Modeling and Forecasting Armed Conflict: AutoML with Human-Guided Machine Learning

Vito D'Orazio
Department of Political Science
University of Texas at Dallas
Dallas, TX
dorazio@utdallas.edu

James Honaker*, Raman Prasad[†], Michael Shoemate[‡]
Center for Research on Computation and Society
Harvard John A. Paulson School of Engineering and Applied Sciences
Cambridge, MA
*honaker@seas.harvard.edu,
[†]raman_prasad@harvard.edu,
[‡]shoematem@seas.harvard.edu

***Abstract*—Machine learning has made slow inroads into quantitative social science due to both a mismatch of machine learning's strengths to the causal and inferential tasks domain researchers pursue [1] and also a lack of algorithmic training among many domain experts [2]. However, conflict research–the empirical examination of political unrest, violence and civil war–has seen a growing emphasis on prediction and forecasting models. We describe automated machine learning (AutoML) to identify models, and human-guided machine learning (HGML), and show how these can incorporate domain knowledge and research requirements into model selection and assessment, and provide high quality machine learning pipelines to domain experts comparable to state-of-the-literature solutions. We examine three peer-reviewed papers with predictive models of conflict [3, 4, 5] and run their data through our HGML system using multiple AutoML engines and find this system produces slightly elevated performance on each paper's model, without any ML expertise required of the user. Our research has three takeaways for computational social science. First, predictive models of conflict would benefit from even minimal applications of AutoML; Secondly, human-guided machine learning offers the attractive option of constraining AutoML systems to address the kinds of questions conflict researchers assess with predictive models; Finally, current existing AutoML implementations produce divergent solutions and so can be productively harnessed in parallel.**

***Index Terms*—Human-guided machine learning; Automated machine learning (AutoML); conflict forecasting.**

I. Introduction

Machine Learning approaches to understanding empirical data typically emphasize predictive accuracy over parsimony or interpretability. As such, they can be harder to leverage for the direct hypothesis tests employed for empirically testing theories, as is the mainstay of quantitative political science and the social sciences more broadly. Increasingly, however, researchers use predictive models of conflict for multiple purposes, including policy guidance [6], sensitivity analysis [7], as well as theory comparison [8]. In hand with these purposes, machine learning has made increasing inroads into the study of conflict.

Automated machine learning (AutoML) is a recently exploding approach to model selection and assessment in machine learning [9, 10]. Typically in classical machine learning (ML), expert researchers construct an ML model that is appropriate to key features of the data, and the machine learning algorithm concentrates on optimizing a large number of available model parameters to increase predictive accuracy. The appropriate selection of the underlying ML algorithm to match the task at hand, as well as routines for cleaning and transforming the original data, require the manual contribution of a researcher with ML expertise. Increasingly, however, in industry and research an emerging goal is to provide ML frameworks which can work off the shelf in easy to use applications for those who are not ML experts.

The AutoML approach is to combine the choice of algorithm, as well as the optimization of parameters and hyperparameters, as well as any necessary data transformation steps, into one larger search problem. An AutoML system is provided with data and a description of the problem, and the system searches the space of potential models to find the best solution to the problem, including all the steps necessary for data preparation, feature selection, hyper-parameter tuning and ensembling. However, the success of the AutoML approach, to automate away all human contribution, makes it hard for the resulting model to be either shaped by the underlying goal of the researcher or leverage the domain knowledge about the world they believe to be true. Human-guided machine learning (HGML) is an approach for AutoML that reincorporates domain knowledge directly into the solution by interacting with users for guidance [11]. As Gil et al. define, the goal of HGML is to create *"…systems that allow a domain expert, without a machine learning expert, to use relevant domain knowledge to inform the automated search for a high quality, impactful and interpretable model, including necessary data preparation steps necessary for analysis."*

To allow the AutoML system to be guided by the domain expert, we require it to allow the user to request *tasks*, bounded, encapsulated points of guidance from the user to the system. As an example, two variables may represent different

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement number FA8750-17-2-0114. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

operationalizations of the same concept, and a researcher may want to assess which variable yields better performance. To conduct this test, the AutoML system must be constrained in a way that aligns with the researcher's objectives. We consider here an extensive (and growing) catalog of tasks that we believe cover the empirical workflow and needs of quantitative social science researchers.

This paper has two goals: first, to investigate whether task oriented HGML interfaces provide sufficient control of machine learning systems to replicate the work of expert quantitative researchers.Second, to present initial results from our system to demonstrate that, given a dataset and a problem, the system is able to automatically produce models that perform comparably to those in the peer-reviewed literature.

II. From Automated to Human-Guided Machine Learning

To discuss automated machine learning (AutoML) and human-guided machine learning (HGML), it is necessary to distinguish a predictive model from an inferential model. We summarize James et al. [12, Chapter 2], who describe the difference well. To begin, we can write the relationship between a variable Y and a set of variables X as

$$Y = f(X) + \epsilon \tag{1}$$

Since f in unknown, we instead form an estimate $\hat{f}$ under some particular modelling assumptions. Any $\hat{f}(X)$, whether considered predictive or inferential, is a model that can produce predicted values, $\hat{Y}$. We refer to these models as solutions $s \in S$.

$$\hat{Y}_s = \hat{f}_s(X) + \epsilon \tag{2}$$

A predictive model emphasizes the ability to predict Y. This is different from an inferential model, where the emphasis is on estimating the expected change in Y as a function of a change in X. To specify a good predictive model, we cycle through solutions to find the one that performs best. Each s is assessed with performance measures, such as *accuracy* or *precision* if the task is classification, and R^2 or $root\ mean\ squared\ error$ if the task is regression. Performance measures typically take observed and predicted values as input, $P(Y, \hat{f}_s(X))$. Holding the data X and Y constant, performance measures allow direct comparison of solutions. Generally, the "best" is the one with the best performance, although researchers often value factors in addition to performance.

Given the above description, there are two ways to improve performance in predictive models. First, we try to figure out the $\hat{f}_s(X)$ that we expect to perform best in out-of-sample testing. A solution, then, includes all modeling decisions we make when holding data X and Y constant to allow for comparison of performance measures. This includes the choice of learning algorithm (e.g., logistic regression and random forest), methods for imputing missing data (e.g., list-wise deletion and multiple imputation), handling of outliers (e.g., dropped or down-weighted), dimensionality reduction (e.g., principal component analysis), methods for class imbalance (e.g., over-sampling and case-control methods [13]), approach to cross-validation (e.g., k-fold and rolling forecasts), and variable specification (e.g., transformations and interactions).

