algorithms

MDPI

Article

Synthetic Data-Based Algorithm Selection for Medical Image Classification Under Limited Data Availability

Maxim Zhabinets [1], Benjamin Tyler [1], Martin Lukac [2,*], Shinobu Nagayama [2], Ferdinand Molnár [3] and Michitaka Kameyama [4]

[1] School of Engineering and Digital Sciences, Nazarbayev University, Astana 010000, Kazakhstan; maxim.zhabinets@nu.edu.kz (M.Z.); btyler@nu.edu.kz (B.T.)
[2] Graduate School of Information Sciences, Hiroshima City University, Hiroshima 731-3166, Japan; s_naga@hiroshima-cu.ac.jp
[3] School of Sciences and Humanities, Nazarbayev University, Astana 010000, Kazakhstan; ferdinand.molnar@gmail.com
[4] Emeritus of Graduate School of Information Sciences, Tohoku University, Sendai 980-8577, Japan; michitaka.kameyama.c2@tohoku.ac.jp
* Correspondence: malu@hiroshima-cu.ac.jp; Tel.: +81-828301856

Abstract: The Algorithm selection approach improves performance by dynamically choosing the optimal Algorithm for each input instance. While this selection strategy has been extensively studied, the amount of data and their nature have not yet been investigated with respect to meta-learning, particularly in scenarios with limited data availability. This paper addresses a critical challenge: where additional data might not be available for training an Algorithm selector, and to implement a selection mechanism, data must be generated. Focusing on medical image classification, we investigate whether synthetic data can effectively train an Algorithm selector when real training data are scarce. Our methodology involves data generation using Generative Adversarial Network. To determine if Algorithm selection trained on synthetically generated data can achieve the same accuracy as if trained on real-world natural data, we systematically evaluate the data generative model using the smallest amount of data needed to choose the right Algorithm and to achieve the expected level of accuracy. Our experimental results demonstrate that using a small amount of real samples can provide enough information to a Generative Adversarial Network to synthesize a new dataset that, when used for training the Algorithm selection, improves image classification in some cases.

Keywords: algorithm selection; medical image classification; synthetic data; GAN

check for updates

Academic Editor: Sud Sudirman

Received: 28 March 2025
Revised: 6 May 2025
Accepted: 19 May 2025
Published: 25 May 2025

Citation: Zhabinets, M.; Tyler, B.; Lukac, M.; Nagayama, S.; Molnár, F.; Kameyama, M. Synthetic Data-Based Algorithm Selection for Medical Image Classification Under Limited Data Availability. *Algorithms* **2025**, *18*, 310. https://doi.org/10.3390/a18060310

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The problem of classification requires data to train the classifiers. A larger availability of data implies that, in general, we can train a more accurate classifier tool. However, constantly annotating new data for novel tasks, new specific cases, and samples is a costly task. In addition, adding only new data does not guarantee a successful improvement of the classifier. Rather, one needs to add meaningful data samples that would provide additional information to the classifier to learn [1]. Data selection and data generation for Machine Learning (ML) is a well-studied topic. For instance, in [2], features are selected to provide the best training support. In [3], data are statistically analyzed to provide the best coverage of the problem space. In general, when a dataset is prepared, the data are also statistically analyzed for different measures in order to represent a good sample distribution.

In order to improve the general training and generalization ability of the trained decision making Algorithm, data generation has been recently used to achieve moderate improvement in certain tasks. For instance, in [4], the generation and augmentation of a real dataset by synthetic data improved the generalization ability of the model, while using synthetic data for training large language models can lead to a serious performance reduction [5].

However, in a real-world context, no single Algorithm can provide best processing for each input instance. This effect has been described for instance in the framework of the no-free-lunch theorem [6] or complementary performance [7,8]. The no-free-lunch theorem states that for a given problem, no single Algorithm can solve all instances of the problem with the highest accuracy. The Algorithm complementarity expands this concepts into the observation that different Algorithms, while trained on the same dataset, will exhibit different per-instance accuracy due to Algorithm bias. The per-instance Algorithm selection is a meta strategy that aims to solve this problem by learning each Algorithm's performance over a certain number of samples and then generalizes this knowledge. For the improvement of the classification task, one starts from a set of pretrained component Algorithms. Then, we construct an Algorithm selection mechanism that chooses the best component Algorithm on a case-by-case basis. The general methodology is to train an Algorithm classifier using a set of meta-features that would capture Algorithm performance and related information. The standard approaches are, for instance, based on meta-features [9], reasoning [10], parallel execution [11], Algorithm configuration [12], etc. In all the cases, however, the data and the features must be able to describe the Algorithms' performance well enough to allow for a reliable Algorithm selection [13].

The training of the meta classifier, i.e., Algorithm selector, however, also requires data. The data can be either obtained from a set of runs of the component Algorithms on a training data or a set of solved instance problems. The main difference is that one can use the Algorithm evaluation as a conceptual step towards task change [14]. In ideal conditions, one would always want to have a specific dataset that would reveal as much information about the individual Algorithms as possible. However, in general, training data for systems in production might not be available and not enough novel annotated data might be generated. For this purpose, we consider a specific scenario for optimization. Let the problem of training an Algorithm selector be a function of available data. Considering a dataset $\mathbb{D}_T$ we analyze a) how much data are experimentally necessary to train an Algorithm selector and, b) how much data augmentation using data generation can improve the selector accuracy. In particular, we study what amount of natural and synthetic data is necessary to train an Algorithm selector that would outperform the best Algorithm, what properties of the selector would allow us to efficiently select component Algorithms, and how to generate synthetic data that would give the most accurate Algorithm selection accuracy. Finally, we evaluate the different approaches to Algorithm selection on input image classification.

While data augmentation by synthetic data has been effectively used in various machine learning tasks, so far there is no work related to Algorithm selection using synthetic dataset. Algorithm selection has been evaluated on various mechanisms as well as on an increasing number of features or meta-features, and therefore we study if additional information required for an Algorithm selector generalization ability can be generated using a synthetic data generator.

The results of this paper can be summarized as follows:

- The evaluation of synthetic data for Algorithm selection;
- An analysis of synthetic vs real-world data with respect to Algorithm selection accuracy;

- The determination of the relation of Algorithms' complementarity for Algorithm selection in classification problems.

This paper is organized as follows: in Section 2, an overview of the previous work is given; in Section 3, we describe the methodology used. In Section 4 we describe the experimental settings and in Section 5, we describe the experiments and the obtained results. Finally, in Section 6 we discuss the results and in Section 7 we conclude the paper.

2. Background

The authors of [8] examine the trade-off between per-instance Algorithm selection and running multiple Algorithms in parallel. While Algorithm selection leverages complementary performance among Algorithms, it requires complex models and additional computation. In contrast, parallel execution is feasible on modern hardware but can lead to resource contention and slowdowns. The results suggest that Algorithm selection is beneficial, particularly for large Algorithm portfolios.

The work [15] proposed an approach for decreasing the time needed for Algorithm selection training. Usually, the training process includes the evaluation of the performance of all pretrained Algorithms on the testing set. This process can take a lot of time. The authors propose to start by evaluating the small batches of testing data and then gradually increase the size of the evaluation data. During the process, Algorithms can be excluded from the pool based on the cross-validation between their learning curves. This allows for the early exclusion of under-performing Algorithms and can decrease the required training time by 20% to 50% without a noticeable downgrade in terms of accuracy.

From the current literature, it is seen that Algorithm selection can help in marginally improving the classification of many problems. Most applications achieve from 1%–5% improvement in relative classification accuracy [16]. Among many studies, the best results for simple Algorithm selection classifiers are often shown by Random Forest, and Support Vector Machines (SVM) classifiers. Boosted ensemble classifiers created with Adaboost also show good results and outperform single classifiers in some cases.

Authors of [17] trained a range of content-similarity-assessing Algorithms and used a Random Forest classifier to chose the best one for a particular document instance. The Random Forest classifier had 15% better accuracy than the random choice strategy for Algorithm selection. However, Random Forest-based Algorithm selection still had lower accuracy compared to the best of single Algorithms (0.51 vs. 0.58).

The authors of work [13] applied Algorithm selection for segmentation problems on Microsoft Common Objects in Context (MSCOCO) and Visual Object Classes Challenge 2012 (VOC2012) datasets without prior extraction of meta-features. For the VOC2012 dataset, the SVM Algorithm selector showed the best accuracy, while for the MSCOCO dataset, the AdaBoost ensemble showed the best result. The best overall result was achieved using an ensemble of five predetermined Algorithms resulting in 3% higher accuracy than the best single Algorithm. In another work, authors of [10] applied a per-object Algorithm selector by dividing the image into parts and using Algorithm selection for choosing the segmentation Algorithm for each part. This approach resulted in 2% improvement in accuracy using combinations of Artificial Neural Network (ANN) + Preference Rules and ANN + SVM classifiers.

A lot of work went into creating fully automated ML learning frameworks that can be unified under the AutoML term. The goal of AutoML systems is to provide users with a fully working and automated pipeline for analyzing the data, choosing the machine learning approach, configuring the Algorithm, and evaluating the results. SmartML [18] and AutoSklearn [19] are examples of such automated suites. SmartML, being the simpler one, executes an extensive grid search on the range of pretrained Algorithms as well

as a range of Algorithm selection classifiers. It also experiments with different training, validation, and test splits in the process. The AutoSklearn framework was created for the AutoML competition with the strict requirement of giving the best predictions in under 20 min. The authors achieved the best result by cutting out unpromising combinations for a grid search. They employed a resource-management strategy that devotes processing time and power to the most promising combinations based on pre-collected data.

Some works are concentrated on preemptively predicting the potential error rate for the classifier based on the uncertainty [20] and imbalance [21] of input data. The preemptive extraction and analysis of datasets' meta-features allow this approach to improve Algorithm selection recommendations. The extensive analysis of classification performance on different datasets revealed that in some cases the imbalance of the training dataset may improve the resulting classification accuracy.

The paper [22] analyzes the optimization dynamics of Generative Adversarial Network (GAN), focusing on gradient descent updates for both the generator and discriminator. The authors show that while GAN optimization is not a convex–concave game, equilibrium points in traditional GANs remain locally stable under certain conditions. In contrast, Wasserstein GANs (WGANs) can exhibit non-convergent limit cycles near equilibrium. To address these stability issues, the paper proposes a regularization term that improves local stability, accelerates convergence, and mitigates mode collapse.

