20 classification algorithms evaluated on 246 OpenML datasets [4]. Unlike previous studies limited to small numbers of datasets and algorithms, LCDB offers over 150 GB of ground truth and probabilistic predictions, enabling comprehensive analysis of learning curve behavior. Initial analysis from LCDB demonstrates that sample-wise learning curves are predominantly monotonic and convex, with peaking being relatively rare. The research also reveals systematic patterns in learning curve crossing behavior, where algorithms may start poorly but eventually outperform other learners with enough training data. However, [9] developed an extension to this database called LCDB 1.1 which contains significantly more data and fixes issues with the database such as adding feature scaling. Research into LCDB 1.1 reveals that ”ill-behaved” learning curves are more frequent than previously thought. Furthermore, some learners are more prone to being ”ill-behaved” than others. On table 1 the results of this research into different learners is shown. We used this table in order to select the learners for our experiment. The learners we used for training the LC-PFNs used in our experiments were selected based on the results shown in the paper. Table 1 shows which learners we used and the percentage of ill-behaved learning curves that are in the dataset for that learner. Based on this we can label Extra Tree, Extra Trees and Perceptron as ”well-behaved learners” and QDA and SVC Sigmoid as ”ill-behaved learners”. We will use these groupings later in the paper.

2.4 Imbalanced Regression

Imbalanced learning traditionally refers to scenarios where certain classes or data distributions are underrepresented which can lead to models that are biased towards the more dominant classes or distributions. While most commonly researched in classification, the concept extends to regression settings - the continuous target values have an imbalanced

Table 1: Percentage of Ill-Behaved Learning Curves per Learner based on results from LCDB 1.1

Learner	% Ill-Behaved Curves
Extra Trees	3.4%
Extra Tree	1.9%
Perceptron	3.8%
SVC Sigmoid	58.1%
QDA	45.7%

distribution. As discussed in this survey of imbalanced learning [2] , such a distribution skew in regression tasks can lead to performance degradation, particularly in the underrepresented regions. When dealing with imbalanced training data it can be beneficial to evaluate the performance of a machine learning model separately for each class or data distribution. This is because poor performance on one class or distribution can be masked by a strong performance on another when results are combined in a mixed test set. The curve prediction done by the LC-PFN is a type of regression, however, in this case the target values are curves. Nevertheless, the idea of an imbalanced target value distribution is still applicable. Therefore, our experiments explore the effect of imbalance by training LC-PFNs on imbalanced splits of learning curves from different learners and analyzing the impact on extrapolation performance.

3 Experimental Setup

This section describes the experiment that has been performed in order to answer our research question. The learning curves used for training and evaluation were sourced from LCDB 1.1, and the model was based on the LC-PFN neural network architecture.

We choose two learners whose learning curves we will use for training. Since it is infeasible to experiment with every possible combination of learners, we used the results from [9] in order to select learners with "well-behaved" learning curves and learners with "ill-behaved" ones. There are 3 possible scenarios for the experiment: "well-behaved" mixed with "well-behaved", "well-behaved" with "ill-behaved" and "ill-behaved" with "ill-behaved".

For each combination of learners A and B we have the same training splits: (80% A, 20% B), (60% A, 40% B), (40% A, 60% B) and (20% A, 80% B). The amount of curves that are used for training remains the same - 5300, but the proportion of curves from each learner is varied. We train 3 PFNs for each split with different random seeds.

We also train 3 PFNs with on different seeds with curves only from one of the learners and 3 other PFNs with curves from the other learner. We use these networks as a baseline to compare the mixed training networks with. We use three different seeds to account for the variability introduced during training - which curves we train on and also there is an element of randomness during the training of the PFN.

We then evaluate each training split for learners A and B on the Unseen Data curves for both learners separately. They are the same for each seed and training split which ensures that the comparisons are fair. The metrics used are MAE, Miscoverage and Area. Their formal definitions are provided later in the report.

Finally, we compare every pairwise combination for training splits, i.e. 80%/20% versus 60%/40%, using the Wilcoxon Signed-Rank test.

For our performance metric we picked the following metrics:
MAE $= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$ - Mean Absolute Error between the ground truth, i.e. the curve after the cutoff, and the prediction mean of the PFN.

Miscoverage $= (\hat{y}_{\text{true}} < \hat{y}_{\text{lower}}) \vee (\hat{y}_{\text{true}} > \hat{y}_{\text{upper}})$: the percentage of the curve that is *outside* the 90% Confidence Interval.

Area $= \sum_{i=1}^{N}\left(\hat{y}_{\text{upper}}^{(i)} - \hat{y}_{\text{lower}}^{(i)}\right)$: the total area covered by the confidence interval for the curve. This metric is used together with Miscoverage, as a reduction in Miscoverage can sometimes be achieved by increasing the confidence interval. A larger area thus can indicate increased uncertainty about the shape of the learning curve.

All of the metrics are defined such that lower values indicate better performance. This is why Miscoverage is used instead of Coverage - to maintain consistency in interpretation. In order to check the statistical significance of the results of our experiments we used the Wilcoxon Signed-Rank Test [5]. It is a non-parametric statistical test used to check if two related samples come from the same distribution. It checks if the population mean ranks of the samples differ,i.e. whether there is a consistent difference between paired observations. It serves as a non-parametric alternative to the paired t-test and does not assume normality of the data. Moreover, it is possible to do this test with a two-tailed or one-tailed hypothesis. A two-tailed test is appropriate when we are interested in detecting any difference between the two paired samples, regardless of direction. A one-tailed test is used when we have a specific directional expectation. For our experiment we used a one-tailed test because we are interested in learning whether the performance metrics are increasing or decreasing. Wilcoxon Signed-Rank test is the appropriate test for these metrics because they are not normally distributed thus a t-test is not applicable. When doing multiple comparisons the chance of a Type 1 error (false positive) increases. Therefore, we also use the Bonferroni correction method [3].

