
AFRL-RI-RS-TR-2021-208

# HIERARCHICAL LEARNING-GUIDED AUTOMATIC MODEL DISCOVERY

THE UNIVERSITY OF CALIFORNIA, BERKELEY

*DECEMBER 2021*

FINAL TECHNICAL REPORT

***APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED***

STINFO COPY

**AIR FORCE RESEARCH LABORATORY**
**INFORMATION DIRECTORATE**

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

# NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2021-208 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

**/ S /**
PETER A. JEDRYSIK
Work Unit Manager

**/ S /**
JULIE BRICHACEK
Chief, Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

# REPORT DOCUMENTATION PAGE

| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED | |
|---|---|---|---|
| | | START DATE | END DATE |
| DECEMBER 2021 | FINAL TECHNICAL REPORT | APRIL 2017 | JUNE 2021 |

**4. TITLE AND SUBTITLE**

HIERARCHICAL LEARNING-GUIDED AUTOMATIC MODEL DISCOVERY

| 5a. CONTRACT NUMBER | 5b. GRANT NUMBER | 5c. PROGRAM ELEMENT NUMBER |
|---|---|---|
| FA8750-17-2-0091 | N/A | 62702E |
| **5d. PROJECT NUMBER** | **5e. TASK NUMBER** | **5f. WORK UNIT NUMBER** |
| D3MP | S0 | 01 |

**6. AUTHOR(S)**

Dawn Song

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

The University of California, Berkeley (50853)
1680 4th Street, STE 220
Berkeley CA 94710-1749

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Air Force Research Laboratory/RISB
525 Brooks Road
Rome NY 13441-4505

DARPA / I2O
675 N. Randolph St.
Arlington VA 22203-2114

**10. SPONSOR/MONITOR'S ACRONYM(S)**

RI

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

AFRL-RI-RS-TR-2021-208

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

While participating in the DARPA Data Driven Discovery (D3M) program, we have explored many aspects necessary for automatic model and pipeline discovery. Based on insights gained we built a modular AutoML system performing well on a wide range of task and data types: tabular, image, graph, audio, video, text, etc. We have contributed code used by other participants in the program to enable collaboration and interoperability.

**15. SUBJECT TERMS**

D3M, AutoML, machine learning, artificial intelligence

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES |
|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | C. THIS PAGE | | |
| U | U | U | SAR | 17 |

| 19a. NAME OF RESPONSIBLE PERSON | 19b. PHONE NUMBER *(Include area code)* |
|---|---|
| PETER A. JEDRYSIK | N/A |



PREVIOUS EDITION IS OBSOLETE.

STANDARD FORM 298 (REV. 5/2020)
*Prescribed by ANSI Std. Z39.18*

## TABLE OF CONTENTS

## 1.0 SUMMARY

To address the issue on how to leverage the growth of data available to build useful models, an automatic approach to model and pipeline discovery is in order, which can utilize this data. We have explored many aspects necessary for automatic model and pipeline discovery: building a knowledge base of models and model ranking based on the recommender system approach, model recommendation through graphical representation of a dataset, pipeline generation by extending Tree-based Pipeline Optimization Tool (TPOT) and by a reinforcement learning based approach. We explored a budget aware hyperparameter tuning algorithm and uncertainty estimation of neural networks. We explored different training approaches including gradient-free optimization, zero-shot learning, and continual learning. We tackled neural network architectures as well. We tied all this together into a modular automated machine learning (AutoML) system supporting a wide range of task types which has been in top three spots in program evaluations.

## 2.0 INTRODUCTION

The explosion of digital data has created a wealth of opportunities. Companies and organizations are now able to collect and mine vast amounts of data that enable them to transform the way they operate and collaborate. However, in many cases the ability to leverage this data into specific insights or effective decision-making models lags far behind the ability to collect it. One of the main reasons for this gap is the reliance on tedious manual work at multiple points in the model discovery process.

Our goal was the design and development of an automated, holistic model discovery framework, able to incorporate human expert feedback. Given a dataset, a prediction task, and a performance metric, the framework utilizes a provided library of model primitives to automatically search through a space of models to generate candidate pipelines with models in an iterative process, and finally output pipelines with models that perform well on the given task and dataset. The dominant challenges in automatic model discovery include the immense search space, limited task-specific evaluation data, and cost associated with training and testing models at scale.