The work [23] investigates the quality of conventional evaluation techniques for measuring GAN performance, such as the Inception Score and Fréchet Inception Distance. The authors propose two additional evaluation techniques. The proposed GAN-train technique evaluates the diversity and realism of a GAN's generated images by training a classifier on synthetic images and testing it on real ones, assessing how well the learned distribution matches the target distribution. In contrast, the proposed GAN-test technique measures the classifier's performance when trained on real images and tested on generated ones, capturing the fidelity of the synthetic data. The results demonstrate that these evaluation methods complement existing ones and, when used together, provide a more comprehensive assessment of GAN performance.

The study [24] explores the applicability of synthetic images from text-to-image models for image recognition tasks, focusing on zero-shot, few-shot learning, and large-scale model pretraining. The authors find that synthetic data significantly improve classification accuracy in zero-shot settings and remain beneficial in few-shot scenarios, though domain gaps pose challenges. They propose strategies like diversified text prompts and real-image-guided generation to enhance effectiveness. Additionally, synthetic data prove highly effective for model pretraining, sometimes surpassing ImageNet pretraining, particularly for architectures based on Vision Transformers. The study highlights both the potential and limitations of synthetic data, encouraging further research in this area.

The authors of [25] use hidden Markov models and regression models to generate synthetic time-series data based on a real smart-home dataset. As part of their research, they investigated a problem of generating the synthetic data from a small amount of real-world samples. Their results show that training Algorithms on a combination of a small real data sample and the synthetic data generated from it can achieve significantly higher activity recognition accuracy than training on the small real data sample alone.

Despite extensive research in both Algorithm selection and synthetic data generation, no prior work has explored the use of synthetically generated data for training Algorithm selectors. Algorithm selection has been widely studied across various domains, demonstrating improvements in classification accuracy and efficiency, while synthetic data generation has shown promise in augmenting training datasets, improving model performance, and reducing reliance on large real-world datasets. However, the potential of leveraging syn-

thetic data to enhance the training process of Algorithm selectors remains unexplored. This gap in the literature suggests an opportunity to investigate whether synthetic data can improve the generalization and robustness of Algorithm selection models, particularly in cases where real-world training data are scarce or expensive to obtain.

3. Methodology

Algorithm selection has been shown to have excellent results with meta-features [7,18,26] and some success with regular features [13,14,27]. In both cases, analyzing the whole dataset can provide valuable statistics-based meta-features, but it also requires having a lot of labeled data. In order to avoid this problem, we study the Algorithm selection training using synthetic dataset. The overall approach is shown in Figure 1. We focus on medical image classification. The task is defined as a mapping $C : I \rightarrow L$; for each image $i \in I$, determine the label $l \in L$.

The Algorithm selection approach improves performance by dynamically choosing the optimal Algorithm for each input instance [28,29]. Unlike ensembles that combine multiple Algorithm outputs, Algorithm selection chooses a single best Algorithm for each specific input, leveraging the complementary strengths of different Algorithms across the input space [7]. This meta-learning strategy works by training a selector that learns to map input features to the most suitable Algorithm from a portfolio of pretrained component Algorithms [30]. The selector essentially learns which Algorithm performs best on which types of inputs, allowing the system to achieve higher overall accuracy than any single Algorithm could achieve alone [8,9].

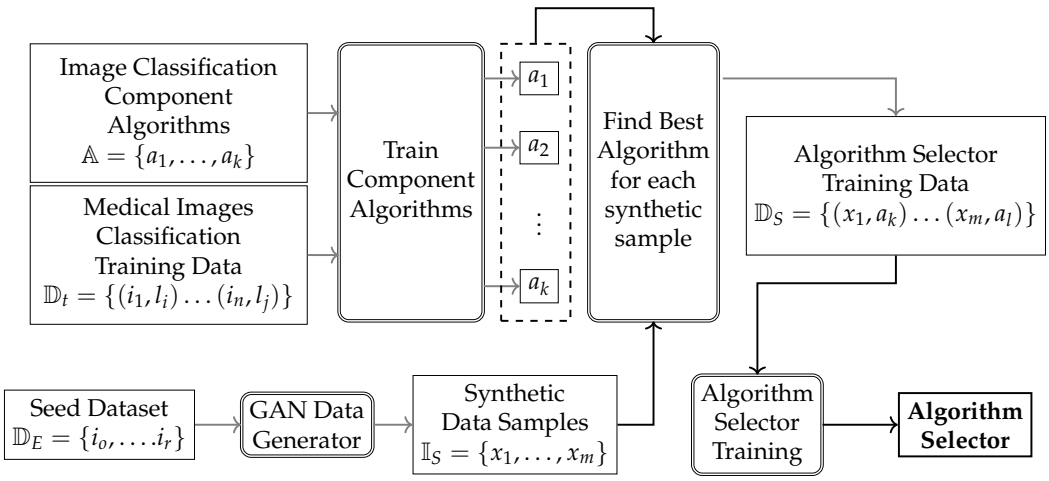


Figure 1. A schematic representation of the whole system.

The method starts with a set of pretrained Algorithms or by training a set of Algorithms $\mathbb{A} = \{a_1, \ldots, a_k\}$ for the classification task defined by mapping C. We prepare an initial set of data $\mathbb{D}$ by separating them into three subsets: $\mathbb{D}_t \cap \mathbb{D}_V \cap \mathbb{D}_T = \mathbb{D}$, training, validation, and testing, respectively. Using the training dataset $\mathbb{D}_t = \{(i_1, l_p) \ldots (i_n, l_q)\}$, we train a set of binary classifiers $\mathbb{A}$. Each Algorithm $a \in \mathbb{A}$ is evaluated on a validation dataset $\mathbb{D}_V$, and its accuracy $acc(a)$ is recorded. The Algorithms $a \in \mathbb{A}$ are all trained on the same training dataset $\mathbb{D}_t$ and will be referred to as component Algorithms.

Next, from the validation dataset $\mathbb{D}_V$, we use a subset of samples to create a subset, referred to as the seed dataset $\mathbb{D}_E \subset \mathbb{D}_V$. Using $\mathbb{D}_E$, we train a Generative Adversarial Network (GAN). Then, using the trained GAN, we create a new set of synthetic images $\mathbb{I}_S$. For each label in the dataset $\mathbb{D}$, we train a new GAN and generate a target number of synthetic images. For each sample of the resulting dataset $x \in \mathbb{I}_S$, we determine the best Algorithm. The best Algorithm is any Algorithm $a \in \mathcal{A}$ that classifies x correctly: $a(x) = l^t$, where l^t is the ground-truth label for the input image x. For synthetic images,

it is important to note that we generate them class by class. When training the GAN on images of a specific class, e.g., "pneumonia" or "healthy", the generated synthetic images inherit the same class label as their training data source. Thus, when we refer to an Algorithm classifying a synthetic image "correctly", we mean it assigns the image to the same class used to train the GAN that generated it. This approach provides unambiguous ground-truth labels for all synthetic images used in our experimental evaluation.

Using the synthetic images and the best Algorithms, we build the Algorithm selector training dataset $\mathbb{D}_S = \{(x_1, a_k), \ldots, (x_N, a_j)\}$. The inputs in $\mathbb{D}_S$ are the synthetic images, and the outputs are the Algorithm names represented by the label set $\mathbb{L}_S$. Using $\mathbb{D}_s$, we train the Algorithm selector $\mathcal{S}$ for the mapping $\mathbb{I}_s \rightarrow \mathbb{L}_s$. Once the Algorithm selector is trained, we evaluate its accuracy and compare it to the accuracy of the individual Algorithms using the $\mathbb{D}_T$ dataset.

The GAN used can be described as follows. Let $G : N \rightarrow S$ represent the mapping performed by the GAN, with $N : \mathcal{N}(0, 1)$ an input generated from a random-intensity pixels distribution and S being an output image. According to the original definition of GAN [31], the GAN plays a zero-sum game, where the generator is trained to make more and more realistic images and the discriminator is trained to differentiate between synthetic and real input images. Both the generator and the discriminator are Convolutional Neural Networks (CNN), in order to enable the generation and recognition of images.

The classification task studied in this paper $C : I \rightarrow L$ is a binary task: the target is to determine if an input image shows a patient with a diagnosis 'pneumonia' or a patient without one, 'healthy'. To assess the accuracy of the classification C, we use a direct accuracy measure. If the classification can be expressed as $a(i) = l$ with $a \in \mathbb{A}$ and $i \in I$ and $l \in L$, then the accuracy measure for a single Algorithm is shown Equation (1).

$$acc(a) = \frac{\sum_{h=1}^{n} a(i_h) == l_h}{n} \tag{1}$$

where $l_h \in L$ is the ground truth for the input image i_h and n is the number of samples in the dataset under evaluation. This accuracy is the ratio of correctly classified images over the total number of images under evaluation.

When evaluating the Algorithm selector approach, the Equation (1) is modified to take into account the Algorithm selection process. First, let $a^h \in \mathbb{A}$ be the Algorithm selected by the Algorithm selection for the input sample $i_h \in I$. Second, let the Algorithm selection be described by $s(i_h, \mathbb{A}) = a^h$. Finally, let the accuracy for the Algorithm selector can be written as in Equation (2).

$$acc(s) = \frac{\sum_{h=1}^{n} a^h(i_h) == l_h}{n} \tag{2}$$

Note that the mapping $S : \mathbb{I}_S \rightarrow \mathbb{L}_S$ is purposefully abstracted in order to allow the evaluation of the Algorithm selector using the image classification accuracy measure. This means that we do not directly measure the Algorithm selector accuracy with respect to the labels $\mathbb{L}_S$ but rather we measure the final classification accuracy over the set of image labels L by implicitly learning the labels $\mathbb{L}_S$ maximizing the classification accuracy. In general, we will refer to accuracy from Equations (1) and (2) by $\mathcal{A}$.

We use different Algorithms on the same dataset to obtain different Algorithm bias instead of a different data bias. Each Algorithm is trained according to its specific training requirements. In most of the cases, the classification Algorithms are CNN based.

Data Preparation and Construction

The data generation was primarily performed using a GAN network [31]. While we initially explored Stable Diffusion [32] as an alternative generative model, all the experi-

ments reported in this paper were conducted using GAN-generated images to maintain consistency in our comparative analysis. Our preliminary tests with Stable Diffusion suggested similar patterns to those observed with GANs, but a comprehensive comparison between different generative models for Algorithm selection remains an interesting direction for future work.

