A Case for Dataset Specific Profiling

Seth Ockerman^{1,2}, John Wu², Christopher Stewart² Grand Valley State University¹, The Ohio State University²

Abstract

Data-driven science is an emerging paradigm where scientific discoveries depend on the execution of computational AI models against rich, discipline-specific datasets. With modern machine learning frameworks, anyone can develop and execute computational models that reveal concepts hidden in the data that could enable scientific applications. For important and widely used datasets, computing the performance of every computational model that can run against a dataset is cost prohibitive in terms of cloud resources. Benchmarking approaches used in practice use representative datasets to infer performance without actually executing models. While practicable, these approaches limit extensive dataset profiling to a few datasets and introduce bias that favors models suited for representative datasets. As a result, each dataset's unique characteristics are left unexplored and subpar models are selected based on inference from generalized datasets. This necessitates a new paradigm that introduces dataset profiling into the model selection process. To demonstrate the need for dataset-specific profiling, we answer two questions: (1) Can scientific datasets significantly permute the rankorder of computational models compared to widely used representative datasets? (2) If so, could lightweight model execution improve benchmarking accuracy? Taken together, the answers to these questions lay the foundation for a new dataset-aware benchmarking paradigm.

1 Introduction

With the rise of data-driven science, an increasing number of large, discipline-specific image datasets are being introduced to the public [21, 45, 46, 48]. However, despite a widespread increase in datasets, an analysis by a team of Google and Berkeley researchers found that a very small number of datasets are used for computer vision benchmarking [17]. This is unsurprising given the popularity of measuring a learning architecture's efficacy by testing it against wellknown datasets (e.g. Imagenet [8], CIFAR-10 [19], MNIST [22], etc). The small number of benchmarking datasets has narrowed the scope of novel model design to a limited domain. Given the importance of data in the model design process and the relatively few datasets that are used for benchmarking, it is worth asking if they function well as representative datasets across different disciplinary areas. In other words, are existing benchmarks good references for new datasets and corresponding workloads [31, 35]. Furthermore, can we take for granted the unique influence that data has on all elements of the design process (CNN filter dimensions, hyperparameter searching, reaction rules, etc.)

and assume that one learning architecture's success on one dataset will transfer to another?

This study advocates for the adoption of the data-driven paradigm through individual dataset profiling. We will quantify the differences between widely used vision datasets used for benchmarking and discipline-specific datasets. We reexamine the use of representative datasets and prove that this practice can skew optimization away from each dataset's unique data paradigm. In addition to advocating for the adoption of data-driven design, we describe a potential solution that can address the cost barrier that drives the existing benchmarking paradigm.

We recognize that individual dataset benchmarking is inherently difficult due to cost. For good reason, most researchers chose a single representative dataset to train and test their model against instead of running against every dataset in its application area. We describe early work that makes use of previous models' training cycles to predict a new model's final accuracy after just a few epochs of training, extending recent efforts to predict accuracy [41, 44].

This paper's main contributions are as follows. To prove the need for dataset-driven profiling, we profile a number of datasets across two model application areas: deep neural networks (DNNs) and particle swarm optimization (PSO) [26]. We examine the inherent differences in neural network performance on 8 different datasets through analysis of a variety of metrics. We also examine the differences in parameter estimations for 4 mass cytometry datasets. Finally, we present early work on an alternative method for DNN benchmarking that uses low-cost lightweight runs to predict test set performance after only a few epochs.

2 Related Work

This section is organized as follows. Section 2.1 will explore past work on dataset profiling and bias to prove the need for a new paradigm. Section 2.2 will describe new trends leading to more datasets and the challenges presented in classical benchmarking. Section 2.3 will provide an overview of past work predicting neural network accuracy from weights.

2.1 Bias in Benchmarking Datasets

It is standard design practice to test new models (trained weights, hyperparameter choices, and architecture itself) by running them against popular datasets like MNIST and CI-FAR ([19, 22]). This is in part due to the popularity of dataset competitions as a method to popularize your learning architecture. However, this popularity comes with drawbacks. An analysis by [12] found no statistical difference between

the performance of the top 10 algorithms in the 2010 PAS-CAL Visual Object Classes competition. This suggests the top algorithms are not fundamentally different from one another. Researchers worry that the lack of dataset diversity among popular datasets is causing models to learn from idiosyncrasies of the images rather than significant generalized characteristics ([27, 40]). [40] also found that models trained on one representative dataset tend to test poorly on other representative datasets of the same category (i.e types of cars). This is not surprising given models tend to favor their own test sets. However, it is concerning that supposedly representative datasets do not create models with high enough levels of generalization to transfer to other similarly representative datasets.

While the diversity of representative datasets has improved over time, they still suffer from limitations. Far more recent work (2021) found that neural networks were learning from noise in biomedical images datasets instead of the relevant medical features ([9]). These datasets were popular image benchmarks that many new algorithms and networks were tested against. While these datasets certainly provided a good sanity check for new approaches, they also can determine the success or failure of a new approach. This bias, as well as the similarity between the medical datasets, can hamper novel approaches from seeing widespread adoption.