Our work has been done as part of the DARPA Data Driven Discovery (D3M) program. The program has been structured into three Technical Areas (TAs): TA1 participants worked on primitives, i.e., building blocks used for building machine learning pipelines, TA2 participants worked on AutoML systems, which were given a library of primitives and prepared datasets and had to produce machine learning pipelines, and TA3 participants who worked on user interfaces to involve subject-matter experts. Our work has been done in the TA2 context.

## 3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

We explored the following technical approaches:

1. Hierarchical Model Design: To manage the complexity of designing sophisticated models spanning many stages such as feature engineering and prediction, use of a hierarchical model formulation that decomposes models into high-level modules each composed of model primitives.

2. Meta-Models: Use of meta-models that are capable of scoring and even generating models given the meta-features of a modeling task. These meta-models leverage the hierarchical decomposition of models to simplify the prediction task.

3. Novel Meta-Model Training: Use of collected data from academic publications and public code and data repositories to construct a corpus for supervised and semi-superivsed meta-model training.

4. Iterative Search Procedure: A generic iterative meta-search procedure that leverages the meta-models to guide the design of the models and explore a range of search techniques.

5. End-to-End Training: Jointly optimizing the generated end-to-end models using first order methods in conjunction with auxiliary variable decomposition techniques to leverage efficient model-primitive training algorithms.

6. Code Synthesis: A domain-specific pipeline language that is capable of concisely capturing abstract model structure while enabling optimized instantiation across multiple system frameworks and hardware architectures.

To enable exploration, our AutoML system has been designed to be modular. This enables each of the components to be interchanged with a different implementation while keeping the rest.

## 4.0 RESULTS AND DISCUSSION

### 4.1 Model Recommendation

To prepare for our work on model recommendation we automatically constructed a knowledge base to use for metalearning. We automatically generated a list of machine learning algorithms by crawling Wikipedia and using machine learning to filter machine learning algorithms. Final manual cleanup of the list was required to remove duplicates, and adjust it to serve as the D3M primitives ontology. We downloaded 2,000 datasets from OpenML. 1129 of the datasets have the algorithm performance results, description text and OpenML meta-features. We extracted PubMed medical parameters data into SQL database. Additionally, we explored the Unified Medical Language System (UMLS) data in order to identify key terms in the medical field. We developed a Python based module that constructs the advanced meta-representations (by hand-crafted meta-features for datasets) of a given dataset and learning task. We evaluated the model ranking based on the recommender system approach (Sommelier) on an extensive public dataset. The proposed approach outperforms the Random Forest which was reported as the best model for this dataset.

Regarding pipeline pre-ranking and knowledge-base, few options for autopipeline generation were examined, TPOT tool was chosen to generate the first collection of pipelines for pipeline recommendation engine. In parallel with the D3M effort to design pipeline and pipeline-run schemas, we designed them also independently, in order to verify that D3M schemas cover all the aspects of internal ones. In order to enrich our knowledge-base we leveraged Kaggle API to download regression and classification datasets and publicly available kernels solutions. Approximately 500 datasets were downloaded.

We developed and evaluated the approach for model recommendation through graphical representation of a dataset called AutoGRD. AutoGRD first represents datasets as graphs and then extracts their latent representation that is used to train a ranking meta-model capable of accurately recommending top-performing algorithms for previously unseen datasets. We evaluated our approach on 250 datasets and demonstrated its effectiveness both for classification and regression tasks. AutoGRD outperforms state-of-the-art metalearning and Bayesian methods.

Our paper AutoGRD: “Model Recommendation Through Graphical Dataset Representation” won “Best research paper” in CIKM 2019.

For our clustering algorithm selection using metalearning and landmarking approach, metalearning method utilizes landmarking concepts recommending the most suitable clustering algorithm for an unseen dataset, evaluated on a corpus of 80 datasets. Landmarking meta-features: 5 meta-features (accuracy of the 5 basic classifiers) + 4 meta-features (DBSCAN vs K-means: metrics for comparison of the dataset partitions). DeepFM recommendation algorithm was used as a metalearner to rank the candidate algorithms that are anticipated to perform the best for the current dataset.

