International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

© Global Society of Scientific Research and Researchers

http://ijcjournal.org/

# An Overview of the Algorithm Selection Problem

Salisu Mamman Abdulrahman[a]*, Alhassan Adamu[b], Yazid Ado Ibrahim[c], Akilu Rilwan Muhammad[d]

*[a,b,c]Kano University of Science and Technology Wudil, Kano State, Nigeria*

*[d]Federal University Dutse, Jigawa State, Nigeria*

*[a]Email: salisu.abdul@gmail.com,*

*[b]Email: kofa062@gmail.com,*

*[c]Email: yazidado2002@yahoo.com*

*[d]Email: akilrilwan@gmail.com*

**Abstract**

Users of machine learning algorithms need methods that can help them to identify algorithm or their combinations (workflows) that achieve the potentially best performance. Selecting the best algorithm to solve a given problem has been the subject of many studies over the past four decades. This survey presents an overview of the contributions made in the area of algorithm selection problems. We present different methods for solving the algorithm selection problem identifying some of the future research challenges in this domain.

***Keywords:*** Machine Learning; Algorithm selection; Workflows.

## 1. Introduction

A large number of data mining algorithms exist, rooted in the fields of machine learning, statistics, pattern recognition, artificial intelligence, and database systems, which are used to perform different data analysis tasks on large volumes of data. The task to recommend the most suitable algorithms has thus become rather challenging. Moreover, the problem is exacerbated by the fact that it is necessary to consider different combinations of parameter settings, or the constituents of composite methods such as ensembles. The algorithm selection problem, originally described by Rice [1], has attracted a great deal of attention, as it endeavours to select and apply the best algorithm(s) for a given task [2, 3].

------------------------------------------------------------------------

* Corresponding author.

The algorithm selection problem can be cast as a *learning* problem: the aim is to learn a model that captures the relationship between the properties of the datasets, or meta-data, and the algorithms, in particular their performance. This model can then be used to predict the most suitable algorithm for a given new dataset. Selecting the best algorithm to solve a given problem has been the subject of many studies over the past four decades [4, 5, 6, 3, and 1]. Researchers have long ago recognized that it is difficult to identify a single best algorithm that will give the best performance across all problems. This is why later on many researchers have developed different approaches to addressing the algorithm selection problems. There are many approaches to addressing the algorithm selection problem. Two of the most popular approaches are the *Metalearning* [2] and *Surrogate models* [30, 31] or *hyperparameter optimization*. Our Review is limited to these two approaches to algorithm selection. The Meta-learning approach leverages knowledge of past algorithm applications to learn how to select the best techniques for future applications, and offers effective techniques that are superior to humans both in terms of the end result and especially in the time required to achieve it. The area of hyperparameter optimization, the aim is to identify a set of hyperparameters for a learning algorithm, usually with the goal of obtaining good generalization performance.

## 2. The Algorithm Selection Framework of Rice

The algorithm selection problem, discussed first by Rice [1], has become especially relevant in the last decades, as researchers are increasingly investigating how to identify the most suitable existing algorithm for solving a problem instead of developing new algorithms. This problem is concerned with selecting the most appropriate algorithm for a given particular problem. The classical area of application for algorithm selection in machine learning is classification [7]. Smith-Miles [3] extended this scheme to other areas including time series prediction, regression, sorting, constraint satisfaction and optimization.

Following Rice [1] and Vanschoren [8], the algorithm selection problem can be stated as follows:

**Definition:** For a given problem instance $x \in P$ , with features $f(x) \in F$ , find the selection mapping $S(f(x))$ into the algorithm space $A$, such that the selected algorithm $\alpha \in A$ maximizes the performance mapping y( $\delta(\alpha \in A)) \in Y$.

The algorithm selection problem as defined above can be briefly described as follows:

Given $x$ as problem subset of the problem space $P$ (the space of all learning problems), the feature space $F$ of all measurable characteristics of each of the problems in $P$, calculated by a feature extraction process $f(x)$ , $\alpha$ as subset of the algorithm space $A$ (the set of all base-level learning algorithms), and the performance measure space $Y$ representing the mapping of each algorithm in $A$ to a set of performance metrics, meta-learning is applied to determine the $S$ (i.e., the mapping of problems to algorithms) so as to obtain an algorithm with high performance. The model is shown in Figure 1 and contains four main components. The **problem space** is characterized by the datasets used for the study. The **feature space** is the set of characteristics of the underlying problem (attributes of the dataset) that are used to represent the problem.