The second way to reduce error is to decide on what variables to include in X to begin with. One approach is to not constrain X and include all predictors that are considered reasonable. More predictors increase predictive performance in a given sample, while out-of-sample testing attempts to prevent over-fit solutions that do not generalize beyond the sample. However, a predictive model is one that *emphasizes* the ability to predict Y. This does not mean, necessarily, that we value nothing but the ability to predict Y. For example, researchers may value parsimony, and require that any solution contain no more than 5 or 10 predictors. Others may value theoretical consistency, and thus each X_s is tied to a particular theory.

An AutoML system searches over the solution grid S systematically. An HGML system incorporates domain expertise into that search to identify appropriate modeling decisions that constrain or aid this search. Another way to phrase this is to say that an AutoML system can provide the "best" solution purely with regards to predictive performance. However, the HGML system can optimize performance conditional on models that satisfy the domain expert's quantitative goals and leverage their substantive knowledge, to find solutions that better fit the researcher's vision of "best."

III. Applying HGML to Models of Conflict

Conflict research has seen a growing emphasis on prediction and forecasting models [14, 15]. This includes civil war [16, 17, 18], genocide [6, 19, 20], international conflict [4, 21, 22], terrorism [23, 5, 24], repression [25], protest [8], political regime change [26, 27], and state failure [13, 28]. There is also a literature on forecasting models that emphasize interactions among smaller numbers of actors, often with event data, and often with an emphasis on conflict [29, 30, 31, 32]. Due in large part to its forecasting value, systems have been developed to automatically code event data in near-real time, as well as provide direct API access from R statistical software [33, 34, 35].

One of the reasons why prediction has become more central in the study of conflict is because it forces researchers to focus on overall model performance, rather than individual variable significance [36]. This is important because the field has identified a large number of factors associated with all components of conflict (e.g., onset, duration, and termination) for the different types mentioned above.[1] Most of these factors are relatively minor contributors to model performance, yet their impact is statistically significant in the same way that the impact of other, relatively major contributors, are statistically significant. With an emphasis on model performance, researchers can disentangle the major contributors from the minor ones, and in doing so provide policy makers and analysts with parsimonious models better suited for policy guidance.

[1] A "large number" on the order of hundreds, but not thousands.

Theory testing and comparison is also critically tied to model performance [37]. [21] and [22] test the Kantian Peace using model performance metrics. [8] compare *grievance-based*, *resource mobilization*, *modernization theory*, and *political opportunity* theories. Although not as explicit in their comparison of theory, in forecasting irregular leadership changes [27] identify "thematic models" as a starting point. These include *leader characteristics*, *public discontent*, and *global instability*, among others. [7] test a single hypothesis—states that are neither democratic nor autocratic, but somewhere in the middle, experience more conflict and violence—across many types of conflict. With an emphasis on performance, researchers can test and compare theory directly.

As the the emphasis on forecasting and prediction grows, the set of modeling decisions, referred to as the solution grid S, expands. Consider the two motivating cases above. In the first, we want to identify the predictors that contribute most to the performance of the model. That is, the goal is to identify a parsimonious model with strong out-of-sample performance. One option here is to begin with a large number of variables (X) and cycle through sets, subject to some parsimony constraint (e.g., something as simple as "no more than 5 predictors"), to find the solution that performs best. The already large X becomes even larger when we consider these variables may be transformed or interacted. We might also iterate through imputation methods for missingness, imbalance methods because conflict is rare, approaches to cross-validation since panel data has temporal and spatial dimensions, ways to account for spatial and temporal patterns in the model itself, and choice of learning algorithm since some capture non-linearities and latent dimensions better than others. Thus, researchers are faced with a choice for X and a choice for $\hat{f}(X)$. Further, these are not two independent decisions. Combinatorics suggests this solution grid becomes large, fast.

In the second case, for theory testing and comparison, we might consider choice of X to be much smaller and perhaps even fixed. For example, there may be sets of variables associated with different theories, as in [8] and [27]. Or, a researcher may have a new theory that she wishes to compare to an existing one. Either way, it is still important to search over the solution grid and not fix the learning algorithm or other model decision *a priori*. For example, a researcher might compare her theory to a *grievance-based* theory using a logistic regression. Let's say the results suggests better performance with the grievance theory and no meaningful value added from any of her variables. However, a random forest and a neural network both suggest the researcher's model outperforms the grievance model. This outcome is realistic because both random forests and neural networks are designed to capture non-linearities, while a logistic regression is not. Perhaps the logistic regression was under-specified and the researcher's variables should have been interacted with an omitted variable. Or, the omitted variable might be latent and difficult to operationalize effectively. For example, power and fear are both valuable theoretical concepts that play complex roles in theories of conflict. Capturing such dynamics is the motivation behind the use of a neural network in [38].[2]

It is unrealistic to expect researchers to manually search this grid. Instead, we propose AutoML to search over that solution grid efficiently and systematically, and HGML to incorporate domain expertise into that search to identify appropriate modeling decisions. This approach allows researchers to expand the number of models tested, and to have greater confidence that the model they report is indeed the best performing.

Our approach also provides a meaningful model for comparison, but one that is specified in a different way than the literature currently assumes. [37] discuss the role and value of benchmarks in predictive models, and state, "Such benchmark models can take two forms: they can either reflect the most recent or best-accepted model already established in what we will call a *state-of-the-literature* model, or they can reflect the best model one can specify without relying on theory in what we call a *baseline* model" (p150). While the latter might be reasonably straightforward, at least conceptually, the former is not. What, precisely, is a state-of-the-literature model? Despite the fact that performance measures are numbers where high (or sometimes lower) is better, the numbers are not always comparable. For example, performance is not perfectly comparable when the out-of-sample test set covers a different time period or has different spatial coverage. The parsimony constraint that researchers place on their model can be different, and the important predictors may vary by learning algorithm. In the conflict literature, some models predict the onset of conflict and will typically drop observations of ongoing conflict [39, 40, 41], some maintain the time-series and predict conflict (in which case, a lagged dependent variable performs well as a predictor) [42, 23, 43], and others predict the transition from either peace→conflict or conflict→peace [16, 44, 18]. Recreating models to reflect the design needed for comparison to a state-of-the-literature model may be extraordinarily difficult.