Past works suggest current representative datasets have the potential to skew development toward a narrow solution space not representative of the complexity of real-world problems. In theory, the simple solution is to create a variety of datasets by sub-area which are perfectly representative. However, this is both impractical due to the black box nature of neural networks and impossible due to pure cost. A new benchmarking paradigm that enables lightweight low-cost model testing against specific datasets is needed.

2.2 A Surge in Datasets

From astronomy [30, 32] to agriculture [2, 4, 47, 48] to K-12 education [5, 28] to software bugs and analysis [34, 38, 39], the velocity of dataset creation has surged in recent years. While machine and deep learning algorithms seek to perform classification and segmentation on all datasets, the semantics around these operations vary for each dataset. In astronomy, outlier pixels are associated with supernovae. In agriculture, classification can distinguish healthy and unhealthy crops. In education, clustering algorithms to identify students in need of additional tutoring. We contend that these different use-cases impact the efficacy of machine learning algorithms, in terms of accuracy, training time and computational cost. With the emergence of IoT for passive data collection, every field can create datasets to improve decision making and understanding.

With the growing number of datasets, dataset management is increasingly important, especially for cost efficacy. Data commons have emerged as storage repositories specifically for datasets [13]. In addition, research on cost effective replication [3, 11, 34, 43], cloud processing models [7], and other systems level techniques have matured. While we believe that these types of systems techniques also warrant dataset-specific data-driven approaches, in this paper, we are concerned with dataset efficacy rather than efficiency. However, in future work, we will explore the unique computational demands imposed by different datasets.

2.3 The Significance of Weights in Predicting Neural Network Accuracy

The primary way past works have attempted to predict the final accuracy of a network is through the use of early training curves ([10]). However, both concurrent and recent work has shown a strong relationship between a network's weights and its characteristics and performance. [44] was able to use weights obtained early in a neural network's training process to predict its eventual testing accuracy with higher accuracy than existing learning curve-based approaches. Very recently, [41] found that using simple summary statistics based on network's fully trained weights could predict test set accuracy with an R^2 score of over 0.98. This presents a compelling case for more investigation into the use of weights to predict neural network performance.

[44] uses a variety of weight features to predict eventual accuracy based on early epoch weights. [41] creates a dataset of 32,000 small-scale neural network's fully trained weights mapped to their final test set accuracy. They then use that dataset (dubbed CNN Zoo) to train gradient boosting machines (GBMs) to predict the test set accuracy of large-scale models. While our early work (explored in 5) is inspired by both [44] and [41], it is also builds on them in a significant way. [41] uses a small 4-layer network with randomly initialized weights. To build on this, we extract features from a large deep neural network that incorporates pre-trained ImageNet weights. We theorize that transfer learning will enable us to reduce the number of hyperparameter configurations needed to create a representative solution space (i.e scale of 1000s vs less than 100). [41]'s work treated a network's training cycle as one unit of data to be mapped to final accuracy. We instead record data at an epoch level. This enables the prediction of final accuracy based on only a few training epochs. In addition, both [44] and [41] focus on classical datasets. We instead focus on complex domainspecific datasets which better accounts for the potential bias of representative datsaets.

3 Experimentation with Digital Agriculture and Deep Neural Networks

To prove the need for individual dataset profiling in the computer vision field, we compare model performance among domain-specific datasets (digital agriculture) and classical datasets. Our experiment will demonstrate two main ideas: 1) A set of models will rank differently relative to each depending on the category of dataset. 2) Each set of models ranks differently on an individual dataset basis. These ideas will demonstrate that representative datasets will always introduce some level of bias into the neural network design process.

3.1 Methodology

To measure the distance between model rankings among different datasets, we select four prevalent model architectures: InceptionV3, VGG16, EfficentNet, ResNet50 ([15, 33, 36, 37]). Each network's weights are randomly initialized. A small fully connected classification network is placed on top of each network to allow for variation in the number of classes (a sample fully connected network is shown in figure 1). We also include an early stopping mechanism to prevent overfitting and provide standardization across different datasets. Our early stop callback will end training after the validation accuracy has stopped improving for 10 epochs.

Model: "sequential_5"

Layer (type)	Output Shape	Param #
dense_23 (Dense)	(224, 224, 512)	2048
dense_24 (Dense)	(224, 224, 256)	131328
dense_25 (Dense)	(224, 224, 128)	32896
<pre>flatten_6 (Flatten)</pre>	(224, 28672)	Θ
dense_26 (Dense)	(224, 2)	57346
Total params: 223,618 Trainable params: 223,618 Non-trainable params: θ		

Fig. 1. Small Fully Connected Classification Network

We train and test these networks using two categories of datasets: classical datasets and digital agriculture datasets. We select 4 classical datasets based on popularity: CIFAR-10, CIFAR-100, imagenette2 (a subset of Imagenet), and MNIST ([8, 19, 22]). The digital agriculture datasets selected are as follows: fruits-360, PlantVillage, weed seedlings, and leaf defoliation dataset ([1, 16, 25, 48]). We select these datasets because they each represent a fundamental task in digital agriculture (e.g. fruit classification, drone-based defoliation detection, diseased plant classification, etc.). More details on each dataset can be found in table 1.