The **algorithm space** is the set of algorithms from which we can obtain a solution to a given problem. In our setting, the algorithm space also includes the set of possible parameter settings that the machine learning algorithm can assume. The parameter settings are designed by the developers of the given algorithm to modify the behavior of the machine learning algorithm. The combinations of possible parameter values make up the configuration or parameter space for a given algorithm.

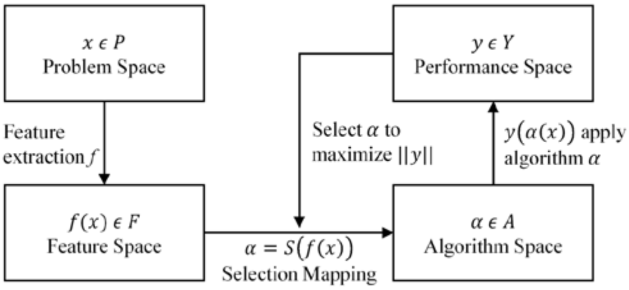

**Figure 1:** Rice's framework for algorithm selection [1, 8].

The **performance measures space** is the range of measures that characterize the behavior of an algorithm on a given problem. These may include for instance classification accuracy, speed of execution, and use of memory. There are many works that are relevant to algorithm selection in Machine Learning literature. Smith-Miles [3] considers algorithm selection as a learning problem and presents a survey of various past approaches. Prudencio and his colleagues [9] investigate on how metalearning for algorithm selection can be applied to select algorithms in the area of time series forecasting. A comprehensive and recent work of the subject are presented in [10, 11], where time series are clustered according to their characteristics and recommendation rules are derived with aid of machine learning algorithms.

### *2.1 An Extension of the Rice's Framework*

There have been various extensions to the original framework of Rice. Vanschoren [8] argues that Rice's framework [1] does not capture some important aspects of meta-learning. That is why an extension was proposed in [8], shown in dashed lines in Figure 2. First, nearly all ML algorithms have a range of parameter settings which have a profound impact on their performance on a specific problem $P$. The aim is consider the effect of these parameter settings as well. His proposal was to introduce an extended space consisting of algorithms with a specific set of parameter settings.

Many authors [12, 13] argue that algorithms with specific learning parameters are simply different learning algorithms. However, Vanschoren maintains a distinction between how well an algorithm can perform in general, and what the effects of their parameters are.

In many situations, it may be crucial to be able to predict useful pre-processing steps for a given algorithm. For this reason, Vanschoren introduced the space of all pre-processed problems $P'$, in which $x'$ is a dataset that has been pre-processed in a certain way. Finally, Vanschoren [8] introduced the space $G$ of measurable algorithm features to be able to generalize over learning algorithms to find patterns involving ***properties*** of algorithms.

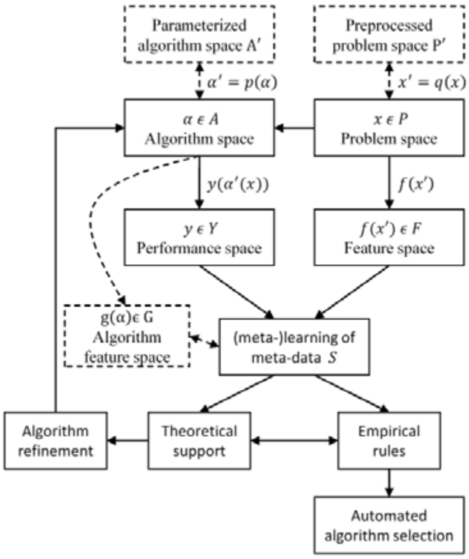

**Figure 2:** Proposed framework for algorithms selection using meta-learning approach in Smith-Miles [3] (in (full lines) and extensions (dashed lines) of Vanschoren [8].

## 3. Different Approaches to Algorithm Selection

In this section, we briefly review some advances in the algorithm selection problems by presenting an overviews of different tools and techniques developed over the last two decades to solve the algorithm selection problems.

### *3.1 StatLog and Data Mining Advisor (DMA)*

The first large scale metalearning study carried-out within the StatLog project [14, 4] used 19 data characteristics and 10 algorithms. This system marked algorithms as either *applicable* or *non/applicable* during the training phase, on the basis of similarity between the best algorithm on a given dataset. A decision tree model was generated for each algorithm predicting whether or not it is applicable on a new dataset. The system finally generated a set of learned rules that had to be checked manually.