Another issue with the development of a state-of-the-literature model is that such models can become fixtures, despite the fact that they are no longer a reasonable state-of-the-literature model. For example, the model of civil war onset reported in [45] has been used for comparison in [3], [17], and [46]. To be fair, these papers use the Fearon and Laitin model appropriately and advance our understanding of conflict modeling, but at the same time they demonstrate how a state-of-the-literature model can become a fixture. We should also keep in mind that the Fearon and Laitin model is not a fixture because of its out-of-sample performance; it is a fixture because the model represents an attractive theory. If our emphasis is on predictive modeling, then our benchmarks should also emphasize prediction.

[2] As [38] state with respect to international conflict, "many qualitative researchers expect the relationships to be highly nonlinear, massively interactive, and heavily context dependent or contingent. Because these characteristics would be missed with standard statistical approaches, particularly the typical linear-normal models imported from studies of American politics, we adopt a form of the highly flexible *neural network model*. This type of model is well suited to data with complex, nonlinear, and contingent relationships" (p22).

The use of AutoML to search the solution grid, and HGML to incorporate domain knowledge, enables researchers to identify benchmarks for comparison that are guaranteed to perform well. Researchers can more easily set the modeling decisions they want to hold constant, and allow other decisions to vary, without the restriction of a state-of-the-literature model where those decisions may have already been set. This is not to say that we should blindly accept modeling decisions provided by an automated system as a benchmark. But, if the system identifies transformations or interactions as powerful out-of-sample predictors, then a researcher should consider whether this predictor is sensible. If it is, then there is no reason that should be excluded from a benchmark. Thus, there is a need for an AutoML component and an HGML component. In the following section we describe the system integration and the points of contact between the front-end HGML system and backend AutoML learners.

IV. System Description and Integrations

The HGML system described here is the latest application developed as part of the TwoRavens project [47, 48]. These applications share a common codebase (github.com/tworavens), and a broader goal of facilitating quantitative reasoning for domain experts, primarily in the social sciences.

In our applications, the bulk of the data and the intensive processing are all done on the back-end, while users interact with a Web application. In this particular instance, we describe an application that has been built as part of DARPA's Data-Driven Discovery of Models (hereafter D3M) program [49]. An underlying motivation of this program is to "automate the data scientist," which is to say build a system that enables subject matter experts to harness the power of machine learning to solve their quantitative problems. Specifically, these are individuals with domain expertise, including those who have a deep understanding of the foundational theories in their field, but whose statistical expertise is not in machine learning. We expect such users can describe their problem and data, but perhaps have difficulty identifying the many modeling decisions that then comprise the solution grid.

The underlying purpose of our system is to connect domain experts with solutions to their machine learning problems. AutoML learners provide such solutions, and in fact automate many of the modeling decisions. However, as argued in [11], there is a need for HGML systems to incorporate domain expertise into the automated search.

A. Front-end HGML Components

Here, we describe some of the core front-end HGML components that comprise the TwoRavens Application. This includes our three base *modes*: Explore, Model, and Results. Each mode is intended to elicit information from users, and provide information that helps users specify their problem and find appropriate solutions. In our application, these modes can be thought of as comprised of the more granular HGML components described in [11].

To provide a sense for the application, Figure 1 is a compilation of pieces from the Model and Results mode.

1) Explore: The Explore mode is intended to facilitate simple and descriptive data visualizations. Users select features of interest, and the system draws an appropriate visualization based on the data. For example, a user may select two quantitative variables, e.g., *GDP per capita* and the *infant mortality rate*, and the system returns a scatterplot. If one of those features were nominal instead of quantitative, e.g., *geographic region*, then the system would return a box and whiskers plot.

Explore leverages Vega-lite [50], a grammar built on D3.js [51]. Doing so allows our system to easily visualize diverse types of data. For example, on ingest we automatically characterize the data in abstractions, such as by denoting *quantitative* or *nominal*. Then, after a user selects the variables to visualize, we map the Vega-lite types to an appropriate visualization. The variable types are stored in the metadata, and users have the ability to refine these types.

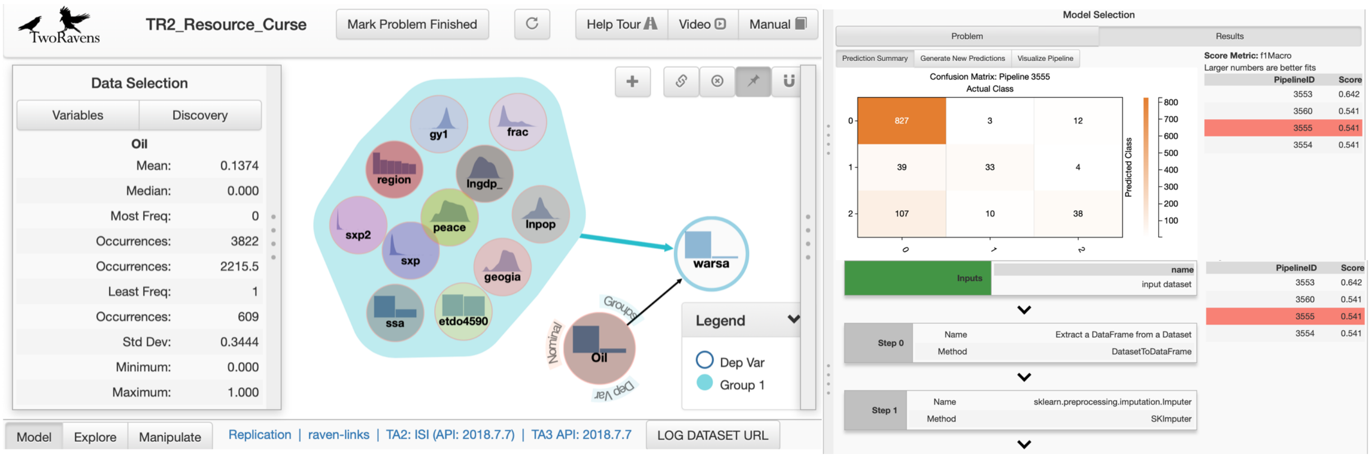
2) Model: Our Model mode is where users specify the many inputs to the AutoML search. This includes the task type (e.g., classification, regression, timeseriesForecasting), task subtype (e.g., binary or multiclass), and the performance metric to optimize on. Users also specify the features to search over and prior knowledge about these variables (eg. ordered or categorical). In Model mode, users can manipulate the data to form new features, such as by taking the natural log of a feature or multiplying two features together. These kinds of transformations are common in quantitative social science. For example, in the model we replicate from [5] there are three natural logs and three interaction terms.