4 Results and Discussion

For each scenario for the experiment we get six plots - comparing the MAE, Miscoverage, and Area values when evaluating on the Unseen Data for each learner. On Figures 4, 5, 6 we can see the boxplots comparing training splits for PFNs trained on Extra Trees and SVC Sigmoid and evaluated on Extra Trees. On these figures we have only plotted the results for one seed to give an idea of how the results look like. All of the results can be found in the Appendix A. Moreover, due to the training setup for the LC-PFN - the results for each seed are statistically independent. Therefore, it is better to analyze each seed separately and look for results that hold across all of them. Furthermore, as can be seen on Figure 4 the distribution of results for MAE is skewed, which is shown by the high number of outliers in the box plot. This is why the Wilcoxon signed-rank test was used as the primary method for comparing the different training splits. Moreover, Table 2

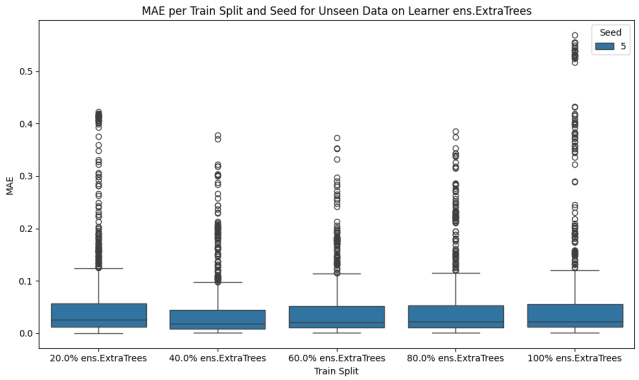

Figure 4: MAE results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for Extra Trees

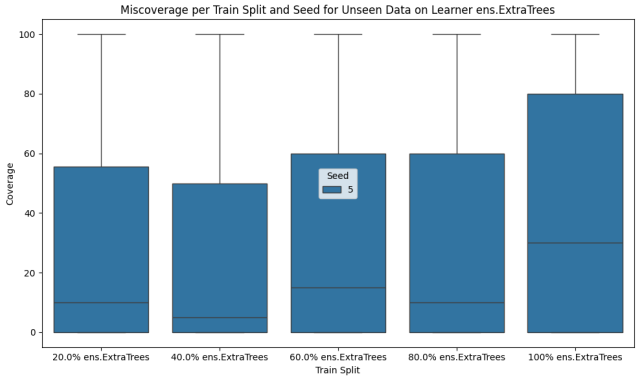

Figure 5: Miscoverage results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for Extra Trees

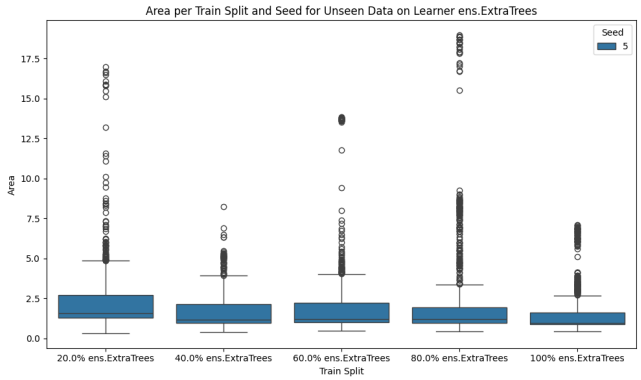

Figure 6: Area results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for Extra Trees

Comparison	Better performing split
100% vs 80%	100% Perceptron
100% vs 60%	100% Perceptron
100% vs 40%	100% Perceptron
100% vs 20%	100% Perceptron
80% vs 60%	60% Perceptron
80% vs 40%	40% Perceptron
80% vs 20%	20% Perceptron
60% vs 40%	40% Perceptron
60% vs 20%	20% Perceptron
40% vs 20%	20% Perceptron

Table 2: Results of multiple Wilcoxon signed-rank tests with correction for PFNs trained on a mixed split of Extra Tree and Perceptron for MAE on Unseen Data for Perceptron for seed 23.

shows the result of applying multiple Wilcoxon signed-rank tests with correction when comparing PFNs trained on different splits of Extra Tree and Perceptron on seed 23. The comparison column shows which training splits we are comparing - 100% refers to the PFN trained only on data from the learner which we are doing the Unseen Data evaluation on. The second column indicates which training split performs better. If the p-value of one of the two one-sided tests is below 0.05, the result is considered statistically significant. If neither test yields a significant result, the outcome is marked as "neither."

Well-behaved and ill-behaved:

For this case we picked Extra Trees as the well-behaved learner and SVC Sigmoid as the ill-behaved one. From Table 1 we can see that SVC Sigmoid has 58.1% ill-behaved curves compared to 3.4% for Extra Trees. When comparing the different training splits in the case of Unseen Data for **Extra Trees** we have the following results:

- **MAE:** Comparing the results of the mixed training sets(80%/20%, 60%/40% etc.) to the results from the PFN trained only on curves from Extra Trees does not give very consistent results. Across all three seeds the 60%/40% split always has lower MAE than the PFN trained on the Extra Trees learner. Moreover, when comparing the the mixed training splits between themselves, the Wilcoxon test reveals that using 20% of Extra Trees is consistently the worst performing PFN. The other comparisons are inconsistent across seeds.
- **Miscoverage:** The overall trend for this metric is that as the amount of curves from Extra Trees we use for training is reduced so does the Miscoverage rate.
- **Area:** This metric follows a similar trend to miscoverage. In this case as we reduce the training data from Extra Trees the area of the Confidence Interval increases. This can be used to explain why the miscoverage is improving - as the confidence interval grows it is more likely to contain the curve.

In the case of comparing the results from evaluating the Unseen Data for **SVC Sigmoid** we have the following results:

- **MAE:** When we compare the PFN trained exclusively on curves from the SVC learner to the mixed training models, we can see that the mixed models with a low amount of curves from Extra Trees (20%) consistently outperforms the former. When analyzing pairwise mixed splits, MAE tends to improve as the proportion of

Extra Trees curves in the training data decreases. However, this improvement plateaus between the 40% and 20% splits

- **Miscoverage:** The mixed training models outperform the PFN trained on only SVC Sigmoid consistently. The pairwise comparisons between the training splits do not give consistent results across splits apart from when we compare the 80%/20% and 60%/40% splits and 80% versus 40%. In these cases the PFNs that are trained on less data from Extra Trees are better.
- **Area:** The area of the confidence interval is lower when we train on only curves from SVC Sigmoid compared to using a mixed training set in all cases. However, doing a pairwise comparison between training splits does not show any trend - for example for Seed 10 a 20%/80% split is worse than a 40%/60% split, but for Seed 23 it is the opposite.