We run all four networks against the eight datasets and collect five metrics: test set accuracy at the 20% training epoch, 75% training epoch, and the last training epoch; sparse categorical crossentropy loss; and epochs run. In this case, the epoch metric functions as a loose cost approximation because the early stopping mechanism attempts to detect the number of epochs required for convergence.

Dataset	Classes	Images
CIFAR-10	10	60,000
CIFAR-100	100	60,000
MNIST	10	60,000
Imagenette	10	13,000
Leaf Defoliation Dataset	2	97,395
Fruits-360	131	90,483
PlantVillage	38	87,000
Weed Seedlings	8	34,666

 Table 1. Individual Dataset Breakdown

Using these metrics, we create ranking vectors for each dataset. Each model is mapped to an arbitrary index in the vector, resulting in a base encoding vector as follows <InceptionV3, VGG16, EfficentNet, ResNet50>. For each metric and specific dataset (i.e accuracy, loss, or epochs run), we enter the ranking of that model into its encoded index. For example, the Leaf Defoliation ranked by accuracy would create the vector <3, 4, 1, 2>. In contrast, MNIST ranked by accurary would create the vector <1, 2, 3, 4>. We repeat this process for all eight datasets and create five ranking vectors per dataset.

By focusing on rankings, we isolate and standardize the effectiveness of a given DNN relative to a specific dataset. This avoids the inherent bias of comparing network accuracy across different datasets, which is often common practice. To measure the distance between rankings, we select CIFAR-10 as our base vector ([19]). To measure the distance between datasets, we use Euclidean and Kendal Tau distances.

3.2 Results: The Inherent Differences of Classical and Discipline Specific Datasets

After running tests across a variety of metrics, we find that digital agricultural datasets display fundamentally different properties than classical datasets. Sections 3.2.1, 3.2.2, 3.2.3, and 3.2.4 will explore our results graphically. Section 3.2.5 will provide a brief numerical analysis of our results related to accuracy.

3.2.1 Accuracy. Focusing specifically on final testing accuracy, we discover a significant difference between the rankings of digital agriculture and classical datasets. Figure 2 shows two different distance metrics that measure the distance between a dataset's accurary ranking vector relative to CIFAR-10's accuracy ranking vector. Both CIFAR-100 and ImageNet have identical ranking vectors to CIFAR-10. In contrast, all four of the digital agriculture datasets create significantly different rank vectors. This indicates that for the purpose of testing models against other classical dataset is an excellent benchmarking technique. However, against domain-specific datasets, such as digital agriculture datasets,

the same models perform at highly different levels of effectiveness. These findings support the need for dataset aware benchmarking.

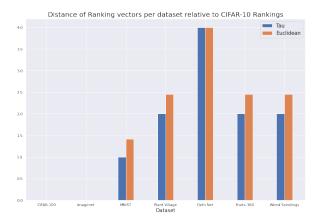


Fig. 2. Final Test Set Accuracy Rankings relative to CIFAR-10

3.2.2 Loss. In addition to final test set accuracy after training, each dataset shows different loss rankings. Figure 3 shows the distance between a given dataset and CIFAR-10 based on three types of ranking vectors: final test set accuracy, loss, and epochs run. Once again, both CIFAR-100's and ImageNet's loss ranking vectors are identical to CIFAR-10. PlantVillage and the Leaf Defoliation dataset vectors are significantly different from all of the classical image datasets. Interestingly, MNIST and Fruits-360 display the same distance from CIFAR-10, while Weed Seedlings is identical to CIFAR-10 in terms of loss. We theorize this is due to the relative simplicity of Fruits-360 and Weed Seedlings compared to the other digital agriculture datasets. Pure object classification (fruit, weeds) is a simple task compared to detecting subtle differences that indicate defoliation or disease. This would allow for higher levels of generalization, enabling similar performance of standardized models from one dataset to the next. In the future, an investigation into an automatic dataset complexity analysis might be warranted to improve the dataset-aware benchmarking paradigm. Regardless, the discipline-specific datasets show an average higher distance in loss rankings compared to the classical datasets.