Model mode also includes the *problem discovery* feature, which is a set of problems that our system has identified as potentially of interest. In this case, a problem is defined as a combination of target variable, predictors, data manipulations, and performance measure. Like Explore, this feature is intended to help users sift through datasets to find relationships of interest. It is the user's domain expertise that is required to shape the specific problem to submit to the AutoML learners.

3) Results: Finally, Results mode allows users to explore solutions found by the back-end learners. In addition to the performance metric, the system extracts predictions for all observations in training data. These predictions are visualized, for example as a Predicted versus Actual scatterplot in the case of regression, and a confusion matrix in the case of classification, and as empirical first differences and partial effects plots. Examples of graphical interpretions of resulting ML models are shown in the right of figure 1 and in 2. In the case of DARPA D3M learners, the primitive components that have been used to formulate the solution are described. For example, if the learner does one-hot encoding for all nominal variables, this is included in the solution description.

B. Back-end AutoML Learners

While the front-end extracts domain knowledge and guidance from the user, this is converted into a task-oriented



Oil

Mean:	0.1374
Median:	0.000
Most Freq:	0
Occurrences:	3822
Occurrences:	2215.5
Least Freq:	1
Occurrences:	609
Std Dev:	0.3444
Minimum:	0.000
Maximum:	1.000

Confusion Matrix: Pipeline 3555

Actual Class

	0	1	2
0	827	3	12
1	39	33	4
2	107	10	38

Score Metric: f1Macro

Larger numbers are better fits

PipelineID	Score
3553	0.642
3560	0.541
3555	0.541
3554	0.541

Inputs	name
	input dataset

Step 0		
	Name	Extract a DataFrame from a Dataset
	Method	DatasetToDataFrame

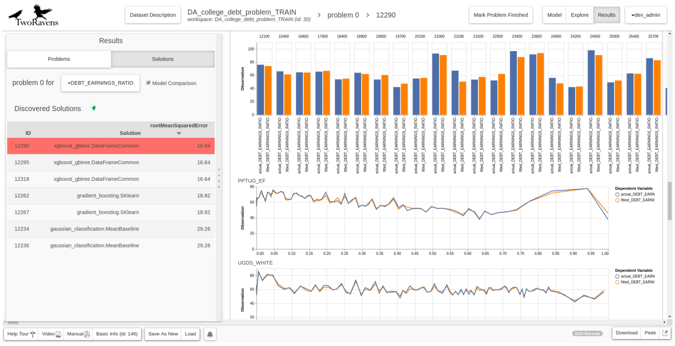
Step 1		
	Name	sklearn.preprocessing.imputation.Imputer
	Method	SKImputer

PipelineID	Score
3553	0.642
3560	0.541
3555	0.541
3554	0.541

Fig. 1. TwoRavens Web Application

API call to a number of AutoML systems run on the back-end. Our API is configured after the D3M API (openly available as gRPC at [52]) developed as a general solution for communication between user interfaces and AutoML systems. While this API can directly call all D3M AutoML engines, we have constructed additional wrappers so that is can also call all the other AutoML systems in our trials. We briefly describe the systems we used in our back-end in turn.

1) *Data Driven Discovery of Models (D3M) systems:* The DARPA D3M project has spearheaded development of ten competing AutoML systems by a diverse set of research groups [53, 54, 55, 56], each sharing a common library of algorithms. As an exemplar of these, we use the Alpine Meadow AutoML engine [57] produced as part of the Northstar data science system [58].
2) *Auto-sklearn* wraps the Sklearn framework and scikit-learn estimators to automatically find solutions to machine learning problems (pipelines) created using the elements of those libraries [10]. For this search, it uses Bayesian optimization utilizing the tree-based approach of the SMAC algorithm [59].
3) *TPOT* (short for Tree-Based Pipeline Optimization Tool) uses a genetic programming approach to construct machine learning pipelines, using the scikit-learn estimators as well as its own extensions [60, 61].
4) *H2O* uses a Java developed machine learning library including random forests, GBMs, GLMs and neural nets to construct solutions, and then additionally composes ensembles of these [62].
5) *MLBox* contains a Python library of ML algorithms oriented around sklearn [63].
6) *mljar* generates ensembles from eight ML algorithms, including various versions of gradient boosting, random forests, as well as KNN and neural nets, to create predictions for either regression problems or binary classification [64].



ID	Solution	rootMeanSquaredError
12290	xgboost_gbtree.DataFrameCommon	16.64
12295	xgboost_gbtree.DataFrameCommon	16.64
12318	xgboost_gbtree.DataFrameCommon	16.64
12262	gradient_boosting.SKlearn	18.82
12267	gradient_boosting.SKlearn	18.82
12234	gaussian_classification.MeanBaseline	29.26
12236	gaussian_classification.MeanBaseline	29.26

Fig. 2. TwoRavens Model Interpretation Plots

While there are stark differences between these systems in the way they search for solutions, there is heavy overlap in the libraries of data cleaning and ML algorithms they are searching over. Thus it is an empirical question whether their different search approaches lead to to divergent model predictions, or instead whether any well-functioning AutoML system is reasonably interchangeable with another. This is a core question we attempt provide insight on.

C. An Integration Approach

The core TwoRavens Application consists of an HGML front-end and multiple services which allow a user to explore data as well as model and discover problems. The core application consists of multiple containers including:

- Web Application, which provides the user with a rich user interface and orchestrates communication between services. A key feature is a persisted workspace data structure consisting of a user-specific problem space: the dataset, summary metadata, discovered problems, and specific solutions/pipelines for these problems.
- Internal statistical services which include preprocessing, automated problem discovery, and a variety of other functions.
- A persistent SQL database, which is used to track user accounts and workspaces.
- MongoDB, which is used to provide timely data transformations, subsets, and views of the dataset being explored.