For our automatic selection of clustering algorithms using supervised graph embedding approach we first transformed datasets into graphs and then utilized graph convolutional neural networks to extract their latent representation. Using the embedding representations obtained we trained a ranking meta-model capable of accurately recommending top-performing algorithms for a new dataset and clustering evaluation measure.  Evaluation was done on 210 datasets, 13 clustering algorithms, and 10 clustering measures demonstrates the effectiveness of our approach and its dominance in terms of predictive and generalization performance over state-of-the-art clustering metalearning approaches.

### 4.2 Pipeline Generation

We explored multiple approaches to pipeline generation for different task types.

As one of the approaches to pipeline generation, we implement/integrate metalearner inside TPOT as a part of genetic algorithm/process to improve optimal pipeline search time to compare against TPOT baseline. We designed necessary pipeline metafeatures for that.

We also approached pipeline generation through reinforcement learning. We made a stable version of the DRL model. We tested it and compared it with baselines with promising results. Baselines that we used: Random Forest, XGBoost, Extra Trees and TPOT. We used D3M metalearning database dump and found ~11,000 pipelines that can be used and easily executed: sklearn based or using up to 3 most popular primitives in databases that are not in sklearn.

We trained metalearner for pipeline scoring\ranking on ~200,000 pipelines. We evaluated it on 100 classification datasets and 50 regression datasets with promising results. Based on TPOT as a baseline with 10,000 pipelines executed during the searching process per dataset vs top10 recommended pipelines in our approach we achieved comparable results.

We explored improvements of our system for semi-supervised problems. We implemented and integrated a few state of the art approaches found in academic literature, but have not been satisfied with results when tested on a wide range of datasets. More work is needed.

For our metalearning approach using dataset-to-image representation (based on dimensionality reduction techniques) we used the image representation of the dataset to train convolutional neural network (CNN) as a meta-model that will be able to recommend best performing algorithms for the unseen dataset. This research direction did not bring fruition.

### 4.3 Hyperparameter Tuning

We explored a budget aware hyperparameter tuning algorithm, which adaptively allocates resources to promising hyperparameters by using the long term predictions of their performances. We conducted extensive experiments to evaluate the performance of the algorithm, and compared to popular existing methods, including a state-of-the-art one. Our proposed method works well under a wide range of budget constraints, and outperforms Hyperband and Bayesian Optimization for most of the time across different tuning tasks. Analysis shows that our method is able to efficiently identify good configurations and allocate sufficient budget on one of such good configurations to achieve good performance.

### 4.4 Uncertainty Estimation

Predicting probability estimates representative of the true failure likelihood is an important and yet unsolved problem in deep learning. Despite the significant improvements in using neural networks to perform various complex predictions, there is still a lack of a unified approach to predict a calibrated uncertainty measurement in non-Bayesian deep models that can capture the true probability of failure associated with predictions made on previously seen and unseen data. We explored a novel approach for reliably learning the uncertainty of deep neural networks. We established comparisons of existing methods using the image classification task on MNIST, and evaluated and analyzed their performances in terms of calibration, and out-of-distribution sample detection.

Concretely, we assume that we are given a trained model $\theta$, and the training set. Given a test point $x'$ and the network's prediction $f(x'; \theta)$, how confident are we about $f(x'; \theta)$? We approached it using a pseudo-ensemble approach. We create a Gaussian distribution around $\theta$, with the covariance matrix being the inverse Hessian of the loss

computed over the training set. We sample multiple models from this distribution to create an ensemble, and use the predictions from this pseudo-ensemble to estimate our confidence for the original prediction f(x' ; θ). Note that we only train the model once to get the pseudo-ensemble, instead of training multiple times. We evaluate the proposed confidence score on uncertainty estimate and robustness tasks, like calibration, out-of-distribution detection, and adversarial example detection. We got compelling results compared to the state-of-the-art methods.

### 4.5 Training Approaches

In our work on gradient-free optimization we evaluated it by using it to perform autonomous driving tasks. By learning from both demonstration and environmental reward we develop a model that can learn with relatively few early catastrophic failures. We first learn an architecture of appropriate complexity to perceive aspects of world state relevant to the expert demonstration, and then mitigate the effect of domain-shift during deployment by adapting a policy demonstrated in a source domain to rewards obtained in a target environment. We show that our approach allows safer learning than baseline methods, offering a reduced cumulative crash metric over the agent's lifetime as it learns to drive in a realistic simulated environment.