Well-behaved and Well-behaved:

For this case we picked Extra Tree as the first well-behaved learner and Perceptron as the second well-behaved one. From Table 1 we can see that they have 1.9% and 3.8% ill-behaved curves respectively. When comparing the different training splits in the case of Unseen Data for **Extra Tree** we have the following results:

- **MAE:** Using a mixed training split with 20% or 40% curves from Extra Tree gives a better performance compared to using a PFN trained only on the Extra Tree learner. For the pairwise comparisons, the 80% split is consistently the worst performing one for MAE. However, the rest do not follow a clear trend and depend on the seed.
- **Miscoverage:** The trend for this metric is that it improves as the amount of curves from Extra Tree is reduced until they are 40-20% of the training data. This is true for both pairwise comparisons and when comparing to PFN trained only on the Extra Tree learner.
- **Area:** For this metric as we reduce the training data from Extra Tree the area of the Confidence Interval increases. This can be used to explain why the miscoverage is improving - as the confidence interval grows it is more likely to contain the curve.

When comparing the different training splits in the case of Unseen Data for **Perceptron** we have the following results:

- **MAE:** When comparing the PFN trained only on curves from Perceptron and the mixed training splits there is not a general trend. However, for the pairwise comparisons between mixed splits the less data is used from Extra Tree the better the MAE.
- **Miscoverage:** The PFNs trained on only data from Perceptron perform best overall here. Analyzing the pairwise comparisons shows that using more curves from the Perceptron learner results in better miscoverage.
- **Area:** The PFNs trained purely on Perceptron show the worst performance in terms of area. Pairwise comparisons reveal that as the proportion of Extra Tree training curves is reduced the area metric consistently improves.

Ill-behaved and ill-behaved:

For this case we picked QDA as the first ill-behaved learner and SVC Sigmoid as the second ill-behaved one. From Table 1 we can see that they have 45.7% and 58.1% ill-behaved curves respectively. Comparing the results of the Wilcoxon tests for each seed revealed no trend for any metric with one exception - the area of the confidence interval is consistently lower when comparing the PFN trained only on curves from SVC Sigmoid compared to the mixed training splits.

Discussion

Based on the comparative analysis, the effectiveness of mixed training splits depends significantly on the characteristics of the learners that are used: Mixing well-behaved and ill-behaved learners consistently improves performance when evaluating on the ill-behaved learner (SVC Sigmoid). Including even 20% well-behaved (Extra Trees) curves reduces MAE and miscoverage for SVC Sigmoid evaluation, suggesting well-behaved data acts as a stabilizer. Moreover, there is evidence that mixing well-behaved and ill-behaved training curves can also improve MAE performance. This suggests that training on a mixed learner set could be inherently better for MAE performance. Although reducing Extra Trees data improves miscoverage due to wider confidence intervals, this presents a trade-off between coverage and certainty which is not always an improvement.

Mixing two well-behaved learners (Extra Trees & Perceptron) shows asymmetric benefits. Extra Trees performance improves (lower MAE & miscoverage) when augmented with Perceptron data (particularly at 20-40% splits). However, evaluating on Unseen Data for Perceptron shows best results when trained only on Perceptron learning curves for miscoverage, suggesting its stability might be diluted by external data despite slight MAE gains from very low Extra Trees inclusion in some cases.

Mixing two ill-behaved learners (SVC Sigmoid & QDA) yields no consistent performance trends for MAE or miscoverage across metrics or seeds. The only clear effect is that mixed splits produce larger confidence intervals than training solely on SVC Sigmoid, mirroring the area/miscoverage trade-off observed elsewhere but without clear performance advantages.

In general, well-behaved learners improve the reliability of ill-behaved learners when included in training. However, the optimal mixing ratio depends on the learner's themselves and the metric prioritized,e.g. MAE vs. coverage. Ill-behaved learners offer little reciprocal benefit when mixed, and combining them can give unreliable results.

5 Responsible Research

This section examines the ethical implications of our research and discusses the reproducibility of our methods.

5.1 Reproducibility

All experiments were done using fixed random seeds (5, 10, 23) which allows for our research to be reproducible. The experimental setup including the specific training splits and evaluation metrics (MAE, Miscoverage, Area) can be found

in this paper in Section 3. Moreover, our code is publicly available at https://github.com/Bozhidar1/ResearchProject.

5.2 Data Ethics

This research used publicly available datasets from the Learning Curve Database (LCDB) 1.1 [9] , which contains learning curves derived from established OpenML datasets. The use of publicly accessible data ensures transparency and eliminates concerns regarding sensitive information.

5.3 Usage of LLM

We used generative AI tools such as LLM as support when writing this report in LaTeX. They were not used to generate code, ideas, or to analyze results. Some examples of prompts used are:
"¡data¿ Can you write this data as a table in latex."
"¡paragraph¿ Rewrite this paragraph so it sounds better."
"I want to say that ¡idea¿, can you help me write it as a sentence.".

6 Conclusions and Future Work

This section first lays out the conclusions we have made from our research. Then it gives suggestions for related future work.

6.1 Conclusion

This research investigated how training Learning Curve Prior-Fitted Networks (LC-PFNs) on imbalanced datasets affects their performance in learning curve extrapolation. We addressed two primary research questions: (RQ1) how imbalanced training compares to single-learner training, and (RQ2) what trends emerge as the proportion of training data from different learners changes.
Our experimental analysis reveals that the effectiveness of mixed training is highly dependent on the underlying characteristics of the learners being combined. Key findings include that well-behaved learners can improve ill-behaved ones when included in training data, but this benefit is not reciprocal. Mixed training consistently improved performance when evaluating on ill-behaved learners, with even small proportions (20%) of well-behaved data providing significant benefits. However, combining two well-behaved learners showed asymmetric effects, and mixing two ill-behaved learners yielded no consistent improvements.
These results suggest that the composition of training data matters significantly for LC-PFN performance, and that strategic mixing of learner types can be beneficial under specific conditions. The optimal mixing strategy depends on both the learner characteristics and the performance metrics prioritized.

