Comparing Sway Custom Clustering and Explanation vs "Off the Shelf" Commercial Models

Greg Tystahl School of Computer Science
North Carolina State University
Raleigh, North Carolina 27606
Email: gttystah@ncsu.edu

***Abstract*—This paper is a comparison study between the state-of-the-art semi-supervised multi-objective recursive clustering model vs a semi-supervised multi-objective bisecting KMeans clustering model. The original Sway1 model was able to achieve steady results with low sample and explanation tax (-0.26 and -0.10 respectively (Normalized)) and low evaluations. The Sway2 model, which is the second model mentioned above, had much more variance and fell behind Sway1 with a sample tax of -0.31 and explanation of 0. Sway2 was put to the test with 4 different case studies: Budget Test, February Study, Ablation Study, and HPO study. These tests proviced the ground work on where to improve for future work by looking deeper into the tradeoff of higher budget cost to filtering percentage and evaluations.**

I. Introduction

For decades, people have been trying to interpret and gather insight from data. As technology continues to improve, so does our ability to gather larger amounts of data. Not only are we able to gather large amount of data, but we are able to parse through that data rather quickly. This has given rise to techniques such as Neural Networks that, with enough computing power, allows us to pull information from very large data sets that gives us humans insight to make informed decisions on future tasks using that data [4].

There are some problems with this though. Even if we are able to create these tools, they take a lot of time and computing power to train. This training process becomes tedious and costly if the model needs to be retrained constantly. This huge cost has led some to speculate that there must be some way to reduce this cost [6]. Beyond just the cost, some of these models are referred to as a "black box" in the sense that the creators are unable to extract the information that influences the decisions of the model [8]. This is another downfall of present day models that needs to be solved.

Not all is as bleak as it seems however. There have been recent work that has been looking into the relationship between data points within a data set. It has been proven by Chen et al. [6] that the attributes (Xs) values of a data set that evaluate to a good outcomes (Ys) are close together within their respective data set. They called their methodology "Sway" which is short for "The sampling way". Their research begged the question: If the data points and their outputs are so similar, do we need to evaluate all of these data points to get the same answers? As it turns out, it is the case that by clustering data in the X space, we can reduce the number of needed evaluations significantly by only picking out of this data set the "good" data points with limited evaluations to cluster. This paper made major improvements to the fishing problem, which is finding the best data points in a data set with a lot of different values by finding the best spots to proverbially fish at.

With Sway being a huge step in the efficiency direction, we wanted to know what is the current state of "Off the Shelf" clustering models and how do they compare to the clustering algorithm used in Sway. Most people who have tried to experiment with machine learning as well as statistical analysis have found themselves using the popular python module sklearn [3]. Within this library, there are lots of machine learning and clustering models that users can experiment with and change. These models have been fine tuned to fit as many different cases as possible, but also have lots of room for customization.

For this study, we chose to use the BisectKMeans and RandomForest models for the underlying clustering and explanation models. We opted to keep the default hyper parameters of the underlying tool as they are, and only changed them for specific tests that required them (Budget tests being one of them). We dubbed this new model Sway2, as it is a new, reproduced version of the original with some variation. Since within Sway there are a lot of hard coded hyper parameters, we decided to compare the baseline bisecting cluster model against the sway unoptimized implementation. This had a two fold benefit because it not only offered a closer evaluation, but also an easily reproducible form of the Sway methodology in general.

While we kept the model with its defaults which are calibrated to handle a plethra of different implementations, we have found some concerning results. Sway2 can come close in terms of medians and averages of the found "good" data and reduces the number of needed evaluations in all cases, the number of data included in those groups is extremely higher than with Sway. For instance, in Table XVII, the number of items in the "good" group is 52981, only reducing from the original 53662 by 681 evaluations. This was not at all surprising based on the difference in the expelling of the rest of the data in comparison to Sway1 which will be explained in more detail later. With the number of items in the good set being as high as they were, we have come to the conclusion that the models in sklearn are limiting and would require quite a bit of work and tweaking to get as good of performance as the base Sway model. Ideas for future improvement will be discussed later in the discussion section.