The specifics of the data generation methodology are as follows. Both parts of the GAN (a generator and a discriminator) are implemented as convolutional neural network. The schematic representation of the learning of the GAN network is shown in Figure 2. The general principle of learning GAN can be explained as follows. Propagate a latent vector z of randomly generated noise to obtain a set of synthetic samples that together with some real images create the discriminator training set. Then, train the discriminator by setting the Training Generator variable to 0 to generate an output from the discriminator and update its parameters (dashed-line Discriminator update). Repeat the process, but this time set the Training Generator variable to 1 to propagate the result of discriminator to the generator and update its parameters (dashed-line generator update). Repeating this process makes the discriminator converge to a high accuracy of discriminating between real and synthetic images and makes the generator converge towards generating synthetic samples very close to the real images.

The generator takes a latent vector z and maps it to an 8×8 feature space using a fully connected layer. It then applies four transposed convolution layers with a kernel size of 4×4 and a stride of 2, progressively upsampling to 128×128. Each layer uses LeakyReLU activation (a = 0.2), except for the final convolutional layer, which applies Tanh to produce an image.

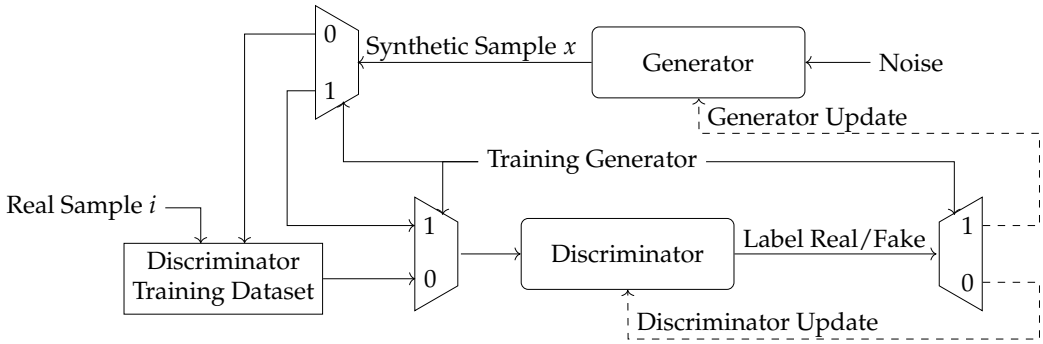


Figure 2. A schematic representation of the GAN learning.

The discriminator is a convolutional network that processes input images through five convolutional layers with a kernel size of 5×5, progressively downsampling the resolution from 128×128 to 8×8 with a stride of 2. Each layer uses LeakyReLU activation. The feature maps are then flattened, followed by a dropout layer (0.4) and a fully connected layer with a sigmoid activation for binary classification.

We use the Inception Score (IS) and the Fréchet Inception Distance (FID) to assess the quality and variety of generated images. The IS is calculated by a pretrained Inception V3 model applied to a sample of generated images. The FID is calculated by first extracting high-level features both from sets of real and generated images, then extracting the mean and covariance statistics of both samples and comparing them to each other.

4. Experimental Settings

4.1. Algorithms

To perform Algorithm selection on this task, we needed a pool of image classification Algorithms. We used the following CNN-based image classification neural networks from the PyTorch "Torchvision" and "Pretrained models for Pytorch" libraries: Dual-Path

Network (DPN), Residual Networks (ResNet), Densely Connected Convolutional Networks (DenseNet), Very Deep Convolutional Networks (VGG), and SqueezeNet. The summary of the Algorithms' main features is shown in Table 1. We used pretrained checkpoints for each Algorithm to lower the training time and data requirements. The pretrained checkpoints were obtained from the Torchvision library, which provides models trained on large-scale datasets such as ImageNet. We are using pretrained models to reduce the amount of data required for training, as the models already learned the way to effectively extract features by training on a vast corpus of images. Each model was initialized with these pretrained weights and subsequently fine-tuned on our dataset for several training epochs.

Table 1. A summary of the individual network models used for the classification task.

Algorithm Name	Specificity	Comments
DPN92 [33]	Combination of ResNet and DenseNet	Trained on ImageNet-1k in MXNet
ResNet50 [34]	Multiple blocks of convolution layers Average pooling, max pool	
DenseNet121 [35]		Global Average Pooling, Extracts features form all feature maps
VGG11 [36]	8 Convolution layers Max pool and three linear layers	
SqueezeNet [37]	All kernels are 1×1 size	Combination of ResNet and Reduced-size filters

Every component Algorithm is by default trained to up to five training epochs when not mentioned otherwise in the experiments. The main goal of this work is not to use the best-performing Algorithms but rather to evaluate Algorithm selection under specific training constraints. Therefore, the focus is more on how the Algorithm selector can improve or not through the use of synthetic data than on how the accuracy of processing improves through training. Therefore, the five epochs were decided as a reasonable amount of training for the dataset. In addition, the Algorithm's accuracy has enough variety, so that the complementarity can be observed. The Algorithms were trained to display a certain amount of complementary performance; that is, we are interested in Algorithms that, even if they have different classification accuracy, their respective true and false labels are not overlapping completely.

4.2. Algorithm Selectors

To train Algorithm selectors, we extracted feature representations from input images using a pretrained AlexNet [38] model. Specifically, we used the intermediate activations from five convolutional layers of the network as a compact image descriptor. This approach enables the capture of meaningful visual patterns at various levels of abstraction, while keeping the feature dimensionality within a practical range. All input images were resized to 128×128 before being passed through the feature-extraction pipeline. For each image, we computed the mean activation values across feature maps from the first, fourth, seventh, ninth, and eleventh convolutional layers. The resulting values were concatenated into a single feature vector. Each vector was then labeled with the identity of the best-performing classification Algorithm for the corresponding image, determined based on validation accuracy.

For the evaluation of the methodology, we used nine different shallow Algorithm selectors: ExtraTreesClassifier [39], SVM [40], DecisionTreeClassifier [41], KNeighborsClassifier [42], GradientBoostingClassifier [43], SGDClassifier [44], LogisticRegression [45],

RandomForestClassifier [46], and MLPClassifier [47]. We represent them as $m \in \mathbb{M}$, where $\mathbb{M} = \{m_1, \ldots, m_k\}$ is the set of all Algorithm selectors.

Each classifier is trained in an identical manner described specifically for each experiment and learning the mapping $\mathcal{M} : \mathbb{I}_S \rightarrow \mathbb{L}_S$. In this context, meta-learning refers to the process where the Algorithm selector learns to predict which component Algorithm will perform best for a given input. This approach falls under the broader meta-learning framework, where knowledge is transferred across learning episodes, allowing the system to learn from previous Algorithm performances to improve future selections. The Algorithm selector essentially operates at a meta level, learning patterns about when different Algorithms perform well rather than solving the original classification task directly. However, because the task at hand is classification, we have to prepare the training data in a specific manner. This is to address the problem of multiple correct answers during the learning of the selector. During the learning process of the selector, the dataset $\mathbb{D}_S$ samples x^i are to be mapped to a unique label l^i. However, due to complementary performance not being totally complementary, many samples x^i are such that they are correctly classified by more than one Algorithm, creating the mapping $\{x_i, [a_i \ldots a_j]\}$. Therefore, during the training process one has to decide which target Algorithm will be fed to the classifier as training sample.

We explore three possible strategies. The first strategy simply creates the sample by providing the target label from the classifier with the highest confidence score, referred to as $conf$. The second approach assign the target label randomly and is referred to as $rand$. The last strategy is basically to create as many samples as there are target labels and then include them into the final dataset with equal probability. This method is referred to as $copy$.

For the evaluation of the studied Algorithm selection, we compare it to two different approaches. First, we compare the meta-strategy to the most accurate component Algorithm (also referred to as the best Algorithm). Second, we compare the Algorithm selector trained on synthetic data to the Algorithm selector trained on real data. We will refer to Algorithm selector trained on synthetic data as SAS (standing for Synthetic data Algorithm Selection) and to the Algorithm selector trained on real data as RAS (standing for Real data Algorithm Selection).

4.3. Dataset

The component Algorithms were trained on the training part of the dataset $\mathbb{D}_t$. We used the chest X-RAY dataset taken from [48,49]. The original dataset contains 5232 training images (1349 healthy, 3883 pneumonia) and 624 test images (234 healthy and 390 pneumonia).

All images in the dataset were cropped to remove the markers that show the orientation of the X-ray capture. In [50], the authors demonstrated that that Algorithms can use these and other markers and focus on them to improve accuracy. This happens because different hospitals have different markers and different rates of sick vs. healthy individuals. X-ray images in which the subject had an unnatural posture (e.g., hunched or distorted) or had medical devices such as tubes attached were manually removed. These images were removed because unnatural posture or the presence of medical attachments often indicate a patient with a severe medical condition. Allowing Algorithms to rely on these indicators for classification could lead to biased results, shifting focus away from assessing the actual condition of the lungs.

The synthetic data are produced by training a GAN separately for each label in the dataset (healthy or pneumonia). To determine the quality of gan training convergence, we

use Inception Score [51] and Fréchet Inception Distance [52]. The data for training GAN were taken from the validation part of the dataset.

The procedure to prepare the different datasets from the original train and test sets is shown in Figure 3. All experiments were performed on balanced dataset. Therefore, we adjusted the amount of images with healthy and pneumonia labels in train and test parts of the dataset. The balanced test dataset is $\mathbb{D}_T$. The train dataset was then split into a new train $\mathbb{D}_t$ and validation $\mathbb{D}_V$ parts using 75% and 25% of the dataset, respectively. The seed dataset $\mathbb{D}_S$ is created on a case-by-case basis. When the size of $|\mathbb{D}_S| \leq |\mathbb{D}_V|$, all samples $i \in \mathbb{D}_E$ are drawn from $\mathbb{D}_V$. If the size is larger than the size of the validation dataset $|\mathbb{D}_S| \leq |\mathbb{D}_V|$, all additional samples are drawn randomly from $\mathbb{D}_t$. The $\mathbb{D}_E$ is used then as the training set for the synthetic data generator.