6.2 Future Work

Improve LC-PFN training:
The current parameters we used for training the LC-PFN are not the best performing ones found in [1]. Moreover, during training, we only used curves with length 80 or less. Therefore, it can be worthwhile to repeat our experiments with a different training setup. One significant limitation of our current work is the sensitivity of the neural network's performance to the random seed used during training. Addressing this issue would allow for more reliable and conclusive experimental findings.
Look into Sampling-Based Strategies for addressing Imbalanced Regression:
There are methods to address the performance degradation that can be introduced by imbalanced learning. It has been suggested that imbalanced regression can be addressed through various data sampling strategies [2]. The paper highlights SMOGN (Synthetic Minority Over-sampling Technique for Regression with Gaussian Noise) as a key method, which enhances the representation of rare target values by generating synthetic samples and adding noise to diversify the training data. Additionally, WERCS (Weighted Relevance-based Combination Strategy) is proposed to adjust oversampling based on the relevance of each sample. Such methods can be explored in order to improve the performance of the LC-PFN.

References

[1] Steven Adriaensen, Herilalaina Rakotoarison, Samuel Müller, and Frank Hutter. Efficient bayesian learning curve extrapolation using prior-data fitted networks, 2023.

[2] Wuxing Chen, Kaixiang Yang, Zhiwen Yu, Yifan Shi, and C. Chen. A survey on imbalanced learning: latest research, applications and future directions. *Artificial Intelligence Review*, 57, 05 2024.

[3] Winston Haynes. *Bonferroni Correction*, pages 154–154. Springer New York, New York, NY, 2013.

[4] Felix Mohr, Tom J. Viering, Marco Loog, and Jan N. van Rijn. Lcdb 1.0: An extensive learning curves database for classification tasks. In Massih-Reza Amini, Stéphane Canu, Asja Fischer, Tias Guns, Petra Kralj Novak, and Grigorios Tsoumakas, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 3–19, Cham, 2023. Springer Nature Switzerland.

[5] Denise Rey and Markus Neuhäuser. *Wilcoxon-Signed-Rank Test*, pages 1658–1659. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[6] Tom J. Viering and Marco Loog. The shape of learning curves: a review. *CoRR*, abs/2103.10948, 2021.

[7] Tom Julian Viering, Steven Adriaensen, Herilalaina Rakotoarison, and Frank Hutter. From epoch to sample size: Developing new data-driven priors for learning curve prior-fitted networks. In *Proceedings of the AutoML Conference 2024 (Workshop Track)*, 2024.

[8] Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. Exploring domain shift in extractive text summarization. *CoRR*, abs/1908.11664, 2019.

[9] Cheng Yan, Felix Mohr, and Tom Viering. Lcdb 1.1: A database illustrating learning curves are more ill-behaved than previously thought, 2025.

A Results of Unseen Data evaluations

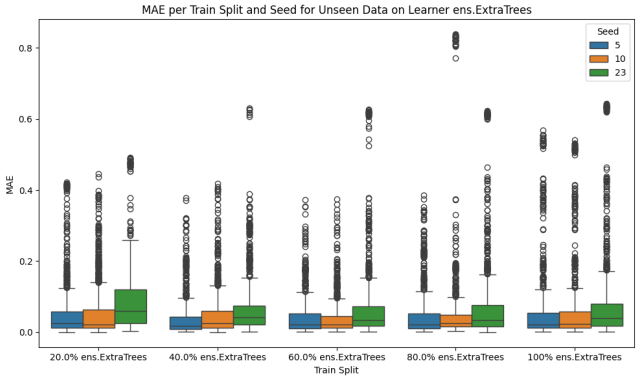

Figure 7: MAE results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for Extra Trees

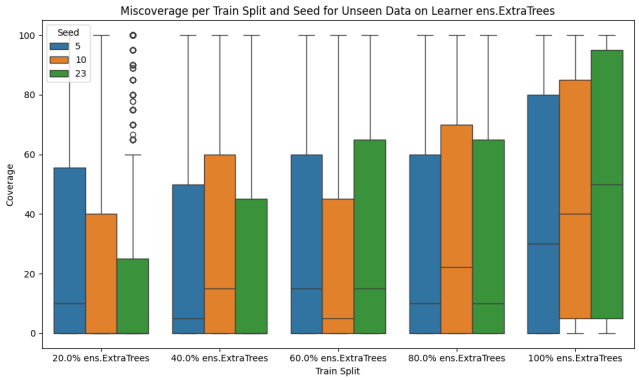

Figure 8: Miscoverage results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for Extra Trees

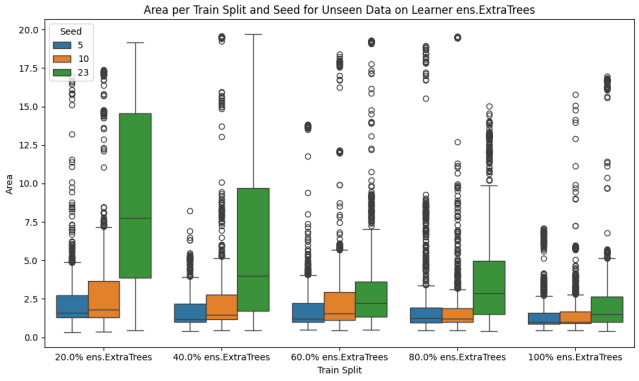

Figure 9: Area results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for Extra Trees