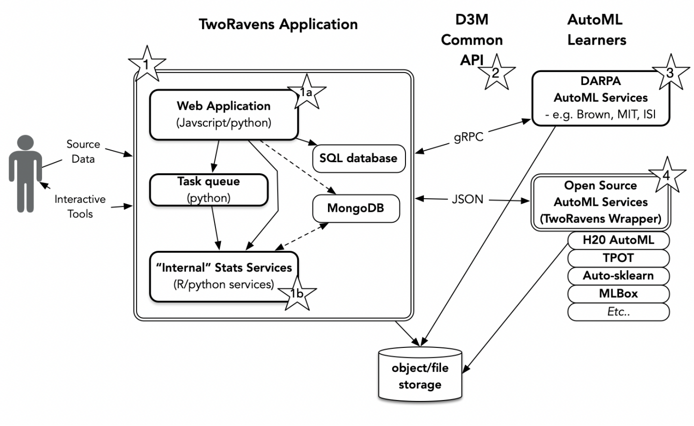


Fig. 3. TwoRavens System Diagram

> This data is not persisted over the long-term but may be created using the metadata saved in a user's workspace.

As part of DARPAs Data-Driven Discovery of Models (D3M) program, a **D3M Common API** was developed to allow consistent communication with a variety of AutoML learners. Given a problem via the API, the AutoML learners will search for solutions and then describe, fit, and score these solutions.

The D3M Common API gives the ability to communicate with multiple AutoML learners developed under the D3M program, including those created by Brown University [57], the USC Information Sciences Institute [54], and MIT [56]. Collectively, we refer to this class of AutoML learner as **DARPA D3M AutoML Services**.

The API requests originate in TwoRavens as JSON and are sent to the AutoML learners via gRPC, a Google-created remote procedure call (RPC) framework that allows streaming responses. When received, these responses are transformed back into JSON and consumed by the TwoRavens system.

Ultimately these responses provide the user with solutions that include prediction and score summaries, measures of variable importance (empirical first differences and partial least squares), and pipeline visualizations. In addition, the application includes the ability to compare solutions.

Leveraging the D3M Common API, we have created the **TwoRavens wrapper**, which allows the use of diverse open source AutoML services. This wrapper allows the core TwoRavens application to call the open-source AutoML learners with the same API requests used for the D3M AutoML learners. Similarly, the responses from each open-source AutoML library are mapped to the D3M Common API format and consumed by the TwoRavens application.

The open source backend AutoML learners include H20 AutoML, TPOT, and Auto-sklearn. It is notable that only the JSON format, not the gRPC format, of the commands is used.

V. Results from Three Automated Selections

We experiment with three conflict models from the peer-reviewed literature [3, 4, 5], each representing a different type of armed conflict (international conflict, civil conflict, and transnational terrorism). Using our HGML system, we obtain solutions from a diverse set of AutoML learners to solve the same problem as in the original research articles.

It is important to note that AutoML is intentionally designed so that S contains nonsense solutions that perform well. For example, a solution might contain an interaction of mountainous terrain and IO memberships, a relationship that has no theoretically grounding. It is not reasonable to constrain the search to avoid all nonsense solutions—they should simply be discarded by the domain expert. The point of a large S is that many sensible solutions do exist, and in fact far more than we can think to test manually, and so we seek to identify the best solutions and use domain expertise to select among them.

A. System Requirements

The system requires a dataset, a dataset specification document, and a problem specification document. The schemas used for the dataset and problem documents have been developed as part of DARPA's Data-Driven Discovery of Models program [49]. Full descriptions of the schemas can be found at gitlab.com/datadrivendiscovery/. The data is split into training data and testing data, which in this early stage was done manually to parallel the split in the published papers.

The dataset schema itself is organized in two fields: *about* and *dataResources*. The *about* field contains items for names (e.g., an ID and a user-friendly name), versions, file sizes, and licensing. The *dataResources* field requires information about each data resource, which in the case of conflict data each resource is likely a data table, and often there is only one. For each table, *dataResources* contains information on the path, the format (e.g., csv), and column-level information that includes an index, a column name, a column type (e.g., "integer" and "string"), and a role (e.g., "index", "attribute", and "suggestedTarget").

In the problem document, the highest-level fields include *about*, *inputs*, and *expectedOutputs*. The *about* field includes names and versions, but also the problem type and subtype. For our purposes, problem type is one of "classification" or "regression," but can take on other values such as "timeseries-Forecast" and "linkPrediction," both of which are applicable to conflict modeling. The allowable problem subtypes is conditional on the type, and for our purposes the most useful distinction is "binary" or "multiClass" for the "classification" type. The *inputs* field describes the target variable and where to find it. It also describes the way the training and testing data was split, and the desired performance metrics to compute for each solution (e.g., "rocAuc" or "accuracy"). The *expectedOutputs* field specifies where to write the predictions for each solution.

B. Preparations

For a meaningful assessment of our system, it is important to match the data, dataset document, and problem document to the original publication. For each paper, we obtained replication data and code. We then processed the data and built dataset and problem documents to reflect the precise comparisons.

[3] forecasts three types of political instability—*civil war*, *adverse regime change*, and *genocide/politicide*—two years in advance. The structure of the target variable is onset, which means they forecast the likelihood that a country will have some type of political instability two years out. Predictions are made at the country-year level of analysis. They use a case selection method where they have all cases of instability in their sample and select cases of no instability at a ratio of 1:2. This is a binary classification problem, and the performance metric is accuracy. Although they conduct a true out-of-sample test, which includes all country-years in the international system, the data to do so are not included in their replication, so instead we compare our cross-validation performance to their in-sample performance reported in the "Full Problem Set" model in Table 1 (p195). Specifically, their reported accuracy is 81.4%, with a recall of 80.3% and a specificity of 81.8%.

[4] forecast international conflict, and are particularly interested in the role of contentious issues. The target variable is structured to represent the onset of conflict and predictions are made at the dyad-year level of analysis, where a dyad is a pair of countries. We prepared the data to include all variable transformations (e.g., *peace years* polynomials) that are present in their combined model. We also split the data so that the training set was pre-1990, and the testing set ranged from 1990-2001 using covariate values from 1989 (p20). This is also a binary classification problem, with the performance metric here being the area under the receiver operating characteristic curve (AUC). The in-sample AUC is .92, while the out-of-sample is .90 (p23-24). The model we replicate here is the *Combined Model* in Table I on page 22.