We explored lifelong learning to address challenges of sequentially learning of tasks arriving in a continuous stream, especially when the model has a fixed capacity. Lifelong learning aims at learning new tasks without forgetting previously learnt ones as well as freeing up capacity for learning future tasks. We approached this by identifying the most influential parameters in a representation learned for one task that plays a critical role to decide on what to remember for continual learning, using statistically-grounded uncertainty defined in Bayesian neural networks.

We researched zero-shot learning as well. The ability to recognize novel classes through their semantic information without seeing them is a challenging problem known as zero-shot learning. Most of the proposed models addressing this challenge rely on learning a cross-modal mapping from the image feature space to a class embedding space or vice versa. However, learning a shared cross-modal embedding by aligning the latent spaces of modality-specific autoencoders is shown to be promising in (generalized) zero-shot learning. By following this direction, we researched a novel approach to both zero shot and generalized zero-shot learning that learns a shared latent representation for image features and additional data by using variational autoencoders.

Continual learning is the ability of learning from a continuous stream of data, building on what was learnt previously, hence exhibiting positive forward transfer, as well as being able to retain information learned on previously seen tasks, also known as backward transfer. Continual learning becomes more challenging when data is imbalanced and has a long tail in its distribution. We developed algorithms that are capable of transferring knowledge from tasks with a limited number of examples as few as one shot to tasks with significantly more examples with thousands of examples, while preserving the representation learned for the tail. Building upon our previous work on few shot learning and lifelong learning we enhanced our Bayesian lifelong learning framework (BLLL) with better few shot learning capabilities to avoid catastrophic forgetting. We approached this by aligning the distributions learned from images and attributes seen at the tail to construct latent features with better generalization capabilities. Such representations learned from the tail are less required to change in order to adapt to new tasks with bigger datasets at the head, hence forgetting less and generalizing more to previously unseen tasks.

We made a new algorithm on how to select the most representative samples from an unlabeled data pool to obtain maximum performance on the desired task using VAE+adversarial learning. It facilitates less human effort to label large scale datasets. We continued our work on continual learning of multiple tasks in model-based RL by learning how to grow without supervision. Continual learning aims to learn new tasks without forgetting previously learned ones. This is especially challenging in reinforcement learning (RL) where policy (in model-free RL) or model (in model-based RL) learned on sequential tasks are usually difficult to transfer and suffer from catastrophic forgetting. Moreover, task boundaries are not always clear in RL and hence automating the process of sequential learning is very beneficial. While continual learning has been widely explored in model-free RL, there is a lack of unified approach that formulates and measures forgetting in model-based RL (MBRL). In our continual learning approach for MBRL we automate learning hundreds of PyBullet tasks by learning how to grow the architecture using uncertainty in Bayesian and Frequentist deep neural networks without access to data belonging to previous tasks. We evaluate our approach's ability on backward and forward knowledge transfer and report superior or on-par performance compared to existing approaches.

We explored utilizing unsupervised and self-supervised learning to improve the data-efficiency and generalization of Reinforcement Learning (RL) from pixels. In CURL: Contrastive Unsupervised Representations for Reinforcement Learning, significant data-efficiency gains over the state of the art were shown on Atari and DeepMind control benchmarks by training contrastive representations alongside a model-free RL objective. We also showed for the first time that learning policies from pixels can be as

data-efficient as learning from coordinate state. In a follow up work, Reinforcement Learning with Augmented Data (RAD), an extensive study of data augmentation for pixel-based RL also showed substantial improvements in data-efficiency and generalization abilities of model-free RL on DeepMind control and ProcGen benchmarks.

### 4.6 Neural Network Architecture Search

The successes of deep learning in recent years has been fueled by the development of innovative new neural network architectures. However, the design of a neural network architecture remains a difficult problem, requiring significant human expertise as well as computational resources. To address this, we developed a method for transforming a discrete neural network architecture space into a continuous and differentiable form, which enables the use of standard gradient-based optimization techniques for this problem, and allows us to learn the architecture and the parameters simultaneously. We have evaluated using CIFAR-10 and Udacity steering angle prediction dataset, and show that our method can discover architectures with similar or better predictive accuracy but with fewer parameters and smaller computational cost.

We extended the previous paradigm (search over chain architectures), to also explore different architectures. The existing aggregation schemes are all human designed, and can be sub-optimal, less accurate, inefficient, or not adaptive to the dataset. To address this problem, we explored ways to learn the aggregation scheme automatically. Different from the previous architecture search method which searches for the whole network, we only search the aggregation scheme with a set of pre-trained layers to be aggregated to find which layers and nodes to be aggregated. This method not only results in achieving the best possible accuracy but also speeds up the search process because we do not need to train the sampled network from scratch, given the pre-trained layer banks.