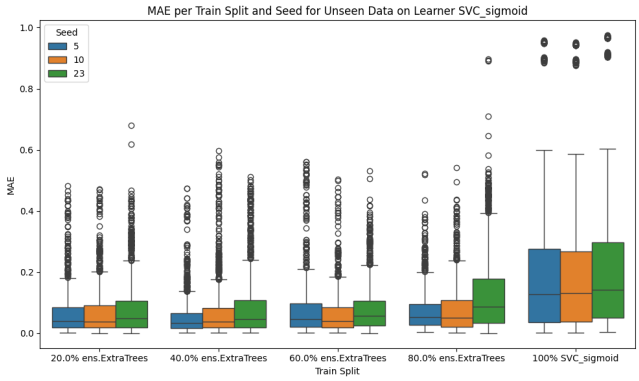

Figure 10: MAE results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for SVC Sigmoid

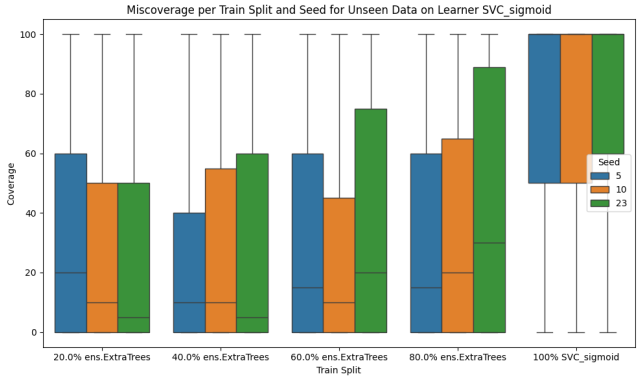

Figure 11: Miscoverage results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for SVC Sigmoid

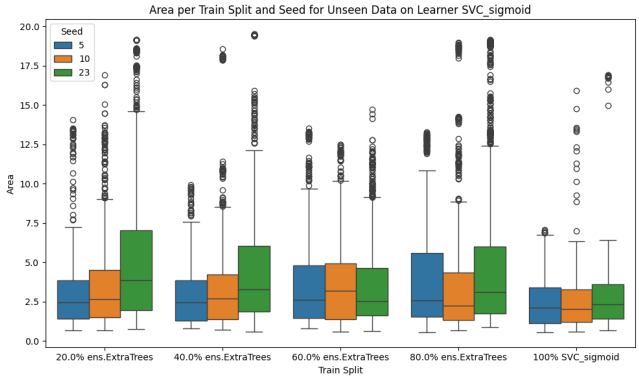

Figure 12: Area results for mixed training with Extra Trees and SVC Sigmoid training splits evaluated on Unseen Data for SVC Sigmoid

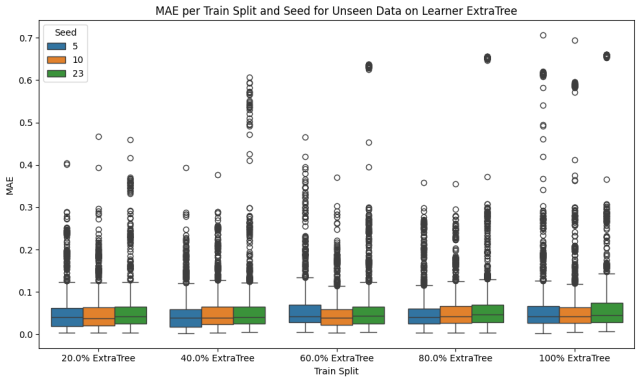

Figure 13: MAE results for mixed training with Extra Tree and Perceptron training splits evaluated on Unseen Data for Extra Tree

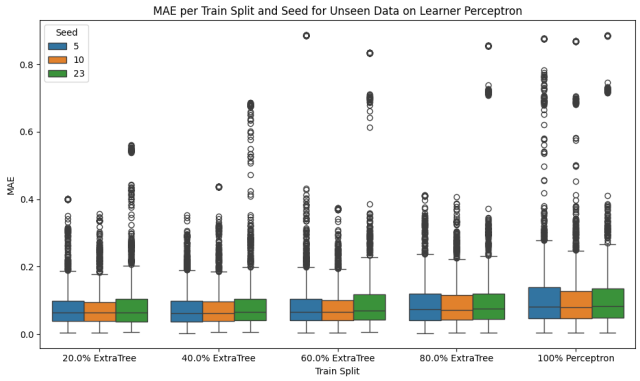

Figure 16: MAE results for mixed training with Extra Tree and Perceptron training splits evaluated on Unseen Data for Perceptron

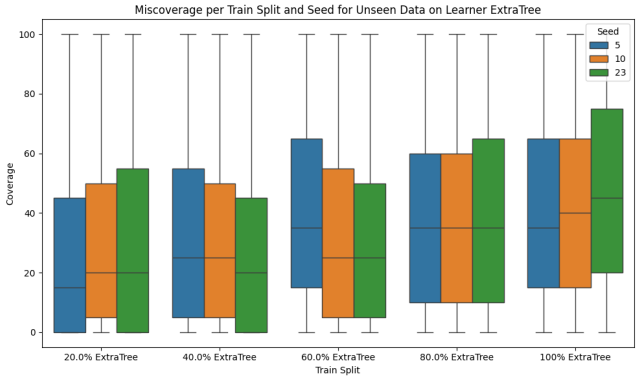

Figure 14: Miscoverage results for mixed training with Extra Tree and Perceptron training splits evaluated on Unseen Data for Extra Tree

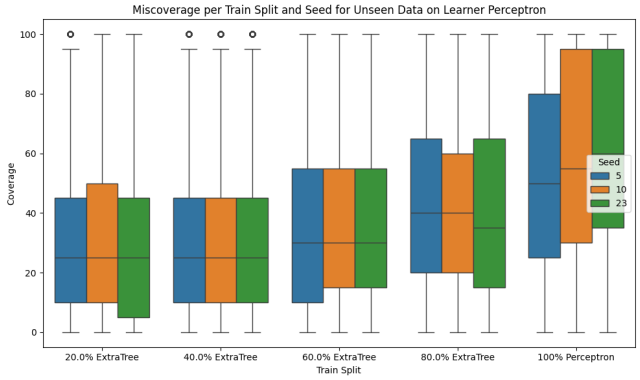

Figure 17: Miscoverage results for mixed training with Extra Tree and Perceptron training splits evaluated on Unseen Data for Perceptron