[5] predict instances of transnational terrorism with a focus on the impact of *democracy* as a predictor. Their binary classification problem is conducted at the dyad-year level of analysis. They use a 10-fold cross-validation to estimate out-of-sample performance, and use primarily AUC as the performance measure. This is also a binary classification problem where the performance metric is rocAuc. The model we replicate is reported in Table 1 on page 27. It has an in-sample AUC of .88 and an out-of-sample AUC of .89 (p29).

In each of the three cases, we prepared the data to include all variables present in the model we replicate. We also split the data so that our training data is identical to the training data in the published paper. While we were able to produce perfect replications for [4] and [3], our replication is not identical for [5]. However, despite minor differences in regression estimates and a 3.2% increase in the number of observations, the inferences drawn are the same so we proceed with our comparison.

C. Results

Table I contains the results for the three conflict models from six different AutoML learners. As a point of comparison, we use the prediction performance metric reported by the original study authors, assuming that their subject matter expertise on the quantity of interest to report guides these choices. The number in parenthesis contains the difference from the reported metric of the author's published model in each paper. Results in **bold** are the leading solution to each problem. Given the nature of how the AutoML systems differ, it is not always possible to generate predictions of the required nature or on the required split for correct comparability, so some systems are omitted from some comparisons.

We find that the HGML system slightly outperforms the original expert created forecasting model in each of our datasets, and essentially create high performing forecasts, with no programming or data science expertise required. The best performing AutoML engine varied by problem, encouraging us in our strategy of searching across multiple AutoML engines, although they were often closely comparable in the performance metric.

TABLE I
RESULTS FROM SIX AUTOML LEARNERS

	AutoML		
	Learner	Performance	Timing
Goldstone et al. (Accuracy=0.814) In Sample	Auto-sklearn	0.880 (+0.066)	30s
	H2O	0.971 (+0.157)	32s
	TPOT	0.837 (+0.023)	9s
	MLBox	**1.000** (+0.186)	28s
	mljar	0.830 (+0.016)	371s
	Alpine Meadow (D3M)	0.929 (+0.115)	1800s
Gleditsch & Ward (AUC=0.90) Out-of-sample	Auto-sklearn	0.50 (-0.40)	600s
	H2O	0.94 (+0.04)	164s
	MLBox	**0.95** (+0.05)	531s
	mljar	0.94 (+0.04)	700s
Gelpi & Avdan (AUC=0.89) 10-fold Cross-val	Auto-sklearn	0.50 (-0.39)	3600s
	H2O	0.95 (+0.06)	3600s
	MLBox	0.95 (+0.06)	3600s
	mljar	**0.97** (+0.08)	3600s
	Alpine Meadow (D3M)	0.54 (-0.35)	3600s

In related work benchmarking Auto-sklearn, H20 and TPOT, the authors of [65] find that TPOT performs well in regression models and auto-sklearn outperforms in classification tasks. Our results show that those findings do not hold in the above tasks when coupled to our HGML system.

We present some simple timing numbers but note that timing is difficult to compare across systems. Some stream results as discovered, while others complete a search before returning all results. Some use an upper time bound that we tuned be large enough to still return the best result we had witnessed from that system. In the Gelpi and Avdan dataset we simply gave all systems an hour.

VI. CONCLUSION AND FUTURE DIRECTIONS

We have described AutoML and HGML, with applications to conflict modeling. Using the D3M Common API, we have integrated our front-end system with a diverse set of back-end AutoML learners. Then, we analyzed results using three models from the literature on conflict forecasting. In each of these we found solutions that performed better than the original, given the authors desired performance metric.

In this initial test, we intentionally limited the system from exploring the bulk of the solution space. For example, in our tests the data was prepared identically to the data in the publications. This means many of the modeling decisions, such as Goldstone et al.'s case control method, Gleditsch and Ward's use of 1989 covariate values for all out-of-sample testing, and Gelpi and Avdan's use of a dummy variable to indicate a transnational attack instead of an integer value for the number of attacks, have already been fixed. Future tests should reduce the number of fixed decisions, allowing more variance on modeling decisions to fill the solution grid.

Additionally, in future work we anticipate expanding the data and problem types for experimentation. Here, we limited ourselves to models of conflict to focus on a particular phenomena that is of substantive interest for a broader community of subject matter experts including policy makers, and commonly uses predictive models. We opted for variation on the type of conflict (international and civil wars, and transnational terrorism) and structure of the data (country-year and dyad-year), but many other data and problem types are also relevant to conflict modeling and could have been explored. For example, [44] models conflict transitions, not just conflict onsets. [31] model a time-series of events. [66] disaggregate the unit of analysis spatially and temporally. Many other predictive models represent very different data and problem structures, but such an exhaustive analysis is left for future research.

Acknowledgment

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement number FA8750-17-2-0114. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. We would like to thank Daniel Garijo, Yolanda Gil, Shikhar Gupta, Pedro Szekely and Emanuel Zgraggen for valuable discussions throughout this project.

Appendix
List of Features for Conflict Models

The features listed here are those in each of the three conflict models, as described in the results tables from those papers. In our analysis we used only these features and excluded any others in the data.

Gelpi and Avdan, Table 1, Model 1, page 27:

- Polity Score Target, Polity Score Origin, Target GDP (logged), Target Logged Population, Target Major Power, Origin Major Power, Colonial Tie, Ethnic Tie, Ethnic X Post Cold War, Ethnic X Post-911, Dyadic Alliance, Alliance X Post Cold War, Dyadic Rivalry, Post-Cold War Era, Post-911 Era, Log of Distance, Peace Years Spline 1, 2, 3, and 4

Gleditsch and Ward, Table 1, Combined Model, page 22:

- Previous MID, peaceyears, peaceyears^2, peaceyears^3, Territorial claim, River claim, Maritime claim, Peaceful settlement attempt - territorial, Peaceful settlement attempt - river, Peaceful settlement attempt - maritime, Lower democracy score, Balance ratio, ln(distance)

Goldstone et al., Table 1, Full Problem Set, page 195:

- Partial Autocracy, Partial Democracy with Factionalism, Partial Democracy without Factionalism, Full Democracy, Infant Mortality, Armed Conflict in 4+ Bordering States, State-Led Discrimination

References

[1] H. Wallach, "Computational social science $\neq$ computer science + social data," *Communications of the ACM*, vol. 61, no. 3, pp. 42–44, 2018.