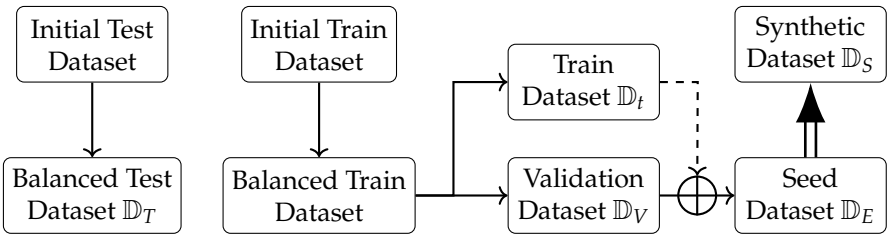


Figure 3. A schematic representation of the various data preparation steps.

The resulting balanced $\mathbb{D}_t$ dataset contains: 1800 training images (900 healthy, 900 pneumonia), the $\mathbb{D}_V$ dataset contains 540 validation images (270 healthy, 270 pneumonia), and the $\mathbb{D}_T$ dataset contains 456 test images (228 healthy, 228 pneumonia).

The component Algorithms are trained in general on $\mathbb{D}_t$ unless specified. The Algorithm selector is trained on the $\mathbb{D}_S$. The accuracy $\mathcal{A}$ is always assessed on the test part of the dataset $\mathbb{D}_T$ unless specified differently. The seed dataset $\mathbb{D}_E$ is created as a subset of the original validation dataset (containing only real images) such that $\mathbb{D}_E \subseteq \mathbb{D}_V$. The training dataset $\mathbb{D}_S$ is constructed from either purely synthetic or purely real images.

5. Experiments

5.1. Component Algorithm Accuracy

The first set of experiments aims to verify the accuracy of the individual Algorithms. Table 2 displays how accurate each Algorithm is for classifying both the validation dataset $\mathbb{D}_V$ and the test set $\mathbb{D}_T$. The first column shows the Algorithm name, and the second column demonstrates the validation accuracy on the validation dataset of real images $\mathbb{D}_V$, and the third column shows the accuracy on the test dataset $\mathbb{D}_T$. The last column shows the accuracy of the validation dataset, which contains only synthetic images $\mathbb{D}_S$. All the Algorithms were trained for five epochs on the train dataset $\mathbb{D}_t$.

In each column of Table 2, the results are ranked in descending order, from the most to the least accurate. In columns three and five the differences in relative ranks of each Algorithm on a given dataset are shown. For instance, the DPNWrapper obtained the highest accuracy on the validation dataset $\mathbb{D}_V$ (column two, row two) but on the test set $\mathbb{D}_T$, the same Algorithm's accuracy is one before the last one (column four), and thus the difference in rank, given the order introduced above, is 3. A rank of 0 means that the algorithm's classification accuracy does not change from one dataset to another with respect to the other algorithm's classification accuracy.

Table 2. Accuracy of individual Algorithms (trained for five epochs).

Model Name	$\mathcal{A}$ **on** $\mathbb{D}_V$	Relative $\mathbb{D}_V \setminus \mathbb{D}_T$	$\mathcal{A}$ **on** $\mathbb{D}_T$	Relative $\mathbb{D}_T \setminus \mathbb{D}_S$	$\mathcal{A}$ **on** $\mathbb{D}_S$
SqueezeNet	89.26	0	82.89	1	63.30
DPNWrapper	96.11	3	89.04	2	66.60
DenseNet	94.81	2	90.79	0	87.409
VGG	94.26	2	89.91	3	50.10
ResNet	95.19	1	89.47	0	65.80

The reason for these experiments is not only to determine the individual Algorithms' accuracy but also to demonstrate the fact that the accuracy of classification on the $\mathbb{D}_V$ dataset is not a very reliable predictor for the accuracy obtained on the $\mathbb{D}_T$ dataset. Even less, the accuracy from the validation dataset $\mathbb{D}_V$ is not a very good predictor of the accuracy on the synthetic dataset $\mathbb{D}_S$. However, the accuracy on the synthetic dataset $\mathbb{D}_S$ is a better predictor for the accuracy on the test dataset $\mathbb{D}_T$. Therefore, it should be possible to leverage this specific relation between the synthetic dataset and the test dataset to the Algorithm selection by training it on purely synthetic data.

There are three main detailed observations from this experiment. First, note that when evaluated on $\mathbb{D}_S$ that contains only synthetic images, only one Algorithm, the DenseNet, results in accuracy higher than 86%. The other Algorithms perform quite poorly, with VGG scoring nearly random result with accuracy 50.16%. The second observation is that on both the validation $\mathbb{D}_V$ and test datasets $\mathbb{D}_T$, all five Algorithms provide comparable accuracy ≈90% with only SqueezeNet evaluating at ≈83% on the test dataset $\mathbb{D}_T$. The final observation is the discrepancy in Algorithm accuracy between the $\mathbb{D}_V$ and $\mathbb{D}_T$. The Table 2 has been colorized to indicate the relative accuracy of the individual Algorithms in descending order, starting from the highest by the following colors: red, green, blue, orange, and yellow. Observe that the evaluation on the $\mathbb{D}_V$ results in a relatively poor generalization. If one would look for most efficient Algorithm based on the validation dataset, then both the $\mathbb{D}_S$ and $\mathbb{D}_T$ would result in poorer accuracy.

To further analyze the performance of the individual Algorithms, Figure 4 shows the complementarity of the five used Algorithms' accuracy displayed as the correlation matrix. Each cell in Figure 4 shows the relative overlap between the result of classification between two Algorithms. For instance, the correlation between SqueezeNet and DPN92 is 0.7636. This scores means that 76.36% of their results are the same for both positive and negative labels. The notably lower accuracy on $\mathbb{D}_S$ compared to real data is expected and informative for our study. This performance gap indicates that synthetic images capture class-specific features while representing a different distribution than real images. This domain shift helps identify which Algorithms are more robust to input variations, which is valuable information for the Algorithm selector to learn. Rather than indicating poor GAN training, this controlled distribution difference creates a more challenging testbed that better differentiates the Algorithms' complementary behaviors. The results of this correlation analysis show that the selected Algorithms have enough complementarity to be used as component Algorithms in our Algorithm selection methodology.

It is worth noting that while more sophisticated generative models such as conditional GANs could explicitly incorporate class-label information into the generation process, we opted for a simpler approach where we train separate GAN models for each class. This class-by-class training implicitly conditions the generation process on the class label without requiring modifications to the GAN architecture. Our approach allows us to control the class distribution in the generated dataset while maintaining architectural simplicity. Future work could explore whether conditional GANs or other label-aware generative

models might produce higher-quality synthetic images that further improve Algorithm selection performance.

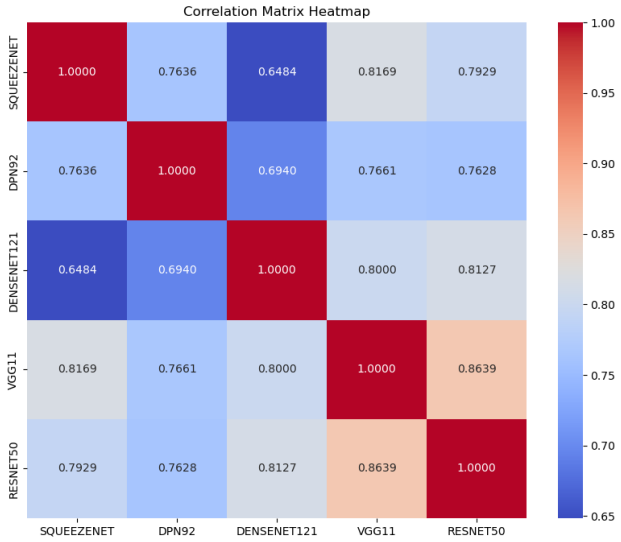
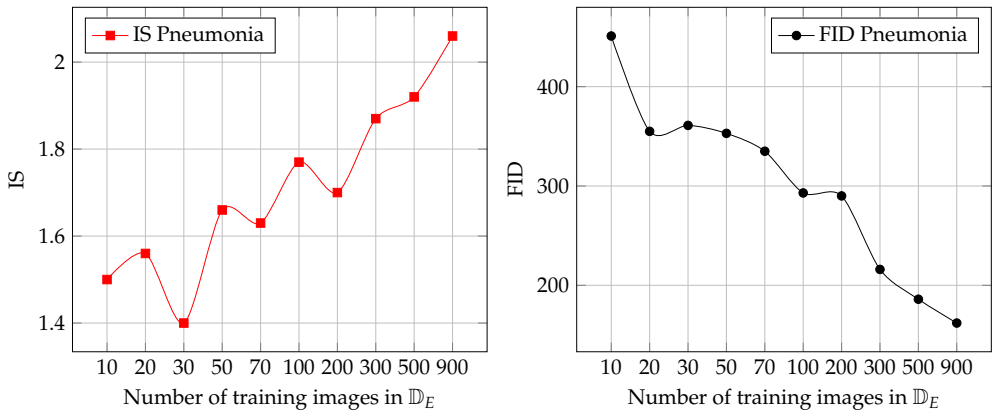


Figure 4. The correlation between the five individual Algorithms trained for five epochs and evaluated on the $\mathbb{D}_T$ dataset.

5.2. Synthetic Data Generators Assessment

Before starting the Algorithm selection evaluation, we evaluate the performance of the data generator networks. Figure 5a,b show two common measures used to asses the quality of generated images. Figure 5a shows the Inception Score (IS) measure [53] applied to the pneumonia synthetic images generated by the used GAN. The IS metric is used to assess how sharp and distinct the generated images are. IS is calculated using the Inception v3 [54] pretrained image classification model applied to images generated by GAN. The score is maximized when two conditions are met. The first condition is high confidence in the label assigned to each image, indicating image sharpness. The second condition is the variety of the labels assigned to different images, indicating a high diversity of the generated images. Figure 5b shows the FID score. The FID score [52] is a measure of the distribution of the synthetic images when compared to the distribution of the real images. In particular, the covariance and the mean of the synthetic distribution of images are compared with the distribution parameters of the real images.



(**a**) The IS score of the used GAN (**b**) The FID score of the used GAN

Figure 5. Graphs of FID and IS metrics for GAN generated synthetic images based on the number of images used for training.

Observe that both measures confirm that at least statistically the evaluated GAN converges properly or as expected. In other words, as more seed images are used to train the GAN, the generated samples more closely follow the original distribution of the real images, and the synthetic images become increasingly similar to the seed images. Note that we are interested not in generating synthetic samples highly similar to the distribution of real images, but rather we aim to capture the noise required for efficient generalization. While the FID score near 0 indicates the highest similarity to the original real-world images, it also implies that for in order to capture the noise beneficial for Algorithm selection, fewer images in the seed dataset $\mathbb{D}_E$ are desired.