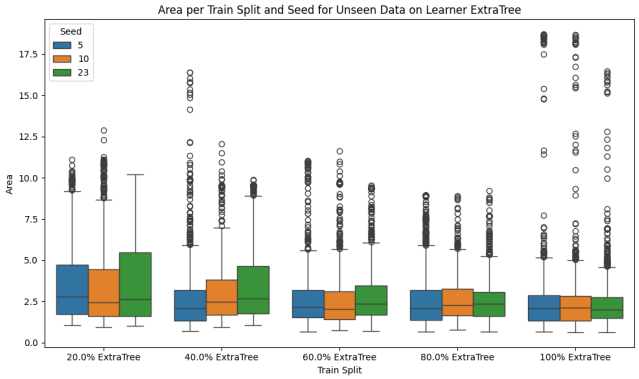

Figure 15: Area results for mixed training with Extra Tree and Perceptron training splits evaluated on Unseen Data for Extra Tree

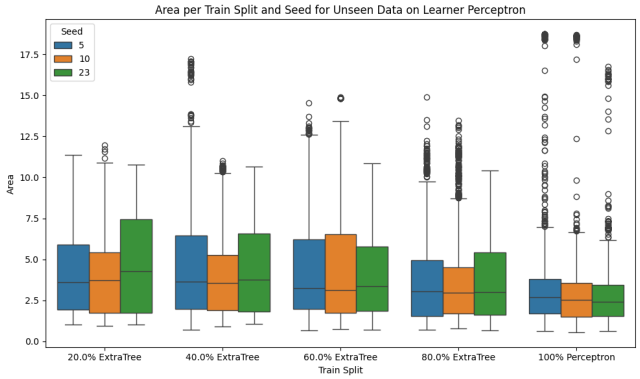

Figure 18: Area results for mixed training with Extra Tree and Perceptron training splits evaluated on Unseen Data for Perceptron

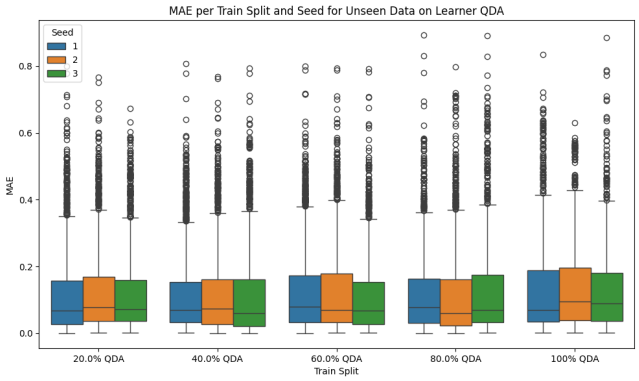

Figure 19: MAE results for mixed training with QDA and SVC Sigmoid training splits evaluated on Unseen Data for QDA

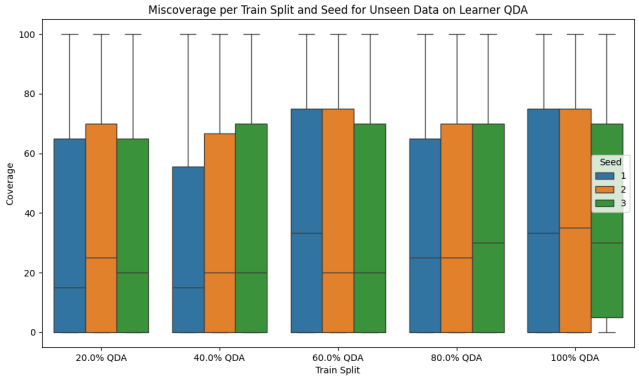

Figure 20: Miscoverage results for mixed training with QDA and SVC Sigmoid training splits evaluated on Unseen Data for QDA

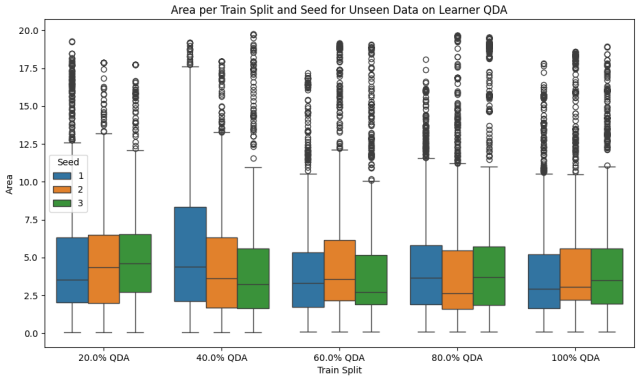

Figure 21: Area results for mixed training with QDA and SVC Sigmoid training splits evaluated on Unseen Data for QDA

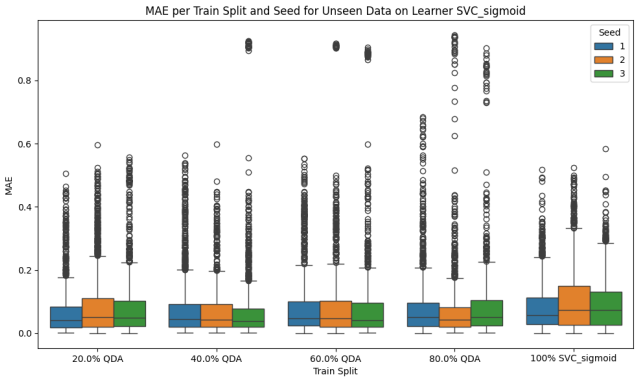

Figure 22: MAE results for mixed training with QDA and SVC Sigmoid training splits evaluated on Unseen Data for SVC Sigmoid