3.2.3 Epochs. The green bar in figure 3 represents distance between epoch ranking vectors relative to CIFAR-10. The Epoch ranking measured the number of epochs the model ran until the early stopping mechanism was triggered. This very loosely allows us to estimate the cost of training a given network. We use this to create a ranking vector of epochs (ranked in ascending order to prioritize low-cost networks). Our ranking system is by no means a perfect measure of cost because time per epoch can vary greatly between datasets, but cost estimation is not the focus of this paper. We discover that per dataset there is a great degree of variance

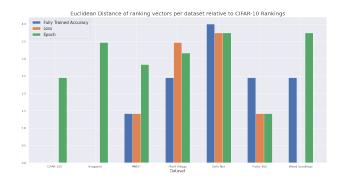


Fig. 3. Euclidean Distance of Ranking vectors relative to CIFAR-10

in ranking vectors. Among ranking vectors, epoch vectors showed by far the most diversity compared to any of the other metrics we collected. The general grouping between classical and digital agricultural datasets that is shown in the accuracy and loss rankings is not shown in the epoch rankings. The lack of grouping by dataset category suggests that convergence significantly varies even within specific categories of datasets, which is unsurprising given convergence is notoriously difficult to estimate. The variance in epoch rankings by dataset supports the need for dataset aware benchmarking, particularly in the area of cost estimation. While our study focuses on estimating test set accuracy, a better method of cost estimation per dataset is a potentially interesting future study.

3.2.4 Relative to Leaf Defoliation Dataset. We discovered a strong grouping of classical dataset rankings, so we investigate if that same grouping exists among the three digital agriculture datasets we selected. To do this, we measure the distance between ranking vectors relative to the Leaf Defoliation Dataset instead of CIFAR-10. Figure 4 shows the distance between final testing accuracy, loss, and epoch vectors relative to the Leaf Defoliation dataset. We clearly see the general grouping of three of the four classical datasets across all three metrics. MNIST is similar to the other classical datasets in terms of accuracy, but varies in loss and epochs. This same degree of consistency does not exist among the digital agricultural datasets. There are some similarities between Weed Seedlings and Fruit-360, however, PlantVillage remains distinct compared to its digital agricultural counterparts. These findings suggest more significant variation between the rankings of models on digital agricultural datasets and classical image datasets.

3.2.5 Numerical Analysis. To try to improve our insights into the difference between accuracy rankings across different datasets, we calculate each datasets accurary vector distance relative to all other datasets. We repeat this process and create average distances from each category of dataset. The average distance of classical datasets from itself is 0.71,

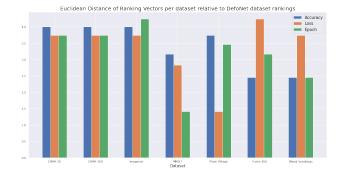


Fig. 4. Euclidean Distance of Ranking vectors relative to Leaf Defoliation Dataset

while the average distance of a digital agriculture dataset from a classical dataset is 2.67. The almost four times increase in ranking distance shows an inherent difference in model performance by dataset group, suggesting that testing digital agriculture models on classical datasets will bias them away from their own data paradigm.

The average distance of digital agriculture datasets from themselves is 2.69, while the average classical datasets distance from digital agriculture datasets is 2.67. The near equivalent distances suggest that model performance varies significantly at a dataset level even among groupings such as digital agriculture. The increased variation in model performance among digital agriculture datasets can partially be explained by the large discrepancy in distance between self accuracy of classical datasets (0.71) and digital agriculture datasets (2.69). These findings indicate that classical datasets are significantly better at benchmarking themselves than digital agriculture datasets are. We theorize the increased difficulty in self-benchmarking is due to the high diversity in discipline-specific datasets when compared to classical datasets. Generalized groupings such as digital agriculture datasets or classical datasets are not adequate to benchmark niche and domain-specific applications. A new paradigm for benchmarking is needed. A summary of all numbers discussed in this section can be found in tables 2 and 3.

Dataset	Classical	Argo.
CIFAR-10	0.47	2.84
CIFAR-100	0.47	2.84
MNIST	1.41	2.20
Imagenette	0.47	2.84
Leaf Defoliation Dataset	3.79	2.88
Fruits-360	2.19	2.06
PlantVillage	2.54	3.74
Weed Seedlings	2.19	2.06

Table 2. Average distance of a dataset from datasets of its type

	Classical	Argo.
Classical	0.71	2.68
Argo.	2.68	2.69

Table 3. Average dataset category distance

3.3 Summary

We demonstrated that neural networks perform differently on classical datasets than on domain-specific areas datasets such as digital agriculture. The average distance of rankings between a digital agriculture dataset and a classical dataset is nearly 4 times the distance of a classical dataset from itself. Additionally, we discover that digital agriculture datasets are not significantly better for benchmarking themselves than classical datasets. As such, the arbitrary grouping of dataset by sub-area (i.e. classical vs digital agriculture) is not a solution to address differences between datasets. The inherent difference between complex datasets makes it nearly impossible to perform nonbiased benchmarking through the use of a single representative dataset. Domain-specific datasets are not sufficiently better at benchmarking than classical datasets. All of this together demonstrates the need for the adoption of individual dataset profiling in the neural network community.