Finally, we also evaluated the quality of synthetic images in a classification setting. For this purpose, we evaluated the accuracy of the five component Algorithms that were trained on images from $\mathbb{D}_t$ for 5 epochs on the $\mathbb{D}_S$ data generated by the GAN from various sizes of the $\mathbb{D}_E$ dataset. Figure 6 shows the individual Algorithm accuracy as a function of the size of the $\mathbb{D}_E$ dataset.

Figure 6 shows that when the GAN is seeded with over 300 real images, the resulting synthetic dataset provides enough information to result in an accuracy over 80% for at least certain Algorithm. When seeded with 900 images, the accuracy of all Algorithms is larger than 80%. This means that the larger the seed dataset is, the greater the ability of the GAN to generate images similar to the seed dataset.

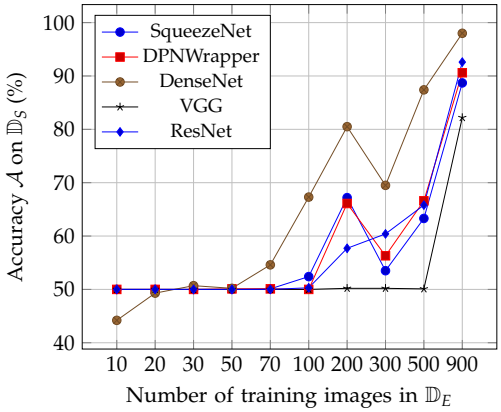


Figure 6. The classification accuracy of individual Algorithms (trained for five epochs) on synthetic data as a function of amount of training images used for GAN training.

This indicates that when the GAN is provided with too many images, its statistical output distribution converges to the original distribution of the seed dataset. While this result is expected [31], in this work we are looking for the complementarity of the Algorithm selection performance to the performance of the individual Algorithms. This means that we are looking for a complementary distribution of data samples that would allow us to predict the Algorithm accuracy $\mathcal{A}$ on $\mathbb{D}_T$ from $\mathbb{D}_E$ using the smallest possible set of seed samples.

Figure 7 shows images both real as well as synthetic ones. Figure 7a,b show examples of healthy and diagnosed images from the original dataset. Figure 7c,d show examples of synthetic images representing healthy individuals. Figure 7c show the generated images for the highest FID score, while the Figure 7d shows generated images for the best IS score. Figure 7e,f show generated images for diagnosed individuals for the highest FID score and for the best IS score, respectively. Note as a general observation, when simply comparing the real images from Figure 7b to the synthetic images from Figure 7c,e, the generated images are more noisy and in general appear as an extrapolation of the real-world images.

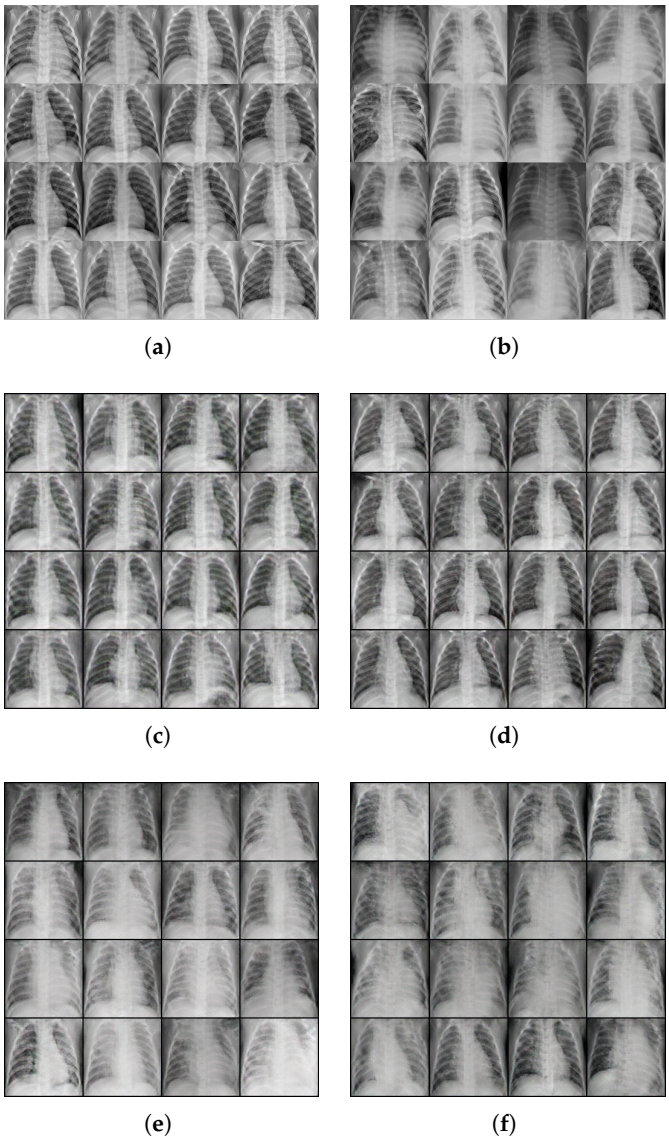


Figure 7. Example images used for our experiments. (**a**,**b**) real images; (**c**,**d**) synthetic images for healthy subjects; and (**e**,**f**) synthetic images for patients with pneumonia. (**a**) Example of real X-rays of healthy subjects; (**b**) Example of real X-rays of patients with pneumonia; (**c**) example of synthetic X-rays for healthy subjects (highest FID score); (**d**) example of synthetic X-rays for healthy subjects (highest IS score); (**e**) example of synthetic X-rays for patients with pneumonia (highest FID score); (**f**) example of synthetic X-rays for patients with pneumonia (highest IS score).

5.3. Algorithm Selection as a Function of Algorithm Selectors

The next set of experiments shows the performance of the individual Algorithm selectors. Figure 8 shows the accuracy of classification when the Algorithm selector is trained on synthetic samples from $\mathbb{D}_S$ generated by GAN from an increasing number of seed samples $\mathbb{D}_E$ which are selected from $\mathbb{D}_V$ and $\mathbb{D}_t$.

As shown in Figure 3, the training dataset used here is constructed in a similar fashion of constructing the sample dataset $\mathbb{D}_E$. This means that only the first samples from $\mathbb{D}_V$ are used, and when their number is larger than $|\mathbb{D}_V|$, samples from $\mathbb{D}_t$ are used. However, all seed images are real and all Algorithm selectors are trained on synthetic images generated by the GAN network.

Figure 8 shows that for each size of seed dataset $\mathbb{D}_E$, different Algorithm selectors provides the highest classification accuracy. Because of this observation, we will further

report only the best performance for each experiment by selecting the most accurate Algorithm selector for each task.

Note that as expected, the more images are used to generate the $\mathbb{D}_S$ dataset, the higher the Algorithm selection accuracy is in average. Interestingly, the variation is lowest for $70 \leq |\mathbb{D}_E| \leq 300$. The implication of this observation is that for very few samples in $\mathbb{D}_E$, GAN-generated synthetic images are too noisy, while for a very large amount in $\mathbb{D}_E$, the generated images do not provide enough variation for all the Algorithms used by the algorithm selector, and thus resulting in almost no classification accuracy increase. However, an interesting observation is that the variation between the classification accuracy of the different Algorithm selectors is also very small when only 10 images are used to seed the GAN generator.

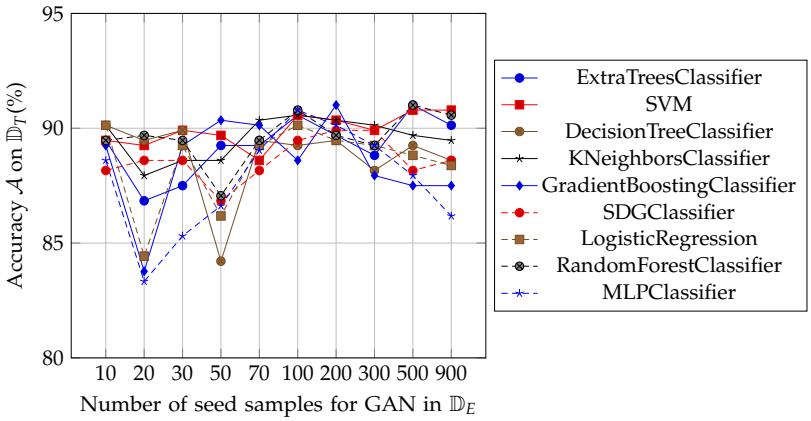


Figure 8. The average accuracy of Algorithm selectors on the classification task as a function of number of seed samples of GAN.

Figure 9 summarizes these results. The plots show the average (Avg.) and most accurate (Best) Algorithm selection accuracies. The average accuracy represents the average over the set of all nine Algorithm selectors. The average is computed over the set of Algorithm selectors trained on the same set of training samples and plots them against the most accurate Algorithm selector for both RAS and SAS approaches. The Algorithm selectors were trained on increasing number of real or synthetic images and the most accurate result for each type of learning is displayed. The classification accuracy was evaluated on 228 real images per class. Note that Algorithm selection trained on synthetic images outperforms or equals Algorithm selection trained on real images on five out of six data points (red line).

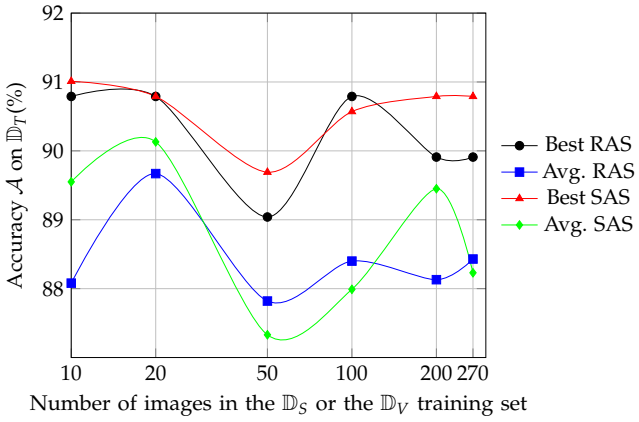


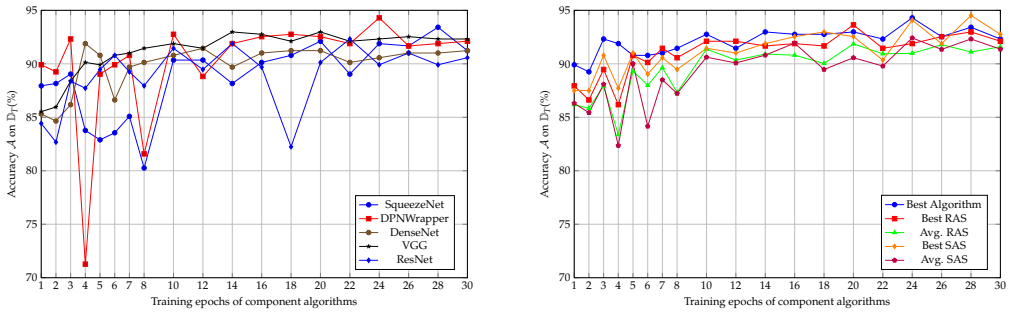
Figure 9. Processing accuracy on 200 real images for best and average Algorithm selection trained on real and synthetic images.