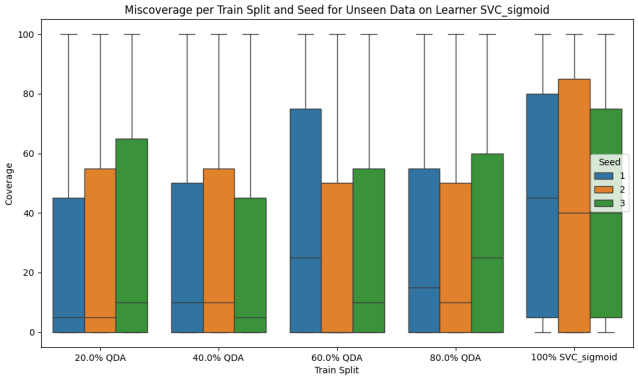

Figure 23: Miscoverage results for mixed training with QDA and SVC Sigmoid training splits evaluated on Unseen Data for SVC Sigmoid

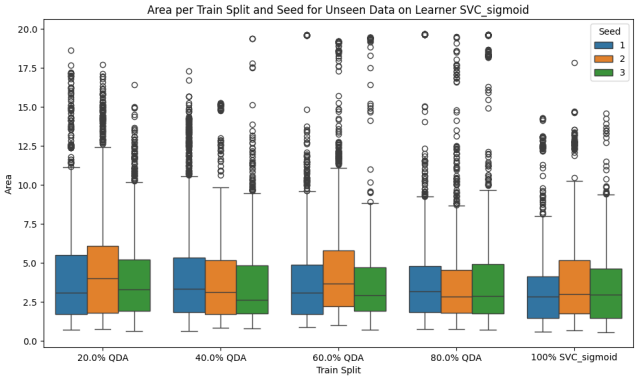

Figure 24: Area results for mixed training with QDA and SVC Sigmoid training splits evaluated on Unseen Data for SVC Sigmoid