4 Particle Swarm Optimization in Mass Cytometry Datasets

This difference in datasets is further seen in even niche domains outside of vision and neural network learning architectures. Such is the case in Mass Cytometry datasets where biophysicists commonly model biological systems using ordinary differential equation (ODE) reaction networks [20, 23, 29]. In these networks, parameter estimation, specifically of rate constants, is performed with a variety of "learning" or optimization heuristics. One commonly used heuristic is the biologically inspired algorithm, particle swarm optimization (PSO). In this heuristic, particles search the parameter-cost space for the optimal set of rate constants that best fit the observed trajectories of time-stamped abundance data in Mass Cytometry. The PSO's update mechanism we apply in our methodology is driven by three weights, (1) the influence of a particle's current best estimate, (2) the influence of the global best estimate, and (3) each particle's inertia. To show the aforementioned differences in learning architecture performance by dataset, we provide euclidean distances of ranking vectors of five configurations of PSO weights for four different mass cytometry datasets.

4.1 Methodology

The four datasets contain both experimental and simulated data. In the case of real data, time-stamped protein species of CD8 T cells [18] and CD56 cells [24] were measured. For

the two simulated datasets, initial conditions were randomly sampled from multivariate lognormal distributions. Then, using the ODE reaction networks defined in Bionetgen [14], initial conditions were evolved to set time points. Each dataset's time points are shown in Table 4.

, ,

Dataset	Times (minutes)
CD56 NK Cells	16, 32
CD8 T Cells	1, 2
Simulated 1	0, 0.5, 2, 10, 20, 30
Simulated 2	0, 1.5

Table 4. Times Points in Each Dataset

All reaction networks are shown and defined in Figure 5. Note that the θ_i 's are the parameters being estimated and that the labels on the left correspond to their respective datasets. Each of the five different configurations of PSO weights and their respective labels are shown in Table 5. To provide reasonable computational constraints in the context of run time costs, each PSO configuration was standardized to 200 particles and 20 epochs. Each PSO configuration is run 30 times against each dataset, giving us a set of estimates for each dataset's respective ODE models. A more rigorous explanation of the PSO used to generate the data can be found in [42].

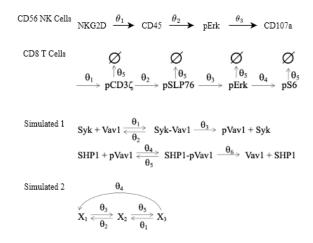


Fig. 5. ODE Reaction Networks used in Bionetgen

Configuration	Particle Best Weight	Global Best Weight	Particle Inertia
А	3.0	1.0	6.0
В	4.0	2.0	5.0
С	5.0	3.0	4.0
D	5.0	2.0	10.0
E	3.0	4.0	3.0

Table 5. PSO Weight Configurations

We ranked each PSO configuration by the average standard deviation of estimates and by cost. In this case, smaller deviations and costs are ranked higher (i.e the smallest standard deviation PSO configuration would be ranked one). We define cost to be the square difference of means, variances, and covariances between the observed data and the data generated from estimates. We chose these two metrics because the range of estimates roughly indicates the efficiency of PSO estimation while the cost indicates a level of dataset fit.

Similar to the ranking method performed in Section 3, we encode configurations of PSO to a vector. In this case, we map each configuration to an index in the vector, creating the mapping <A, B, C, D, E> (e.g. Configuration A is mapped to index 0 of the vector). Using this mapping, ranking vectors of PSO configurations were formed for each dataset. For instance, when estimating using the CD8 T Cells dataset, a ranking of D, E, C, B, and A in descending order produces the vector <5, 4, 3, 1, 2>). Once encoded, relative distances of ranking vectors were computed with respect to each dataset.

4.2 Differences Across Mass Cytometry Datasets

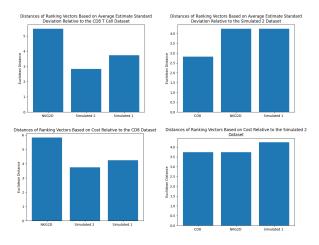


Fig. 6. Distances Between Ranking Vectors of PSO Configurations

For sake of brevity, only the euclidean distances with respect to the CD8 T cells and Simulated 2 are shown in Figure 6. As shown by the bottom two bar charts in Figure 6, in terms of cost, rankings dramatically differ. This difference in rankings implies that the optimal PSO configuration in terms of dataset fit can be dataset-specific.

Furthermore, this trend continues to be seen in the standard deviation rankings. In this case, smaller standard deviations imply that each repeated PSO execution converges on a smaller interval of estimates. Figure 6 reveals that certain PSO configurations converge faster depending on the dataset.

Also, observe that the ranking differences remain between the two synthetic datasets where although reaction networks differ their source probability distributions remained the same, demonstrating that structural differences within a dataset may lead to different optimal PSO configurations.

Even in fields outside of computer vision such as computational biology, datasets fundamentally influence model performance. Ultimately, these discrepancies further reinforce the need for dataset specific profiling and benchmarking.