The idea of the StatLog project was further automated [15] in project: *Meta-learning assistant for providing user support in machine learning and data mining* (METAL 2002) and investigated model selection and combination approaches. A web-based system providing rankings of classification algorithms for users resulted in a tool called the *Data Mining Advisor* (DMA) [15]. This system stored the actual performance measures for all algorithms and a k-Nearest Neighbor algorithm (k-NN) was trained to predict how well the algorithms would perform on a given new dataset. It then produced a ranking of all algorithms according to user-specified objectives. The user of the system could upload a new dataset via a web-based system. The system automatically calculated the meta-features of the new dataset and the ranking was returned subsequently.

### *3.2 The Intelligent Discovery Electronic Assistant (IDEA)*

The *Intelligent Discovery Electronic Assistant}* (IDEA) Bernstein and his colleagues [16] is the first planning-based data analysis system for data mining able to construct workflows. This system considers pre-processing, modeling, and post-processing techniques as operators and returns all valid plans (sequences of operations) that are possible for the given problem. This system contains an ontology of operators (which serve as its meta-knowledge) describing the preconditions and effects of each operator, as well as manually defined heuristics which allows it to produce a ranking of all generated plans according to the user's objectives. Finally, based on this ranking the user may select a number of processes to be executed on the provided data. After the execution of a plan, the user can review the results and refine the weights to obtain alternative rankings. For instance, the user might sacrifice some speed in order to obtain a more accurate model. Finally, if useful partial workflows have been discovered, the system also allows extending the ontology by adding them as new operators.

Although in IDEA there is no actual meta-learning involved, planning can be viewed as a search for the best plan given the new problem (dataset), just as learning is regarded as a search for the best hypothesis given new problem. In the case of IDEA, this is achieved by intensively generating all possible knowledge discovery plans with the hope to finding useful workflows.

### *3.3 The e-Lico Intelligent Discovery Assistant (eIDA)*

The *e-Lico Intelligent Discovery Assistant* (eIDA), born out of the e-Lico [17] creates data mining processes based on the specification of input data and the user's specific goal. It makes use of the *Data Mining Workflow Ontology* (DMWF) [18] which stores operator inputs, outputs with preconditions and effects in the form of *Semantic Web Rule Language* (SWRL) rules (stored as annotations in the ontology). It also uses a hierarchical task network (HTN) planner implemented in Flora2 [19]. The DMWF ontology is queried and the inputs, outputs, preconditions and effects are translated to Flora2 for planning.

These plans are then ranked using a second ontology called *Data Mining Optimization Ontology* (DMOP) [20], which stores detailed properties of the operators. The system also offers a modeling tool, eProPlan [18] for modeling data mining operators and defining the HTN grammar for guiding the planning process.

The system uses the specification of input data, as well as the modelling task, to automatically create processes tailored specifically to this data. It analyzes hundreds of processes and selects the ones that are well-suited for the problem and data set at hand. This is done by choosing operators that have achieved good accuracy on similar data sets in the past. It also handles preprocessing tasks, such as normalization, discretization, or missing value replacement when required by the learning algorithm which may be necessary for applying certain algorithms. eIDA has an API interface that can be easily integrated into existing data mining-suites such as RapidMiner [21].

Many more algorithms selection techniques are described in the literature (e.g., Mining Mart [22]). Some of these systems introduced novel data characteristics such as (subsampling) landmarkers, or make use of different algorithms for building meta-models, such as boosted decision trees [23], predictive clustering trees [24], regression algorithms [25] and neural networks [26]. Some introduce new implementation frameworks, such as METALA [27, 28] and [29]. One overview of these systems can be found in [8].

### *3.4 Auto-WEKA*

Auto-WEKA [30, 31] is a tool designed to help novice users of ML by automatically searching through the joint space of WEKA's learning algorithms and their respective hyperparameter settings to maximize a given performance measure (for instance accuracy, AUC, etc.) by using a state-of-the-art Bayesian optimization method. This problem referred in [31] as *Combined Algorithms Selection and Hyperparameter Optimization* (CASH), is then considered as single hierarchical hyperparameter optimization problem in which even the choice of algorithms is itself considered a hyperparameter. Based on this consideration, recent Bayesian optimization methods namely: *Sequential Model-based Algorithm Configuration SMAC* [32] and *Tree-structured Parzen Estimator* (TPE) [33] are used as candidates for the task of combined algorithm selection and hyperparameter optimization.