B Results of Statistical tests

Mixed training for learners ens.ExtraTrees and SVC_sigmoid evaluated on Unseen Data for ens.ExtraTrees.

```
==============================
STATISTICAL ANALYSIS FOR SEED 5
==============================
--- MAE (Seed 5) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2: less
Split 0.8 vs 0.6: greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4: greater
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
--- Miscoverage (Seed 5) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2: less
--- Area (Seed 5) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
==============================
STATISTICAL ANALYSIS FOR SEED 10
==============================
--- MAE (Seed 10) ---
Split 1.0 vs 0.8: greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4: not significant
Split 1.0 vs 0.2: not significant
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4: less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2: less
--- Miscoverage (Seed 10) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:greater
--- Area (Seed 10) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4: not significant
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
==============================
STATISTICAL ANALYSIS FOR SEED 23
==============================
--- MAE (Seed 23) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4: not significant
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6: not significant
Split 0.8 vs 0.4: less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4: less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
--- Miscoverage (Seed 23) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6: not significant
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:greater
--- Area (Seed 23) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less

Mixed training for learners
ens.ExtraTrees and SVC_sigmoid
evaluated on Unseen Data for SVC_sigmoid.
==============================
STATISTICAL ANALYSIS FOR SEED 5
==============================
--- MAE (Seed 5) ---
Split 1.0 vs 0.8:(*) less
Split 1.0 vs 0.6: not significant
```

```
Split 1.0 vs 0.4:(*) greater
Split 1.0 vs 0.2:(*) greater
Split 0.8 vs 0.6:(*) greater
Split 0.8 vs 0.4:(*) greater
Split 0.8 vs 0.2:(*) greater
Split 0.6 vs 0.4:(*) greater
Split 0.6 vs 0.2:(*) greater
Split 0.4 vs 0.2:(*) less
--- Miscoverage (Seed 5) ---
Split 1.0 vs 0.8:(*) greater
Split 1.0 vs 0.6:(*) greater
Split 1.0 vs 0.4:(*) greater
Split 1.0 vs 0.2:(*) greater
Split 0.8 vs 0.6:(*) greater
Split 0.8 vs 0.4:(*) greater
Split 0.8 vs 0.2: not significant
Split 0.6 vs 0.4:(*) greater
Split 0.6 vs 0.2: not significant
Split 0.4 vs 0.2:(*) less
--- Area (Seed 5) ---
Split 1.0 vs 0.8:(*) less
Split 1.0 vs 0.6:(*) less
Split 1.0 vs 0.4: not significant
Split 1.0 vs 0.2: not significant
Split 0.8 vs 0.6:(*) greater
Split 0.8 vs 0.4:(*) greater
Split 0.8 vs 0.2:(*) greater
Split 0.6 vs 0.4:(*) greater
Split 0.6 vs 0.2:(*) greater
Split 0.4 vs 0.2:(*) greater
==============================
STATISTICAL ANALYSIS FOR SEED 10
==============================
--- MAE (Seed 10) ---
Split 1.0 vs 0.8: not significant
Split 1.0 vs 0.6:(*) greater
Split 1.0 vs 0.4:(*) greater
Split 1.0 vs 0.2:(*) greater
Split 0.8 vs 0.6:(*) greater
Split 0.8 vs 0.4:(*) greater
Split 0.8 vs 0.2:(*) greater
Split 0.6 vs 0.4:(*) greater
Split 0.6 vs 0.2:(*) greater
Split 0.4 vs 0.2: not significant
--- Miscoverage (Seed 10) ---
Split 1.0 vs 0.8:(*) greater
Split 1.0 vs 0.6:(*) greater
Split 1.0 vs 0.4:(*) greater
Split 1.0 vs 0.2:(*) greater
Split 0.8 vs 0.6:(*) greater
Split 0.8 vs 0.4:(*) greater
Split 0.8 vs 0.2:(*) greater
Split 0.6 vs 0.4:(*) less
Split 0.6 vs 0.2: not significant
Split 0.4 vs 0.2: greater
--- Area (Seed 10) ---
Split 1.0 vs 0.8:(*) less
Split 1.0 vs 0.6:(*) less
Split 1.0 vs 0.4:(*) less
Split 1.0 vs 0.2:(*) less
Split 0.8 vs 0.6:(*) less
Split 0.8 vs 0.4: not significant
Split 0.8 vs 0.2: not significant
Split 0.6 vs 0.4:(*) greater
Split 0.6 vs 0.2:(*) greater
Split 0.4 vs 0.2:(*) greater
==============================
STATISTICAL ANALYSIS FOR SEED 23
==============================
--- MAE (Seed 23) ---
Split 1.0 vs 0.8:(*) less
Split 1.0 vs 0.6:(*) less
Split 1.0 vs 0.4: not significant
Split 1.0 vs 0.2:(*) greater
Split 0.8 vs 0.6:(*) greater
Split 0.8 vs 0.4:(*) greater
Split 0.8 vs 0.2:(*) greater
Split 0.6 vs 0.4:(*) greater
Split 0.6 vs 0.2:(*) greater
Split 0.4 vs 0.2:(*) greater
--- Miscoverage (Seed 23) ---
Split 1.0 vs 0.8: not significant
Split 1.0 vs 0.6:(*) greater
Split 1.0 vs 0.4:(*) greater
Split 1.0 vs 0.2:(*) greater
Split 0.8 vs 0.6:(*) greater
Split 0.8 vs 0.4:(*) greater
Split 0.8 vs 0.2:(*) greater
Split 0.6 vs 0.4:(*) greater
Split 0.6 vs 0.2:(*) greater
Split 0.4 vs 0.2:(*) greater
--- Area (Seed 23) ---
Split 1.0 vs 0.8:(*) less
Split 1.0 vs 0.6:(*) less
Split 1.0 vs 0.4:(*) less
Split 1.0 vs 0.2:(*) less
Split 0.8 vs 0.6:(*) greater
Split 0.8 vs 0.4: not significant
Split 0.8 vs 0.2: less
Split 0.6 vs 0.4:(*) less
Split 0.6 vs 0.2:(*) less
Split 0.4 vs 0.2:(*) less

Mixed training for learners ExtraTree and
Perceptron evaluated on Unseen Data for ExtraTree.
==============================
STATISTICAL ANALYSIS FOR SEED 5
==============================
--- MAE (Seed 5) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:not significant
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
```

```
Split 0.4 vs 0.2:less
--- Miscoverage (Seed 5) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:not significant
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:greater
--- Area (Seed 5) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:not significant
Split 0.8 vs 0.4:not significant
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
==============================
STATISTICAL ANALYSIS FOR SEED 10
==============================
--- MAE (Seed 10) ---
Split 1.0 vs 0.8:not significant
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:not significant
--- Miscoverage (Seed 10) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:not significant
--- Area (Seed 10) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:not significant
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
==============================
STATISTICAL ANALYSIS FOR SEED 23
==============================
--- MAE (Seed 23) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:not significant
--- Miscoverage (Seed 23) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:less
--- Area (Seed 23) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less

Mixed training for learners ExtraTree and
Perceptron evaluated on Unseen Data for Perceptron.
==============================
STATISTICAL ANALYSIS FOR SEED 5
==============================
--- MAE (Seed 5) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:not significant
--- Miscoverage (Seed 5) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:greater
```

```
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:greater
--- Area (Seed 5) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
==============================
STATISTICAL ANALYSIS FOR SEED 10
==============================
--- MAE (Seed 10) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:not significant
--- Miscoverage (Seed 10) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:less
--- Area (Seed 10) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
==============================
STATISTICAL ANALYSIS FOR SEED 23
==============================
--- MAE (Seed 23) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
--- Miscoverage (Seed 23) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:greater
--- Area (Seed 23) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:not significant
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less

Mixed training for learners QDA and SVC_sigmoid
evaluated on Unseen Data for QDA.
==============================
STATISTICAL ANALYSIS FOR SEED 1
==============================
--- MAE (Seed 1) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:not significant
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:not significant
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:greater
--- Miscoverage (Seed 1) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:less
--- Area (Seed 1) ---
Split 1.0 vs 0.8:less
```

```
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:greater
==============================
STATISTICAL ANALYSIS FOR SEED 2
==============================
--- MAE (Seed 2) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:less
--- Miscoverage (Seed 2) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:not significant
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:not significant
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:not significant
Split 0.4 vs 0.2:less
--- Area (Seed 2) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:not significant
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:not significant
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
==============================
STATISTICAL ANALYSIS FOR SEED 3
==============================
--- MAE (Seed 3) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:not significant
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
--- Miscoverage (Seed 3) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:not significant
Split 0.6 vs 0.2:not significant
Split 0.4 vs 0.2:not significant
--- Area (Seed 3) ---
Split 1.0 vs 0.8:not significant
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:not significant
Split 0.8 vs 0.4:not significant
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:not significant
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less

Mixed training for learners QDA and SVC_sigmoid
evaluated on Unseen Data for SVC_sigmoid.
==============================
STATISTICAL ANALYSIS FOR SEED 1
==============================
--- MAE (Seed 1) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:not significant
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:greater
--- Miscoverage (Seed 1) ---
Split 1.0 vs 0.8:not significant
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:greater
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:greater
Split 0.4 vs 0.2:greater
--- Area (Seed 1) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
```

```
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:greater
==============================
STATISTICAL ANALYSIS FOR SEED 2
==============================
--- MAE (Seed 2) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:not significant
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
--- Miscoverage (Seed 2) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:not significant
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:not significant
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:not significant
--- Area (Seed 2) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:less
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:not significant
Split 0.4 vs 0.2:less
==============================
STATISTICAL ANALYSIS FOR SEED 3
==============================
--- MAE (Seed 3) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:not significant
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:not significant
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:not significant
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
--- Miscoverage (Seed 3) ---
Split 1.0 vs 0.8:greater
Split 1.0 vs 0.6:greater
Split 1.0 vs 0.4:greater
Split 1.0 vs 0.2:greater
Split 0.8 vs 0.6:greater
Split 0.8 vs 0.4:greater
Split 0.8 vs 0.2:not significant
Split 0.6 vs 0.4:greater
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:less
--- Area (Seed 3) ---
Split 1.0 vs 0.8:less
Split 1.0 vs 0.6:less
Split 1.0 vs 0.4:less
Split 1.0 vs 0.2:less
Split 0.8 vs 0.6:not significant
Split 0.8 vs 0.4:less
Split 0.8 vs 0.2:less
Split 0.6 vs 0.4:less
Split 0.6 vs 0.2:less
Split 0.4 vs 0.2:not significant
```

C Usage of LLM