[2] G. King, "Big data is not about the data!" in *Computational social science: Discovery and prediction*, R. M. Alvarez, Ed. Cambridge University Press, 2016.

[3] J. A. Goldstone, R. H. Bates, D. L. Epstein, T. R. Gurr, M. B. Lustik, M. G. Marshall, J. Ulfelder, and M. Woodward, "A global model for forecasting political instability," *American Journal of Political Science*, vol. 54, no. 1, pp. 190–208, 2010.

[4] K. S. Gleditsch and M. D. Ward, "Forecasting is difficult, especially about the future: Using contentious issues to forecast interstate disputes," *Journal of Peace Research*, vol. 50, no. 1, pp. 17–31, 2013.

[5] C. Gelpi and N. Avdan, "Democracies at risk? a forecasting analysis of regime type and the risk of terrorist attack," *Conflict Management and Peace Science*, vol. 35, no. 1, pp. 18–42, 2018.

[6] B. E. Goldsmith and C. Butcher, "Genocide forecasting: Past accuracy and new forecasts to 2020," *Journal of Genocide Research*, vol. 20, no. 1, pp. 90–107, 2018.

[7] Z. M. Jones and Y. Lupu, "Is there more violence in the middle?" *American Journal of Political Science*, vol. 62, no. 3, pp. 652–667, 2018.

[8] E. Chenoweth and J. Ulfelder, "Can structural conditions explain the onset of nonviolent uprisings?" *Journal of Conflict Resolution*, vol. 61, no. 2, pp. 298–324, 2017.

[9] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "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*. ACM, 2013, pp. 847–855.

[10] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2962–2970. [Online]. Available: http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf

[11] Y. Gil, J. Honaker, S. Gupta, Y. Ma, V. D'Orazio, D. Garijo, S. Gadewar, Q. Yang, and N. Jahanshad, "Towards human-guided machine learning," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, ser. IUI '19, 2019.

[12] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[13] R. Kennedy, "Making useful conflict predictions: Methods for addressing skewed classes and implementing cost-sensitive learning in the study of state failure," *Journal of Peace Research*, vol. 52, no. 5, pp. 649–664, 2015.

[14] V. D'Orazio, "Conflict forecasting and prediction," , unpublished.

[15] L.-E. Cederman and N. B. Weidmann, "Predicting armed conflict: Time to adjust our expectations?" *Science*, vol. 355, no. 6324, pp. 474–476, 2017.

[16] D. Chiba and K. S. Gleditsch, "The shape of things to come? expanding the inequality and grievance model for civil war forecasts with event data," *Journal of Peace Research*, vol. 54, no. 2, pp. 275–297, 2017.

[17] D. Muchlinski, D. Siroky, J. He, and M. Kocher, "Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data," *Political Analysis*, vol. 24, no. 1, pp. 87–103, 2016.

[18] H. Hegre, J. Karlsen, H. M. Nygård, H. Strand, and H. Urdal, "Predicting armed conflict, 2010–2050," *International Studies Quarterly*, vol. 57, no. 2, pp. 250–270, 2013.

[19] B. E. Goldsmith, C. R. Butcher, D. Semenovich, and A. Sowmya, "Forecasting the onset of genocide and politicide: Annual out-of-sample forecasts on a global dataset, 1988–2003," *Journal of Peace Research*, vol. 50, no. 4, pp. 437–452, 2013.

[20] N. Rost, "Will it happen again? on the possibility of forecasting the risk of genocide," *Journal of Genocide Research*, vol. 15, no. 1, pp. 41–67, 2013.

[21] S. J. Cranmer, E. J. Menninga, and P. J. Mucha, "Kantian fractionalization predicts the conflict propensity of the international system," *Proceedings of the National Academy of Sciences*, vol. 112, no. 38, pp. 11 812–11 816, 2015.

[22] M. D. Ward, R. M. Siverson, and X. Cao, "Disputes, democracies, and dependencies: A reexamination of the kantian peace," *American Journal of Political Science*, vol. 51, no. 3, pp. 583–601, 2007.

[23] B. A. Desmarais and S. J. Cranmer, "Forecasting the locational dynamics of transnational terrorism: A network analytic approach," *Security Informatics*, vol. 2, no. 1, p. 8, 2013.

[24] W. Enders, Y. Liu, and R. Prodan, "Forecasting series containing offsetting breaks: Old school and new school methods of forecasting transnational terrorism," *Defence and Peace Economics*, vol. 20, no. 6, pp. 441–463, 2009.

[25] D. W. Hill and Z. M. Jones, "An empirical evaluation of explanations for state repression," *American Political Science Review*, vol. 108, no. 3, pp. 661–687, 2014.

[26] A. Beger, C. L. Dorff, and M. D. Ward, "Irregular leadership changes in 2014: Forecasts using ensemble, split-population duration models," *International Journal of Forecasting*, vol. 32, no. 1, pp. 98–111, 2016.

[27] ——, "Ensemble forecasting of irregular leadership change," *Research & Politics*, vol. 1, no. 3, p. 2053168014557511, 2014.

[28] G. King and L. Zeng, "Improving forecasts of state failure," *World Politics*, vol. 53, no. 4, pp. 623–658, 2001.

[29] J. C. Pevehouse and J. S. Goldstein, "Serbian compliance or defiance in kosovo? statistical analysis and real-time predictions," *Journal of Conflict Resolution*, vol. 43, no. 4, pp. 538–546, 1999.

[30] P. A. Schrodt and D. J. Gerner, "Cluster-based early warning indicators for political change in the contemporary levant," *American Political Science Review*, vol. 94, no. 4, pp. 803–817, 2000.

[31] P. T. Brandt, M. Colaresi, and J. R. Freeman, "The dynamics of reciprocity, accountability, and credibility," *Journal of Conflict Resolution*, vol. 52, no. 3, pp. 343–374, 2008.

[32] T. Zeitzoff, "Using social media to measure conflict dynamics: An application to the 2008–2009 gaza conflict," *Journal of Conflict Resolution*, vol. 55, no. 6, pp. 938–969, 2011.