Another observation is that the Algorithm selection provides the highest accuracy when the Algorithm selection provides the highest degree of noise. Here, it means that the Algorithm selection is trained on smallest number of samples (10 in Figure 9) in $\mathbb{D}_S$.

5.4. Algorithm Selection as a Function of Component Algorithms Training Epochs

The next set of experiments evaluates Algorithm selection accuracy as a function of the number of training epochs of the component Algorithms. The Algorithm selectors were trained on either real and synthetic images, 270 real and 270 synthetic, respectively. Figure 10 shows the results of both the evaluation of the accuracy of the component Algorithms as well as the accuracy of Algorithm selection as a function of component Algorithms' training epochs. Figure 10a shows the accuracy of the individual component Algorithms. Figure 10b also shows the the accuracy of Algorithm selection when using images $\mathbb{D}_S$ generated by GAN (in the case of SAS) or real images $\mathbb{D}_V$ (in the case of RAS) for the training of Algorithm selectors. In Figure 10b, we consider for each epoch the most accurate Algorithm selection.

In Figure 10a, the more the Algorithms are specifically trained to the dataset, the more they converge to individual accuracies with reduced variation. It can be concluded that that the component Algorithms' complementary performance becomes less relevant due to Algorithm relative overfitting or training. By relative overfitting, we mean that an Algorithm's accuracy $\mathcal{A}$ tends to a much lower variance by matching the training dataset more and more precisely. In other words, as the performance overlap becomes larger, the Algorithm selection converges to select Algorithms with diminishing effect of improvement. Note that when component Algorithms are trained for one epoch, the difference in the average accuracy is larger than 7%, while, when trained for 30 epochs, their relative accuracy difference is <3%. In addition, their relative accuracy increases from the interval [84%–90%] at epoch 1 to [91%–93%] at 30 epochs. In addition, note that the highest and lowest accuracy of classification was achieved for 24 training epochs and for 4 training epochs by the DPN.



(**a**) (**b**)

Figure 10. Algorithm selection and component Algorithms' accuracy $\mathcal{A}$ as a function of training epoch number of the component Algorithms. (**a**) Average Classification accuracy of Component Algorithms' $\mathcal{A}$ as a function of training epochs; (**b**) Average Classification accuracy $\mathcal{A}$ when using the Algorithm selection as a function of training epochs.

Figure 10b displays a similar trend—it shows the accuracy of the best component Algorithm as well as the accuracy of Algorithm selection trained on real and on synthetic images. Similarly to Figure 10a, at epoch 1, the difference of accuracies is in interval [86%–90%], while at training epoch 30%, the interval is [91%–93%]. The best Algorithm from all five component Algorithms is taken for each epoch. First, observe that both the average synthetic and real Algorithm selection accuracy always under perform when compared to the best Algorithm.

The best Algorithm accuracy (blue curve) shows the highest value for eleven data points. The SAS (orange line) shows the highest value for four data points. The RAS (red line) shows the highest value for three data points. The best Algorithm is more accurate for most of the training at lower epoch numbers. These epochs are [1–4,5,8–10,14–16,22]. The Algorithm selection, on the other hand, is more accurate for most of the experiments with a higher number of training epochs; in particular, for epochs [5,7,12,20,24–30].

Interestingly, while the highest Algorithms' complementarity occurs in the early training epochs (due to highly randomized nature of under-trained Algorithms), the highest accuracy in the early training epochs is dominated by the best Algorithm (Figure 10b). This would imply that for Algorithms trained for few epochs only, the predictability of their behavior might be too difficult for the Algorithm selector despite their high complementary performance. It is only once the Algorithms are predictable enough that Algorithm selection improves the overall accuracy. This would confirm the fact that while Algorithms are not well converged, predicting their classification is difficult due to largely randomized nature.

5.5. Algorithm Selection as a Function of Number of Real Images Used to Train the Generator

The next set of experiments is conducted in order to understand what amount of synthetic samples is necessary for an Algorithm selector to improve over the best Algorithm. First, three types of Algorithm selection strategies are evaluated in this experiment: $conf$, $rand$, and $copy$ (as previously described in Section 4.2), along with two types of training datasets. In these experiments, the Algorithm selector was either trained on 270 real images or 500 synthetic images. The results are shown in Figure 11.

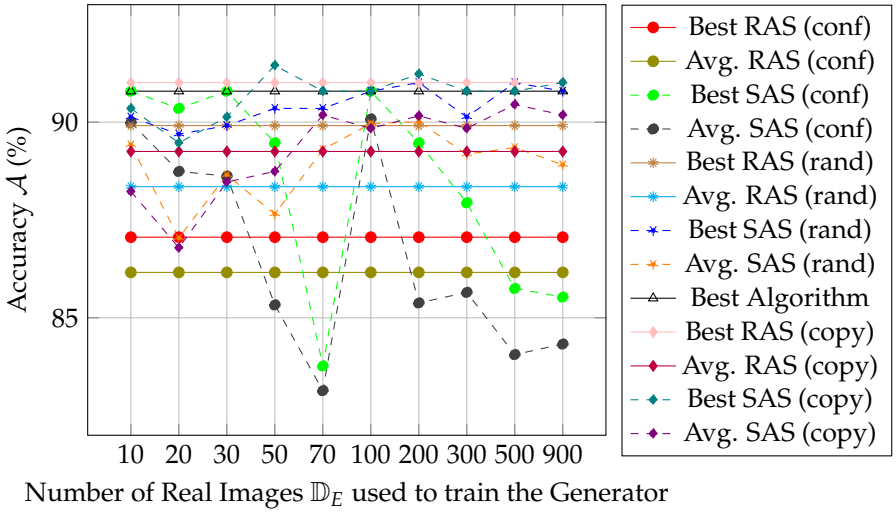


Figure 11. Algorithm selection accuracy as a function of real images used to train the image generator.

The first observation is that, when using real data for training, all three Algorithm selection $rand$, $conf$, and $copy$ result in constant accuracy. This is natural because the Algorithm selection is always trained on the same 270 real images (solid plot lines). The synthetic images are, however, generated for each different number of seed samples from $\mathbb{D}_E$, and therefore the Algorithm selector accuracy varies (dashed plot lines).

The second observation is the evolution of Algorithm selector accuracy in the classification task. First, observe that from the Algorithm selection methods trained on real data, the $copy$ is the most accurate, followed by $rand$, and the least accurate is $conf$. Second, for 50, 200 and 900 seed images, the $copy$ Algorithm selection is the most accurate, while for 500 seed images, the $rand$ Algorithm selection is the most accurate. Finally, observe that when the number of training images grows to over 100, the $copy$ and $rand$ Algorithm selections are much more accurate than the $conf$ Algorithm selection.

The last observation is that with the *copy* strategy, both Best RAS and Best SAS obtain the highest accuracy. Only the SAS *copy* Algorithm selection strategy outperforms the best Algorithm (best Algorithm accuracy is shows as black line with triangles as data points marker) as well as the Best RAS *copy* strategy in some cases.

5.6. SAS vs. RAS Under Limited Data Availability

Another set of experiments was conducted to compare the performance of RAS and SAS under conditions of limited data availability. In these experiments, we used the same five Algorithms, each fine-tuned on data from $\mathbb{D}_t$ for five epochs. The amount of validation data from $\mathbb{D}_V$ used in each run was reduced to 50 images per class. We performed 10 runs of the experiment for both RAS and SAS. The results are presented in Table 3. For each RAS run, eight Algorithm selectors were trained on 50 randomly sampled images from $\mathbb{D}_V$. For each SAS run, we used the same 50 images selected for RAS as the seed dataset $\mathbb{D}_E$ to train a GAN. This ensured that neither RAS nor SAS had an advantage in data sampling. For each run, the GAN was trained independently with a distinct fixed random seed. For five of the runs, we used the Inception Score (IS) to select the generator training epoch for data generation, and for the other five runs, we used the Fréchet Inception Distance (FID). The results show that the difference between using IS and FID for generator epoch selection was not statistically significant for this set of experiments. Using the selected generator epoch, 1000 synthetic images per class were generated for each run. The Algorithm selectors for SAS were trained on the generated synthetic images, and the final classification accuracy after Algorithm selection was recorded. The "Best Classifier" row reports the average result of the best-performing classifier from each run. The "Average" row shows the mean accuracy of all eight Algorithm selectors across all runs. For each Algorithm selector, we report the mean accuracy along with the 95% confidence interval computed across 10 runs.

Table 3. Performance metrics of Algorithm selectors evaluated over 10 different seeds.