5 Early Work on Lightweight Model Profiling

Sections 3 and 4 demonstrated that reusing learning architectures between datasets reduces the overall effectiveness of each dataset-model combination. A simple but effective solution to the demonstrated problem is to view the dataset profiling process as part of hyperparameter searching. However, profiling each dataset-model combination makes the naive assumption that cost can grow towards infinity. Individual dataset profiling is not feasible on a large scale for the majority of developers. A new method of model benchmarking and dataset profiling is needed. This section details our experimentation using early training weights to predict final training accuracy.

5.1 Methodology

We design an experiment to test the use of weights to predict final testing accuracy on complex a domain-specific dataset: the Leaf Defoliation Dataset ([48]. To create a solution space, we test a variety of hyperparameter configurations using VGG16 DNN with pre-trained ImageNet weights combined with a small, fully connected classification neural network (a sample is shown in 1).

We chose 35 different hyperparameter configurations to explore that vary optimizer, learning rate, and final layer activation function. We run each of the 35 configurations for 75 epochs and record the final test set accuracy of each configuration.

Each epoch we save the weights to extract summary statistics from them. We calculate the mean, variance, and q-th percentiles where $q \in \{0, 25, 50, 75, 100\}$ ([41]) for biases and kernel weights separately. WWe calculate these statistics for each neural network layer, creating a 2x7 vector for each layer. Combining all 17 layers into a single matrix, we generate a 17x2x7 representation of means, variances, and percentiles. This matrix is then mapped to the final testing accuracy of its respective model's configuration. Because we create mappings at an epoch level, we significantly increase the sample space we explore. In total, we create 2625 accuracy mappings from the Leaf Defoliation Dataset.

We perform an 80/20 train/test split of our vector accuracy mappings. In contrast to typical train/test splits, we do not randomize the placement of the mappings. Instead, we ensure the 20% in the test set is composed entirely of hyperparameter combinations that do not exist in the training set. The nonrandom nature of the test set is to prevent

. .

overfitting and leakage from the training dataset to the test set. Our approach results in a training set with 31 hyperparameter configurations and a testing set with four unseen hyperparameter configurations.

For prediction, we select gradient boosting machines implemented in XGBoost's gradient boosting forest package ([6]). We split our training set into a train and validation set at an 80% 20% ratio. Using that validation set, we perform hyperparameter tuning, resulting in a model of 128 estimators with a max depth of 7 per tree. A full breakdown of all selected hyperparameters can be found in 6.

Hyperparameter	Value
Objective	Regressive: Linear
Column Sample By Tree	0.3
Learning Rate	0.1
Max Depth	7
Alpha	40
Number of Estimators	128

Table 6. GBM hyperparamater selection

The testing data consists of the same number of hyperparameter configurations each time. However, we vary the percent of each model configuration's training cycle we include in the prediction process. By limiting the data inputted to the GBM to an artificial n-th epoch of training time, we simulate lightweight benchmarking runs. For example, by reducing input data to the first five epochs of weight data, we test the accuracy of predictions given only a fraction of the total training time. We select epochs 4, 8, 19, 38, 56, 76 which represent 5%, 10%, 25%, 50%, 75%, and 100% of the training time respectively.

5.2 Results

After hyperparameter tuning on the validation set, we test our trained model on each of the test splits. Using the entirety of the test data, we achieve an accuracy of 81.33% and a relative root mean squared error (RRMSE) of 0.196. Crucially, there is a minimal decrease in accuracy if we reduce the number of epochs we use as an input for our test set. Using only 4 epochs of input data from the test set (representing roughly 5% of its training time) we achieve an accuracy of 80.61% and an RRMSE of 0.201. This trend continues across all designated input data splits. Across all splits, there is less than a 1% change in accuracy and RRMSE. This indicates that the number of epochs of input data has little impact on the accuracy of a fully trained model. A full breakdown of accuracy and RRMSE by epoch of input data can be found in 7.

Epoch	Training Time (%)	Accuracy	RRMSE
4	5%	80.61%	0.201
8	10%	80.84%	0.199
19	25%	81.09%	0.197
38	50%	81.24%	0.197
56	75%	81.31%	0.196
75	100%	81.33%	0.197

Table 7. Accuracy and RRMSE of different percents of test

 set model training time

6 Conclusion

In section 3, we demonstrated that discipline-specific datasets significantly alter model performance compared to widespread classical benchmarking datasets. In addition, we discovered that the complexity of discipline-specific datasets causes them to suffer similar benchmarking limitations to classical datasets.

In section 4, we explored three key findings: (1) that dataset differences in "learning" is not limited to neural networks, (2) that this discrepancy remains within datasets of even a highly specific domain such as computational biology, and (3) that this difference in ranking vectors applies to both experimental as well as synthetic data where a dataset's probability distribution is known.

Our findings demonstrate that datasets shape model performance in fundamental ways, necessitating that dataset profiling becomes part of the machine learning design process.