To test their automatic approach to solving the CASH problem, the tool was evaluated on 21 prominent benchmark datasets and 39 WEKA classification algorithms consisting of 27 base classifiers, 10 meta-methods and 2 ensemble classifiers that can take any number of base classifiers as inputs (uses five classifiers in Auto-WEKA) [31]. A feature selection method was used for pre-processing before building a classifier using WEKA's 3 feature search method as well as its 8 feature evaluators. Two baseline methods were used in Auto-WEKA. The first uses default parameter settings and performs exhaustive 10-fold cross validation on the training set and return the classifier with the smallest average misclassification error. The second stronger baseline referred to as *random grid* uses grid search for hyperparameters for each of the 27 base classifiers and executes the random grid search for all the 21 datasets in parallel, using 400 CPU hours on average per dataset and compare this performance to the one that uses default parameters.

The authors compare how effective SMAC and TPE are in searching the complex space of hierarchical hyperparameters to optimize performance with respect to the two baseline methods. On the choice of the two different optimizers for searching Auto-WEKA's 786-dimensional parameter space, they recommend the Auto-WEKA variant based on the Bayesian optimization method SMAC [32]

Other algorithm selection tools include *ASlib* [34] }, a benchmark library for algorithm selection containing 17 algorithm selection scenarios from six different areas with a focus on (but not limited to) constraint satisfaction problems, *AutoFolio* [35] that makes use of SMAC [32] to automatically determine a well-performing algorithm selection approach and its hyper-parameters for a given algorithm selection data and *Leveraging Learning to Automatically Manage Algorithms* (LLAMA) [36], an R package for algorithm portfolios and selection.

### *3.5 Auto-SKLearn*

Auto-sklearn [37] leverages the recent advantages in Bayesian optimization, meta-learning and ensemble construction to provide an automated machine learning toolkit. Similar to the Auto-WEKA, it makes use of the state-of-the-art Bayesian optimization techniques to configure a flexible machine learning pipeline implemented *scikit-learn* [38] (a machine learning library for the Python programming language that includes various classification, regression and clustering algorithms). In one experiment [37], auto-sklearn uses 15 classifiers, 14 feature pre-processing methods, and 4 data pre-processing methods, giving rise to a structured hypothesis space with 110 hyperparameters. It improves on existing AutoML methods like Auto-WEKA by using the meta-learning by automatically taking into account past performance on similar datasets to warm-start the Bayesian optimization procedure which results in a considerable boost in efficiency. Auto-sklearn also includes an automated ensemble construction step that allows the use of all classifiers evaluated during the Bayesian optimization.

## 4. Recommendations

As explained in the introduction, two of the most popular approaches to algorithm selection are the *Metalearning* and *hyperparameter optimization* approaches. One of the interesting areas to explore in algorithm selection is on ***combining metalearning and optimization approaches***. In paper, we have presented different approaches to algorithm selection using both  metalearning that leverages knowledge of past experiments and the search-based approach that uses experience gained on the new dataset to intelligently try out various algorithms (and parameter settings) to improve the learning episode. Recently, attempts to combine the two resulted in interesting ideas and solid results, such as Auto-WEKA [30, 31] and Auto-SKLEARN [37], however there has been little follow-up. Combining these two paradigms successfully has the potential to push the state of the art and lead to even better results.

## 5. Conclusion

Selecting the best algorithm to solve a given problem has been the subject of many studies over the past four decades [4, 5, 6, 3, and 1]. In this paper, we have covered briefly the state-of-the-art in the field machine learning for solving the algorithm selection problem. We have given an overview of algorithms selection framework, discussing different existing approaches to addressing the algorithm selection problem.

## References

[1] Rice, J. R. (1976). The algorithm selection problem. Advances in computers, 15, 65-118.

[2] Brazdil, P., Carrier, C. G., Soares, C., & Vilalta, R. (2008). Metalearning: Applications to data mining. Springer Science & Business Media.

[3] Smith-Miles, K. A. (2009). Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Computing Surveys (CSUR), 41(1), 6.

[4] Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine learning, neural and statistical classification.

[5] Brazdil, P. B., Soares, C., & Da Costa, J. P. (2003). Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning, 50(3), 251-277.

[6] Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. Artificial Intelligence Review, 18(2), 77-95.

[7] Lemke, C., Budka, M., & Gabrys, B. (2015). Metalearning: a survey of trends and technologies. Artificial intelligence review, 44(1), 117.

[8] Vanschoren, J. (2010). Understanding machine learning performance with experiment databases. lirias. kuleuven. be, no. May.

[9] Prudêncio, R. B., & Ludermir, T. B. (2004). Meta-learning approaches to selecting time series models. Neurocomputing, 61, 121-137.

[10] Wang, X., Smith-Miles, K., & Hyndman, R. (2009). Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. Neurocomputing, 72(10), 2581-2594.