Algorithm Selector	Avg. RAS	Avg. SAS	Δ (SAS − RAS)	p (*t*-Test)	p (Wilcox)
ExtraTreesClassifier	88.38 ± 1.18	89.43 ± 0.96	1.054	0.0613	0.0273
SVM	89.39 ± 0.53	90.77 ± 0.05	1.378	0.0004	0.0139
DecisionTreeClassifier	90.70 ± 0.20	88.77 ± 0.89	−1.930	0.0003	0.0020
KNeighborsClassifier	90.88 ± 0.13	90.66 ± 0.42	−0.220	0.2729	0.1394
SDGClassifier	88.33 ± 0.75	89.10 ± 0.82	0.767	0.1545	0.1394
LogisticRegression	87.67 ± 0.82	89.25 ± 0.76	1.580	0.0030	0.0139
RandomForestClassifier	89.65 ± 0.32	90.35 ± 0.52	0.703	0.0020	0.0273
MLPClassifier	88.07 ± 2.23	89.85 ± 0.56	1.775	0.1530	0.0273
Best Classifier	90.92 ± 0.13	90.99 ± 0.25	0.066	0.6275	0.1094
Average	89.13 ± 0.17	89.77 ± 0.27	0.639	0.0028	0.0020

We performed paired *t*-tests between RAS and SAS results for each Algorithm selector. A p-value less than 0.05 indicates a statistically significant difference between RAS and SAS. A p-value greater than 0.05 suggests that the difference is not statistically significant. Some Algorithm selectors performed very consistently despite being trained on different data with varying random seeds. However, the distribution of test results violated the normality assumption required for the paired *t*-test for several models: SVM, KNeighborsClassifier, SGDClassifier, LogisticRegression, RandomForestClassifier, MLPClassifier, and Best Classifier. Therefore, we also performed the Wilcoxon signed-rank test as a non-parametric alternative to paired *t*-tests. The results showed a statistically significant improvement for SVM, LogisticRegression, RandomForestClassifier, and MLPClassifier when trained on synthetic data compared to real data. Conversely, DecisionTreeClassifier showed a statistically significant decrease in performance under the same condition. KNeighborsClassifier

achieved the highest average performance across 10 runs and did not show a statistically significant difference in results between training on synthetic versus real data.

5.7. Algorithm Selection as a Function of Component Algorithms

The final set of experiments illustrates the impact of the individual Algorithms used for the target task on the Algorithm selection accuracy. For this experiment, we consider the removal of the individual Algorithms from the set $\mathbb{A}$. The purpose of this experiment is to determine how the set of component Algorithms affects the Algorithm selection accuracy.

Figure 12 shows the results of the Algorithm selection as a function of components of the set $\mathbb{A}$. The x axis shows which Algorithms have been removed from the initial $\mathbb{A}$, while the y axis shows the accuracy $\mathcal{A}$. Both Algorithm selection trained on synthetic and on real data are illustrated. The first five data points (x axis) show the results of classification when a single Algorithm is removed from $\mathbb{A}$ and the last two data points show the removal of two and three Algorithms from $\mathbb{A}$, respectively.

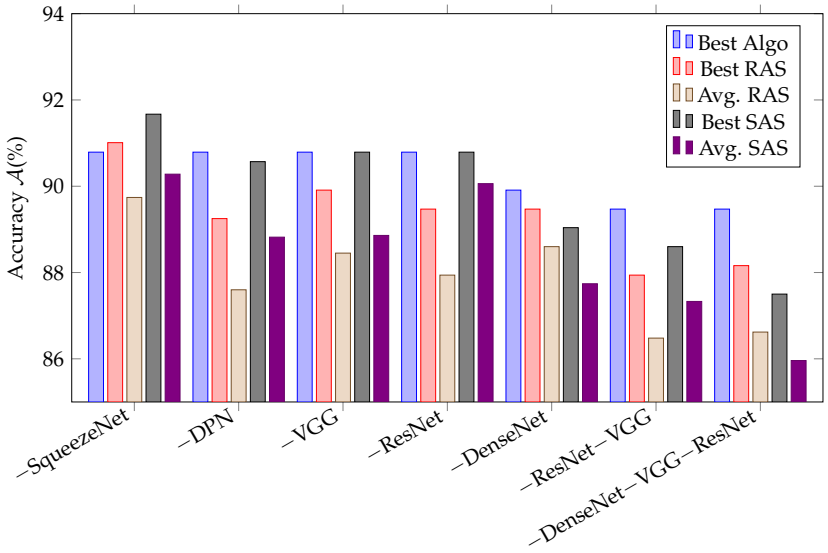


Figure 12. Algorithm selection results obtained by excluding some of the component Algorithms.

The first interesting observation is that the classification accuracy when using Algorithm selection is strongly improved only when SqueezeNet is removed. Interestingly, SqueezeNet is the least accurate Algorithm on the training and testing dataset and is second to last on the $\mathcal{S}_S$ in Table 1. This indicates that SqueezeNet has either low predictability due to its lower evaluation accuracy, or that having SqueezeNet in the learning dataset alters the Algorithm selection data bias in such a way that the Algorithm selector increases the amount of false selections.

Second, similarly to previous experiments, the synthetic Algorithm selection here also almost always outperforms the real Algorithm selection. There are only two instances where real Algorithm selection outperformed synthetic Algorithm selection. These two cases are the DenseNet and the DenseNet-VGG-ResNet removal. However, in both cases, the best Algorithm remained more accurate than the Algorithm selection approach.

6. Results and Discussion

The results of the performed experiments can be summarized from several points of view. First, the Algorithm selection has experimentally demonstrated that it can improve the accuracy when trained on well-behaved Algorithms. Well-behaved Algorithms are such that the difference between the selection confidence and the result of processing result tends to 0. This Algorithmic good behavior does, however, go against the idea of improvement by

generalization by modeling processing noise. The reason is because, in order to effectively select Algorithms for the classification task, the prediction must take into account the outliers of each Algorithm. In general, such outliers are not directly predictable, as they are outside of the main distribution of the Algorithm's input–output mapping. Therefore, training an Algorithm selection on well-behaved Algorithms can result in selecting Algorithms in such a way that does not allow for proper selection for the sample outliers. Both of these effects can be seen on Figure 10b. The best Algorithm selection accuracy is obtained at epochs 24, 28, and 30, pointing towards the fact that the Algorithm selector improves the predictive accuracy with increasing epochs of training of the component Algorithms. However, at the same time, for epochs 14, 16, and 22, the Algorithm selection does not provide any improvement when compared to the most accurate Algorithm. This would indicate that the ability to accurately select an Algorithm would benefit from directly predicting each Algorithm's outliers. Regarding the component Algorithms' training duration, while five epochs might seem limited for deep learning models, our results demonstrate this was sufficient for our experimental purposes. As seen in Figure 8, these pre-trained models achieved reasonably stable performance within this timeframe on our medical imaging dataset. In fact, the early training epochs created opportunities for complementary performance patterns that were valuable for Algorithm selection. As models train longer and approach convergence, they tend to find similar solutions, potentially reducing the benefits of Algorithm selection. Our approach intentionally balanced adequate individual performance with maintaining Algorithmic diversity, which aligns with our research focus on Algorithm selection rather than maximizing individual Algorithm accuracy.

It is worth noting that our component Algorithms were deliberately chosen to represent a diverse range of architectures commonly used in medical image classification. While more advanced state-of-the-art architectures might achieve higher absolute accuracy, our focus was on demonstrating the efficacy of Algorithm selection using synthetic data rather than maximizing classification performance per se. The moderate accuracy levels (<95%) of our component Algorithms actually provide a realistic scenario where Algorithm selection can make meaningful improvements. In settings where all Algorithms achieve near-perfect accuracy, the potential benefits of selection would be minimal. Future work could explore how Algorithm selection using synthetic data performs with more advanced architectures like Vision Transformers [55], MedicalNet [56], or other domain-specific models, though we expect the core findings about synthetic data's utility for Algorithm selection to remain valid regardless of the specific architectures employed.

Second, Algorithm selection experimentally benefits from synthetic data. In particular, when using the synthetic images generated from certain number of seed samples, the SAS outperforms RAS and most of the experimental cases. As shown in experiments illustrated for instance in Figures 8 and 9, the improvement provided by Algorithm selection is well observed for Algorithm selectors trained on few images, implying that even with very partial knowledge about the well-behaved Algorithms, the noisy samples from the GAN generator provide enough information to obtain an accuracy $\mathcal{A}$ at least as good as the best Algorithm. It also shows that for a small size of $\mathbb{D}_E$, the GAN generates images that have enough information to model the sample distribution from the dataset (Figure 8). Similarly, this is reflected by the fact that for few synthetic images, the SAS approach results in highest accuracy $\mathcal{A}$ (Figure 9).

Third, training the *rand* or the *copy* Algorithm selector experimentally, shows that it negates the Algorithm selection bias. This can be observed in Figure 11, where not only do the *rand* and *copy* SAS improve other instances of Algorithm selection training, but the best Algorithm is actually the *copy* RAS Algorithm. Note that the sample replication is not a novel method and it is not Algorithm selection specific. The unique position of our

approach is that we specifically target synthetic images. In addition, one could ask why we need the GAN generator if we potentially have enough images to train a shallow classifier. As shown in Figures 8 and 9, even if the maximum of 540 (270×2) is used for training, the resulting Algorithm selection is statistically less accurate than the one that is trained purely on synthetic data.

Finally, we also verified the statistical significance of the experimental setup. Interestingly, while the statistical verification demonstrated that the synthetic images are a viable option as a tool to efficiently select Algorithm, it also showed that the improvement is also model dependent. Four out of nine classifiers had constantly improved performance when using the synthetic data, while additional two showed statistical improvement when using the Wilcox test only. This would also imply that this result can be also task specific. Therefore we consider this as a motivation for a direct extension of this work into other domain-specific works.

7. Conclusions

Under limited data conditions, our experiments demonstrated that synthetic data generated by GANs can effectively support Algorithm selection. In several cases, SAS not only matched but statistically outperformed RAS in final classification accuracy, with significant gains observed for SVM, LogisticRegression, RandomForestClassifier, and MLPClassifier. Importantly, these improvements were consistent across multiple seeds used for GAN and Algorithm selector training. Although DecisionTreeClassifier showed a performance drop when trained on synthetic data, and KNeighborsClassifier exhibited no significant difference, the overall trend supports the viability of synthetic data for Algorithm selection in low-resource settings. The synthetic images we create help us make medical image classification more accurate by using Algorithm selection, achieving better results than just using the best Algorithm or the Algorithm selection trained on the original dataset. While our study primarily used traditional machine learning Algorithms as selectors due to their interpretability and efficiency with limited training data, future work could explore deep neural networks as Algorithm selectors. Deep learning-based selectors might capture more complex patterns in the data that determine Algorithm effectiveness, potentially further improving selection accuracy. However, such approaches would likely require larger training datasets and might introduce additional computational overhead during both training and inference. The trade-off between selector complexity and performance gains represents an interesting direction for future research, particularly in the context of synthetic training data, where generating larger volumes of data is feasible.

One limitation of our current study is that we investigated Algorithm selection trained on either purely real or purely synthetic data, but did not explore mixed datasets combining both types. Future work could examine how varying the proportion of real to synthetic images in the training data affects Algorithm selection accuracy. Such an investigation might reveal an optimal mixing ratio that leverages the complementary information in both real and synthetic samples. This approach could potentially further improve Algorithm selection performance while minimizing the amount of real data required. Additionally, it would be valuable to study whether certain types of real samples are more informative when combined with synthetic data, which could lead to more efficient strategies for creating hybrid training datasets.