6.1 Vision for the Future Of Our Early Work

Using GBMs, we achieve an accuracy of roughly 81% across all different splits of input data. Our early work (described in section 5) has not progressed enough to significantly change benchmarking practices. We hope, however, that our work will start a discussion about what a more developed datasetspecific benchmarking system could look like. It is especially promising that changing the percent of training time that was inputted into our model had little to no effect on its overall accuracy. Given a model trained on robust data, predicting the final accuracy of a neural network is possible after only a small number of epochs. This finding transforms lightweight weight-based benchmarking from a niche theory into a plausible widespread reality. However, for this to be possible, we need to save the weights of networks as they are trained and tested against a dataset. Using this approach, we could construct a weight-based solution space per dataset. This solution space could then be used to predict the performance of future learning architectures after just a few training epochs. Early prediction of model accuracy would exponentially reduce the cost of testing new architectures against new datasets, enabling more domain-specific models to emerge.

Despite being limited in time and resources, our approach showed initial success in predicting final network accuracy after only a few epochs. In future work, we hope to build on our findings by testing other types of neural networks (e.g. using a DNN instead of PSO on mass cytometry) and data-driven changes (explore more hyperparameter configurations, more epochs, more models, etc). We also hope to experiment with transferable prediction mechanisms to create a unified solution space for vastly different model architectures.

References

- M. A. Beck, C.-Y. Liu, C. P. Bidinosti, C. J. Henry, C. M. Godee, and M. Ajmani. An embedded system for the automated generation of labeled plant images to enable machine learning applications in agriculture. *PLOS ONE*, 15:1–23, 12 2020.
- [2] J. Boubin, J. Chumley, C. Stewart, and S. Khanal. Autonomic computing challenges in fully autonomous precision agriculture. In 2019 IEEE International Conference on Autonomic Computing (ICAC). IEEE, 2019.
- [3] J. Boubin and C. Stewart. Softwarepilot: Fully autonomous aerial systems made easier. In 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C), pages 250–251. IEEE, 2020.
- [4] J. G. Boubin, N. T. Babu, C. Stewart, J. Chumley, and S. Zhang. Managing edge resources for fully autonomous aerial systems. In *Proceedings* of the 4th ACM/IEEE Symposium on Edge Computing, pages 74–87. ACM, 2019.
- [5] J. Buele, V. M. López, L. Franklin Salazar, J.-H. Edisson, C. Reinoso, S. Carrillo, A. Soria, R. Andrango, and P. Urrutia-Urrutia. Interactive system to improve the skills of children with dyslexia: a preliminary study. In *Developments and advances in defense and security*, pages 439–449. Springer, 2020.
- [6] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.
- [7] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.
- [9] S. Dhar and L. Shamir. Evaluation of the benchmark datasets for testing the efficacy of deep convolutional neural networks. *Visual Informatics*, 5(3):92–101, 2021.
- [10] T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 3460–3468. AAAI Press, 2015.
- [11] Y. Du, Z. Xu, K. Zhang, J. Liu, J. Huang, and C. Stewart. Cost-effective strong consistency on scalable geo-diverse data replicas. *IEEE Transactions on Cloud Computing*, 2022.
- [12] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal* of Computer Vision, 88:303–338, 06 2010.
- [13] R. L. Grossman, A. Heath, M. Murphy, M. Patterson, and W. Wells. A case for data commons: toward data science as a service. *Computing* in science & engineering, 18(5):10–20, 2016.
- [14] L. A. Harris, J. S. Hogg, J.-J. Tapia, J. A. P. Sekar, S. Gupta, I. Korsunsky, A. Arora, D. Barua, R. P. Sheehan, and J. R. Faeder. BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics*, 32(21):3366–3368, 07 2016.