### 4.7 AutoML System

We designed and developed an AutoML system called Aika which serves both as a stand-alone AutoML system and as a platform to test novel approaches to AutoML. Its modular architecture enables reuse of code while exploring a novel approach in one part: pipeline generation, pipeline rewriting, hyper-parameter tuning, runner, evaluator, and searcher. It is written in Python and built on top of the Ray framework for distributed

computation, allowing it to run both on a single machine and on a cluster. It supports a large range of task types. It is available at https://gitlab.com/aika/aika.

In D3M evaluations it has usually been in top three spots in comparison with other AutoML systems in the program.

### 4.8 Shared Codebase

We collaborated with other participants in the D3M program to assure interoperability and improved quality of the D3M ecosystem. We greatly contributed to the design and implementation of the shared codebase which also serves as a base over which our AutoML system is made. We contributed towards a shared metalearning database where all pipelines produced by all AutoML systems in the D3M program can be stored and shared so that they are available to all.

## 5.0 CONCLUSIONS

During the project we have explored many aspects necessary for automatic model and pipeline discovery and tied all this together into a modular AutoML system supporting a wide range of task and data types which has been in top three spots in D3M program evaluations. Our AutoML system supports tabular, image, graph, audio, video, and text datasets and is able to combine primitives from the D3M primitives library into machine learning pipelines automatically. The AutoML system is built on top of the shared codebase towards which we have contributed to enable collaboration and interoperability between all participants in the program.

**APPENDIX - Publications**

Vainshtein, R., Greenstein-Messica, A., Katz, G., Shapira, B., Rokach, L., "A Hybrid Approach for Automatic Model Recommendation," *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*, New York, NY, USA, 2018, pp. 1623-1626.

Heffetz, Y., Vainshtein, R., Katz, G., Rokach, L., "DeepLine: AutoML Tool for Pipelines Generation using Deep Reinforcement Learning and Hierarchical Actions Filtering," *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug 2020, pp. 2103-2113.

Laadan, D., Vainshtein, R., Curiel, Y., Katz, G., Rokach, L., "MetaTPOT: Enhancing A Tree-based Pipeline Optimization Tool Using metalearning," *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, Oct 2020, pp. 2097-2100.

Cohen-Shapira, N., Rokach, L., Shapira, B., Katz, G., Vainshtein, R., "AutoGRD: Model Recommendation Through Graphical Dataset Representation," *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*, New York, NY, USA, 2019, pp. 821-830.

Laadan, D., Vainshtein, R., Curiel, Y., Katz, G., Rokach, L., "RankML: a Meta Learning-Based Approach for Pre-Ranking Machine Learning Pipelines," arXiv preprint arXiv:1911.00108, 2019.

Greenstein-Messica, A., Vainshtein, R., Katz, G., Shapira, B., Rokach, L. "Automatic Machine Learning Derived from Scholarly Big Data," arXiv preprint arXiv:2003.03470, 2020.

Milutinovic, M., Güneş Baydin, A., Zinkov, R., Harvey, W., Song, D., Wood, F., Shen, W., "End-to-end Training of Differentiable Pipelines Across Machine Learning Frameworks," *NIPS 2017 Workshop Autodiff Submission*, Oct 2017.

Milutinovic, M., Schoenfeld, B., Martinez-Garcia, D., Ray, S., Shah, S., Yan, D., "On Evaluation of AutoML System," *7th ICML Workshop on Automated Machine Learning*, 2020.

## LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| AutoML | Automated machine learning |
| BLLL | Bayesian Lifelong Learning Framework |
| CNN | Convolutional neural network |
| D3M | Data-Driven Discovery of Models |
| DARPA | Defense Advanced Research Projects Agency |
| DBSCAN | Density-Based Spatial Clustering of Applications with Noise |
| DRL | Deep Reinforcement Learning |
| MBRL | Model-Based Reinforcement Learning |
| RL | Reinforcement Learning |
| SQL | Structured Query Language |
| TA | Technical Area |
| TPOT | Tree-based Pipeline Optimization Tool |
| UMLS | Unified Medical Language System |
| VAE | Variational Autoencoder |