Author Contributions: Conceptualization, M.L. and M.K.; Methodology, F.M. and M.K.; Software, M.Z.; Validation, B.T. and S.N.; Investigation, M.Z. and M.L.; Data curation, M.Z.; Supervision, B.T. and M.L.; Writing—original draft, M.Z. and M.L.; Writing—review and editing, B.T., F.M., M.K. and S.N.; Visualization, M.Z. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by "Nazarbayev University Collaborative Research Proposal #091019CRP2108 to F. Molnár." and "Nazarbayev University Faculty-Development Competitive Research Grant Program for 2023-2025 Grant No. 20122022FD4109, B. Tyler.".

Data Availability Statement: All data were obtained form https://data.mendeley.com/datasets/rscbjbr9sj/2.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mohammed, S.; Budach, L.; Feuerpfeil, M.; Ihde, N.; Nathansen, A.; Noack, N.; Patzlaff, H.; Naumann, F.; Harmouch, H. The Effects of Data Quality on Machine Learning Performance. *arXiv* **2024**, arXiv:2207.14529.
2. Noroozi, Z.; Orooji, A.; Erfannia, L. Analyzing the impact of feature selection methods on machine learning algorithms for heart disease prediction. *Sci. Rep.* **2023**, *13*, 22588. [CrossRef]
3. Rainio, O.; Teuho, J.; Klén, R. Evaluation metrics and statistical tests for machine learning. *Sci. Rep.* **2024**, *14*, 6086. [CrossRef] [PubMed]
4. Chen, R.J.; Lu, M.Y.; Chen, T.Y.; Williamson, D.F.K.; Mahmood, F. Synthetic data in machine learning for medicine and healthcare. *Nat. Biomed. Eng.* **2021**, *5*, 493–497. [CrossRef] [PubMed]
5. Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Papernot, N.; Anderson, R.; Gal, Y. AI models collapse when trained on recursively generated data. *Nature* **2024**, *631*, 755–759. [CrossRef]
6. Wolpert, D.; Macready, W. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
7. Kerschke, P.; Hoos, H.H.; Neumann, F.; Trautmann, H. Automated Algorithm Selection: Survey and Perspectives. *Evol. Comput.* **2019**, *27*, 3–45. [CrossRef]
8. Kashgarani, H.; Kotthoff, L. Is Algorithm Selection Worth It? Comparing Selecting Single Algorithms and Parallel Execution. *Proc. Mach. Learn. Res.* **2021**, *140*, 58–64.
9. Xu, L.; Hutter, F.; Hoos, H.H.; Leyton-Brown, K. SATzilla: Portfolio-based algorithm selection for SAT. *J. Artif. Intell. Res.* **2008**, *32*, 565–606. [CrossRef]
10. Kameyama, M.; Lukac, M.; Abdiyeva, K.; Kim, A. Reasoning and algorithm selection augmented symbolic segmentation. In Proceedings of the 2017 Intelligent Systems Conference (IntelliSys), London, UK, 7–8 September 2017; pp. 259–266. [CrossRef]
11. Fukunaga, A. Genetic algorithm portfolios. In Proceedings of the 2000 Congress on Evolutionary Computation, CEC00 (Cat. No.00TH8512), La Jolla, CA, USA, 16–19 July 2000; Volume 2, pp. 1304–1311. [CrossRef]
12. Belkhir, N.; Dréo, J.; Savéant, P.; Schoenauer, M. Feature Based Algorithm Configuration: A Case Study with Differential Evolution. In *Parallel Problem Solving from Nature–PPSN XIV: 14th International Conference, Edinburgh, UK, 17–21 September 2016*; Handl, J., Hart, E., Lewis, P.R., López-Ibáñez, M., Ochoa, G., Paechter, B., Eds.; Springer: Cham, Swizerland, 2016; pp. 156–166.
13. Lukac, M.; Bayanov, A.; Li, A.; Abiyeva, K.; Izbassarova, N.; Gabidolla, M.; Kameyama, M. Selecting Algorithms Without Meta-features. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, 10–15 January 2021, Proceedings, Part IV*; Springer: Cham, Swizerland, 2021; pp. 607–621.
14. Saparova, Z.; Lukac, M. Algorithm Selection with Priority Order for Instances. In Proceedings of the 37th NeurIPS Workshop on Attributing Model Behavior at Scale, New Orleans, LA, USA, 10–16 December 2023.
15. Mohr, F.; van Rijn, J.N. Fast and Informative Model Selection using Learning Curve Cross-Validation. *arXiv* **2021**, arXiv:2111.13914. [CrossRef]
16. Li, L.; Wang, Y.; Xu, Y.; Lin, K.Y. Meta-learning based industrial intelligence of feature nearest algorithm selection framework for classification problems. *J. Manuf. Syst.* **2022**, *62*, 767–776. [CrossRef]
17. Collins, A.; Beel, J. Meta-Learned Per-Instance Algorithm Selection in Scholarly Recommender Systems. *arXiv* **2019**, arXiv:1912.08694.
18. Maher, M.; Sakr, S. SmartML: A Meta Learning-Based Framework for Automated Selection and Hyperparameter Tuning for Machine Learning Algorithms. In Proceedings of the 22nd International Conference on Extending Database Technology (EDBT), Lisbon, Portugal, 26–29 March 2019. [CrossRef]
19. Feurer, M.; Eggensperger, K.; Falkner, S.; Lindauer, M.; Hutter, F. Auto-sklearn 2.0: Hands-free automl via meta-learning. *J. Mach. Learn. Res.* **2022**, *23*, 1–61.
20. König, H.; Hoos, H.H.; van Rijn, J.N. Towards algorithm-agnostic uncertainty estimation: Predicting classification error in an automated machine learning setting. In Proceedings of the ICML Workshop on Automated Machine Learning, Vienna, Austria, 12–18 July 2020.
21. Costa, A.J.; Santos, M.S.; Soares, C.; Abreu, P.H. Analysis of imbalance strategies recommendation using a meta-learning approach. In Proceedings of the 7th ICML Workshop on Automated Machine Learning (AutoML-ICML2020), Vienna, Austria, 12–18 July 2020; pp. 1–10.

22. Nagarajan, V.; Kolter, J.Z. Gradient descent GAN optimization is locally stable. *arXiv* **2017**, arXiv:1706.04156.
23. Shmelkov, K.; Schmid, C.; Alahari, K. How good is my GAN? In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 213–229.
24. He, R.; Sun, S.; Yu, X.; Xue, C.; Zhang, W.; Torr, P.; Bai, S.; Qi, X. Is synthetic data from generative models ready for image recognition? *arXiv* **2022**, arXiv:2210.07574.
25. Dahmen, J.; Cook, D. SynSys: A synthetic data generation system for healthcare applications. *Sensors* **2019**, *19*, 1181. [CrossRef]
26. Cruz, R.M.; Sabourin, R.; Cavalcanti, G.D.; Ing Ren, T. META-DES: A dynamic ensemble selection framework using meta-learning. *Pattern Recognit.* **2015**, *48*, 1925–1935. [CrossRef]
27. Zhang, S.; Zhu, Q.; Roy-Chowdhury, A. Adaptive algorithm selection, with applications in pedestrian detection. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3768–3772. [CrossRef]
28. Rice, J.R. The algorithm selection problem. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 1976; Volume 15, pp. 65–118.
29. Smith-Miles, K.A. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–25. [CrossRef]
30. Kotthoff, L. Algorithm selection for combinatorial search problems: A survey. In *Data Mining and Constraint Programming: Foundations of a Cross-Disciplinary Approach*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 149–190.
31. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661. [CrossRef]
32. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv* **2021**, arXiv:2112.10752.
33. Chen, Y.; Li, J.; Xiao, H.; Jin, X.; Yan, S.; Feng, J. Dual Path Networks. *arXiv* **2017**, arXiv:1707.01629.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
35. Huang, G.; Liu, Z.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2016**, arXiv:1608.06993.
36. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
37. Iandola, F.N.; Moskewicz, M.W.; Ashraf, K.; Han, S.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *arXiv* **2016**, arXiv:1602.07360.
38. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25, pp. 1097–1105.
39. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
40. Cristianini, N.; Ricci, E. Support Vector Machines. In *Encyclopedia of Algorithms*; Kao, M.Y., Ed.; Springer: Boston, MA, USA, 2008; pp. 928–932. [CrossRef]
41. Dobra, A. Decision Tree Classification. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: Boston, MA, USA, 2009; pp. 765–769. [CrossRef]
42. Mucherino, A.; Papajorgji, P.J.; Pardalos, P.M. k-Nearest Neighbor Classification. In *Data Mining in Agriculture*; Springer: New York, NY, USA, 2009; pp. 83–106. [CrossRef]
43. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
44. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
45. Cox, D.R. The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B (Methodol.)* **1958**, *20*, 215–232. [CrossRef]
46. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
47. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Hoboken, NJ, USA, 1994.
48. Kermany, D.S.; Goldbaum, M.; Cai, W.; Valentim, C.C.S.; Liang, H.; Baxter, S.L.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; et al. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell* **2018**, *172*, 1122–1131.e9. [CrossRef] [PubMed]
49. Kermany, D. Labeled optical coherence tomography (OCT) and chest X-ray images for classification. *Mendeley Data* **2018**. Available online: https://data.mendeley.com/datasets/rscbjbr9sj/2 (accessed on 5 September 2023).
50. Huang, S.C.; Chaudhari, A.S.; Langlotz, C.P.; Shah, N.; Yeung, S.; Lungren, M.P. Developing medical imaging AI for emerging infectious diseases. *Nat. Commun.* **2022**, *13*, 7060. [CrossRef]
51. Barratt, S.; Sharma, R. A Note on the Inception Score. *arXiv* **2018**, arXiv:1801.01973.
52. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *arXiv* **2018**, arXiv:1706.08500.
53. Salimans, T.; Goodfellow, I.J.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. *arXiv* **2016**, arXiv:1606.03498.
54. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *arXiv* **2015**, arXiv:1512.00567.

55. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
56. Chen, S.; Ma, K.; Zheng, Y. Med3d: Transfer learning for 3d medical image analysis. *arXiv* **2019**, arXiv:1904.00625.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.