[11] Lemke, C., & Gabrys, B. (2010). Meta-learning for time series forecasting and forecast combination. Neurocomputing, 73(10), 2006-2016.

[12] Soares, C., & Brazdil, P. B. (2006, April). Selecting parameters of SVM using meta-learning and kernel matrix-based meta-features. In Proceedings of the 2006 ACM symposium on Applied computing (pp. 564-568). ACM.

[13] Aha, D. W. (1992). Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. International Journal of Man-Machine Studies, 36(2), 267-287.

[14] King, R. D., Feng, C., & Sutherland, A. (1995). Statlog: comparison of classification algorithms on large real-world problems. Applied Artificial Intelligence an International Journal, 9(3), 289-333.

[15] Giraud-Carrier, C. (2005, December). The data mining advisor: meta-learning at the service of practitioners. In Machine Learning and Applications, 2005. Proceedings. Fourth International Conference on (pp. 7-pp). IEEE.

[16] Bernstein, A., Provost, F., & Hill, S. (2005). Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. IEEE Transactions on knowledge and data engineering, 17(4), 503-518.

[17] www.e-lico.eu, Retrieved on 5th January 2017.

[18] Kietz, J., Serban, F., Bernstein, A., & Fischer, S. (2009, September). Towards cooperative planning of data mining workflows. In Proceedings of the Third Generation Data Mining Workshop at the 2009 European Conference on Machine Learning (ECML 2009) (pp. 1-12).

[19] Yang, G., Kifer, M., Zhao, C., & Chowdhary, V. (2005). FLORA-2: User’s manual. Version 0.94

(Narumigata). April, 30.

[20] Hilario, M., Kalousis, A., Nguyen, P., & Woznica, A. (2009, September). A data mining ontology for algorithm selection and meta-mining. In Proceedings of the ECML/PKDD09 Workshop on 3rd generation Data Mining (SoKD-09) (pp. 76-87).

[21] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006, August). Yale: Rapid prototyping for complex data mining tasks. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 935-940). ACM.

[22] Morik, K., & Scholz, M. (2004). The miningmart approach to knowledge discovery in databases. Intelligent technologies for information analysis, 47-65.

[23] Kalousis, A. (2002). Algorithm selection via meta-learning. University of Geneva, Genebra.

[24] Todorovski, L., Blockeel, H., & Dzeroski, S. (2002, August). Ranking with predictive clustering trees. In European Conference on Machine Learning (pp. 444-455). Springer, Berlin, Heidelberg.

[25] Bensusan, H., & Kalousis, A. (2001). Estimating the predictive accuracy of a classifier. Machine Learning: ECML 2001, 25-36.

[26] Castellano, G., Castiello, C., Fanelli, A. M., & Mencar, C. (2005). Knowledge discovery by a neuro-fuzzy modeling framework. Fuzzy sets and Systems, 149(1), 187-207.

[27] Botía, J., Gómez-Skarmeta, A., Valdés, M., & Padilla, A. (2001). Metala: A meta-learning architecture. Computational Intelligence. Theory and Applications, 688-698.

[28] Hernansaenz, J. M., Botía, J. A., & Skarmeta, A. F. (2016). METALA: a J2EE technology based framework for web mining. Revista Colombiana de Computación-RCC, 5(1).

[29] Grabczewski, K., & Jankowski, N. (2007, March). Versatile and efficient meta-learning architecture: Knowledge representation and management in computational intelligence. In Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on (pp. 51-58). IEEE.

[30] Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2016). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. Journal of Machine Learning Research, 17, 1-5.

[31] Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013, August). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 847-855). ACM.

[32] Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential Model-Based Optimization for General Algorithm Configuration. LION, 5, 507-523.

[33] Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In Advances in Neural Information Processing Systems (pp. 2546-2554).

[34] Bischl, B., Kerschke, P., Kotthoff, L., Lindauer, M., Malitsky, Y., Fréchette, A., ... & Vanschoren, J. (2016). Aslib: A benchmark library for algorithm selection. Artificial Intelligence, 237, 41-58.

[35] Lindauer, M., Hoos, H. H., Hutter, F., & Schaub, T. (2015). Autofolio: An automatically configured algorithm selector. Journal of Artificial Intelligence Research, 53, 745-778.

[36] Kotthoff, L. (2013). LLAMA: leveraging learning to automatically manage algorithms. arXiv preprint arXiv:1306.1031.

[37] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In Advances in Neural Information Processing Systems (pp. 2962-2970).

[38] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O.,  & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.