[33] M. Solaimani, R. Gopalan, L. Khan, P. T. Brandt, and B. Thuraisingham, "Spark-based political event coding," in *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2016, pp. 14–23.

[34] H. Kim, V. D'Orazio, P. Brandt, J. Looper, S. Salam, L. Khan, and M. Shoemate, "Utdeventdata: An r package to access political event data," *The Journal of Open Source Software*, vol. 4, p. 1322, 2019.

[35] A. K. Gunasekaran, M. B. Imani, L. Khan, C. Grant, P. T. Brandt, and J. S. Holmes, "Sperg: Scalable political event report geoparsing in big data," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2018, pp. 187–192.

[36] M. D. Ward, B. D. Greenhill, and K. M. Bakke, "The perils of policy by p-value: Predicting civil conflicts," *Journal of Peace Research*, vol. 47, no. 4, pp. 363–375, 2010.

[37] S. J. Cranmer and B. A. Desmarais, "What can we learn from predictive modeling?" *Political Analysis*, vol. 25, no. 2, pp. 145–166, 2017.

[38] N. Beck, G. King, and L. Zeng, "Improving quantitative studies of international conflict: A conjecture," *American Political Science Review*, vol. 94, no. 1, pp. 21–35, 2000.

[39] H. Mueller and C. Rauh, "Reading between the lines: Prediction of political violence using newspaper text," *American Political Science Review*, vol. 112, no. 2, pp. 358–375, 2018.

[40] N. Rost, G. Schneider, and J. Kleibl, "A global risk assessment model for civil wars," *Social Science Research*, vol. 38, no. 4, pp. 921–933, 2009.

[41] B. Harff, "No lessons learned from the holocaust? assessing risks of genocide and political mass murder since 1955," *American Political Science Review*, vol. 97, no. 1,

pp. 57–73, 2003.

[42] T. Chadefaux, "Early warning signals for war in the news," *Journal of Peace Research*, vol. 51, no. 1, pp. 5–18, 2014.

[43] A. Basuchoudhary, J. T. Bang, T. Sen, and J. David, *Predicting Hotspots: Using Machine Learning to Understand Civil Conflict*. Rowman & Littlefield, 2018.

[44] H. Hegre, H. M. Nygård, and R. F. Ræder, "Evaluating the scope and intensity of the conflict trap: A dynamic simulation approach," *Journal of Peace Research*, vol. 54, no. 2, pp. 243–261, 2017.

[45] J. D. Fearon and D. D. Laitin, "Ethnicity, insurgency, and civil war," *American political science review*, vol. 97, no. 1, pp. 75–90, 2003.

[46] M. Colaresi and Z. Mahmood, "Do the robot: Lessons from machine learning to improve conflict forecasting," *Journal of Peace Research*, vol. 54, no. 2, pp. 193–214, 2017.

[47] J. Honaker and V. D'Orazio, "Statistical modeling by gesture: A graphical, browser-based statistical interface for data repositories," in *Extended Proceedings of the 25th ACM Conference on Hypertext and Social Media*, ser. DataWiz 2014, vol. 1210. ACM, 2014. [Online]. Available: http://ceur-ws.org/Vol-1210/datawiz2014_05.pdf

[48] V. D'Orazio, M. Deng, and M. Shoemate, "Tworavens for event data," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, 2018, pp. 394–401.

[49] W. Shen, "Data-driven discovery of models (d3m)," *Defense Advanced Research Projects Agency, Arlington, VA*, 2016.

[50] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 341–350, 2016.

[51] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[52] [Online]. Available: https://gitlab.com/datadrivendiscovery/ta3ta2-api

[53] A. Santos, S. Castelo, C. Felix, J. P. Ono, B. Yu, S. Hong, C. T. Silva, E. Bertini, and J. Freire, "Visus: An interactive system for automatic machine learning model building and curation," *arXiv preprint arXiv:1907.02889*, 2019.

[54] Y. Gil, K.-T. Yao, V. Ratnakar, D. Garijo, G. V. Steeg, P. Szekely, R. Brekelmans, M. Kejriwal, F. Luo, and I.-H. Huang, "P4ml: A phased performance-based pipeline planner for automated machine learning," in *Proceedings of Machine Learning Research, ICML 2018 AutoML Workshop*, 2018. [Online]. Available: http://dgarijo.com/papers/widoco-iswc2017.pdf

[55] M. Milutinovic, A. G. Baydin, R. Zinkov, W. Harvey, D. Song, F. Wood, and W. Shen, "End-to-end training of differentiable pipelines across machine learning frameworks," 2017.

[56] T. Swearingen, W. Drevo, B. Cyphers, A. Cuesta-Infante, A. Ross, and K. Veeramachaneni, "Atm: A distributed, collaborative, scalable system for automated machine learning," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 151–162.

[57] Z. Shang, E. Zgraggen, B. Buratti, F. Kossmann, P. Eichmann, Y. Chung, C. Binnig, E. Upfal, and T. Kraska, "Democratizing data science through interactive curation of ml pipelines," in *Proceedings of the 2019 International Conference on Management of Data*. ACM, 2019, pp. 1171–1188.

[58] T. Kraska, "Northstar: An interactive data science system," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 2150–2164, 2018.

[59] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *International conference on learning and intelligent optimization*. Springer, 2011, pp. 507–523.

[60] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd, and J. H. Moore, *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I*. Springer International Publishing, 2016, ch. Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pp. 123–137. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-31204-0_9

[61] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, ser. GECCO '16. New York, NY, USA: ACM, 2016, pp. 485–492. [Online]. Available: http://doi.acm.org/10.1145/2908812.2908918

[62] A. Candel, V. Parmar, E. LeDell, and A. Arora, "Deep learning with h2o," *H2O. ai Inc*, 2016.

[63] A. Aronio De Romblay. [Online]. Available: https://mlbox.readthedocs.io/en/latest/

[64] [Online]. Available: https://docs.mljar.com

[65] A. Balaji and A. Allen, "Benchmarking automatic machine learning frameworks," *arXiv preprint arXiv:1808.06492*, 2018.

[66] H. Hegre, M. Allansson, M. Basedau, M. Colaresi, M. Croicu, H. Fjelde, F. Hoyles, L. Hultman, S. Högbladh, R. Jansen *et al.*, "Views: A political violence early-warning system," *Journal of Peace Research*, p. 0022343319823860, 2019.