- [15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.
- [16] D. P. Hughes and M. Salathe. An open access repository of images on plant health to enable the development of mobile disease diagnostics, 2015.
- [17] B. Koch, E. Denton, A. Hanna, and J. Foster. Reduced, reused and recycled: The life of a dataset in machine learning research. 12 2021.
- [18] S. Krishnaswamy, M. H. Spitzer, M. Mingueneau, S. C. Bendall, O. Litvin, E. Stone, D. Pe'er, and G. P. Nolan. Conditional density-based analysis of t cell signaling in single-cell data. *Science*, 346(6213):1250689, 2014.
- [19] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.
- [20] G. La Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018.
- [21] Y. Lan, K. Huang, C. Yang, L. Lei, J. Ye, J. Zhang, W. Zeng, Y. Zhang, and J. Deng. Real-time identification of rice weeds by uav low-altitude remote sensing based on improved semantic segmentation model. *Remote Sensing*, 13(21):4370, 2021.
- [22] Y. LECUN. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/.
- [23] P. Loskot, K. Atitey, and L. Mihaylova. Comprehensive review of models and methods for inferences in bio-chemical reaction networks. *Frontiers in Genetics*, 10:549, 2019.
- [24] S. Mukherjee, H. Jensen, W. Stewart, D. Stewart, W. Ray, S.-Y. Chen, G. Nolan, L. Lanier, and J. Das. In silico modeling identifies cd45 as a regulator of il-2 synergy in the nkg2d-mediated activation of immature human nk cells. *Science Signaling*, 10, 06 2017.
- [25] H. Mureşan and M. Oltean. Fruit recognition from images using deep learning. Acta Universitatis Sapientiae, Informatica, 10:26–42, 06 2018.
- [26] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. Swarm intelligence, 1(1):33–57, 2007.
- [27] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. Russell, A. Torralba, C. Williams, J. Zhang, and A. Zisserman. *Dataset Issues in Object Recognition*, volume 4170, pages 29–48. 01 2007.
- [28] R. J. Rao, C. Stewart, A. Perez, and S. M. Renganathan. Assessing learning behavior and cognitive bias from web logs. In 2018 IEEE Frontiers in Education Conference (FIE), pages 1–5. IEEE, 2018.
- [29] J. A. Rohrs, P. Wang, and S. D. Finley. Understanding the dynamics of tcell activation in health and disease through the lens of computational modeling. *JCO clinical cancer informatics*, 3:1–8, 2019.
- [30] R. Scalzo, G. Aldering, P. Antilogus, C. Aragon, S. Bailey, C. Baltay, S. Bongard, C. Buton, M. Childress, N. Chotard, et al. Nearby supernova factory observations of sn 2007if: First total mass measurement of a super-chandrasekhar-mass progenitor. *The Astrophysical Journal*, 713(2):1073, 2010.
- [31] K. Shen, C. Stewart, C. Li, and X. Li. Reference-driven performance anomaly identification. ACM SIGMETRICS Performance Evaluation Review, 37(1):85–96, 2009.
- [32] J. M. Silverman, M. Ganeshalingam, W. Li, A. V. Filippenko, A. A. Miller, and D. Poznanski. Fourteen months of observations of the possible super-chandrasekhar mass type ia supernova 2009dc. *Monthly Notices* of the Royal Astronomical Society, 410(1):585–611, 2011.
- [33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556, 09 2014.
- [34] C. Stewart, A. Chakrabarti, and R. Griffith. Zoolander:efficiently meeting very strict, low-latency slos. In 10th International Conference on Autonomic Computing (ICAC 13), pages 265–277, 2013.
- [35] C. Stewart, M. Leventi, and K. Shen. Empirical examination of a collaborative web application. In 2008 IEEE International Symposium on Workload Characterization, pages 90–96. IEEE, 2008.

- [36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2818–2826, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society.
- [37] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. 97:6105–6114, 09–15 Jun 2019.
- [38] M. Taylor, J. Boubin, H. Chen, C. Stewart, and F. Qin. A study on software bugs in unmanned aircraft systems. In 2021 International Conference on Unmanned Aircraft Systems (ICUAS), pages 1439–1448. IEEE, 2021.
- [39] M. Taylor, H. Chen, F. Qin, and C. Stewart. Avis: In-situ model checking for unmanned aerial vehicles. In 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pages 471–483. IEEE, 2021.
- [40] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In CVPR 2011, pages 1521–1528, 2011.
- [41] T. Unterthiner, D. Keysers, S. Gelly, O. Bousquet, and I. O. Tolstikhin. Predicting neural network accuracy from weights. ArXiv, abs/2002.11448, 2020.
- [42] J. Wu, W. Stewart, C. Jayaprakash, and J. Das. Generalized method of moments improves parameter estimation in biochemical signaling models of time-stamped single-cell snapshot data. *bioRxiv*, 2022.
- [43] Z. Xu, C. Stewart, N. Deng, and X. Wang. Blending on-demand and spot instances to lower costs for in-memory storage. In *IEEE INFO-COM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.
- [44] Y. Yamada and T. Morimura. Weight features for predicting future model performance of deep neural networks. *IJCIA-16*, 2016.
- [45] M.-D. Yang, J. G. Boubin, H. P. Tsai, H.-H. Tseng, Y.-C. Hsu, and C. C. Stewart. Adaptive autonomous uav scouting for rice lodging assessment using edge computing with deep learning edanet. *Computers and Electronics in Agriculture*, 179:105817, 2020.
- [46] M.-D. Yang, H.-H. Tseng, Y.-C. Hsu, C.-Y. Yang, M.-H. Lai, and D.-H. Wu. A uav open dataset of rice paddies for deep learning practice. *Remote Sensing*, 13(7):1358, 2021.
- [47] Z. Zhang, J. Boubin, C. Stewart, and S. Khanal. Whole-field reinforcement learning: A fully autonomous aerial scouting method for precision agriculture. *Sensors*, 20(22):6585, 2020.
- [48] Z. Zhang, S. Khanal, A. Raudenbush, K. Tilmon, and C. Stewart. Assessing the efficacy of machine learning techniques to characterize soybean defoliation from unmanned aerial vehicles. *Computers and Electronics in Agriculture*, 193:106682, 02 2022.