A. Structure of This Paper

This paper aims to answer 2 research questions:

- Can the current "Off the Shelf" clustering model reduce the number of needed evaluations?
- How does the re-imagined model compare to the original Sway?

We found that even if in the worst case as seen in XVII below, Sway2 with the sklearn models incorperated were able to reduce the space of needed evaluation and nearly boolean dominated the baseline every time (All except for Table XVII again). This is significant due to the issue of reproducibility [9]. If users understand the principles of Sway, but do not have access to the working implementation of it, even the default values of sklearn models with no changes can do better with minimal evaluations to reduce the input space It will not be great, but it will be better than doing nothing and the users won't have to fear losing lots of the important data with the true good values.

There were problems with Sway2 though. First and foremost, the model did very poorly in cases with high data and low clusterability. While it was able to at least shave off the really bad examples, it wasn't nearly as much as the original Sway model. In terms of the good groups found, Sway2 was not as consistent as the original sway. In some cases it appeared to do better but not all cases. Thus in terms of consistency and overall evaluation reduction, we have concluded that the original Sway model (Sway1) is better than our adjusted Sway2 model. Results to backup this claim can be found throughout the result section.

Even though the Sway2 model was not better than the original Sway model, it was able to show once again the benefits of clustering on the fishing problem as seen with some of the case studies. Not only has this paper shown the benefits of clustering and it's affect on the evaluations needed to understand and extract information from large data sets, but it has also provided a new customizable model to improve upon that is built upon less custom code than the original Sway model. Hopefully the integration of a popular library and its tools will allow for an easier learning curve when getting into this

research area and will allow others to begin implementing their own ideas faster.

II. Related Work

Deep Learning: Deep learning has been popular in software engineering for some time. With respect to our model, similar techniques have done feature extraction. Wang et al. [12] used deep learning to extract the semantic features from programs abstract syntax trees and did better on precision, recall, and F1 against other strategies by 14.7%, 11.5%, and 14.2%. Sadly, as pointed out by Fu et al. [7] in their case study on deep learning techniques, the authors failed to include the time it took to train this model. However, since the underlying deep learning algorithm would still need to utilize extensive computational power, it is safe to assume that the training time would be quite large as per other methods. Thus, regular users who do not have the computational power of large corporations need a smaller scale, less intensive, and more intuitive solution.

Sway "Sway" which is short for "the sampling way" is a technique to reduce the number of evaluations needed to find the best values given a very large input space. The idea comes from a paper written by Chen et al. [6] in which the authors compared their model against 12 state-of-the-art optimization models. It was found that for the categories they were comparing on (Generational Distance, Generated Spread, Pareto Frontier Size, and Hyper-volume) sway did at worst better than 6 of the other models for the pareto frontier size and at most beat 11 out of the 12 in generated spread. Sway was also faster than all of these models as well by a significant amount. This research model is what we based our Sway1 model after. If our algorithm is better than this model, it would then be better than all of these models as well.

Semi-Supervised It is computationally and temporally expensive to evaluate every data point in a data set to gather the necessary information for a fully supervised model. On the other hand, an unsupervised model can only gather data about the inputs and cannot gather any data on the resulting outputs. Thus, semi-supervised models try to combine the objectives of both into one model [11]. By doing so, the model is able to use the information about similar inputs, to find groups of good outputs using only a small number of computed outputs. We used this idea as a pruning method at the end of our clustering instead of in the middle of clustering like Sway1. Instead of cutting the data set in half due to evaluations, we let the clusterer decide how to break up the big data and reduced the space to the best cluster based on the evaluations on all of the centroids.

Multiple Objective The world we live in is often not so black and white as we would like. There are multiple factors that go into decisions and all must be considered. Thus, the need for multiple objective algorithms to determine the best items of a data set was born. Now, there are many different algorithms that can be used including relaxed dominance, diversity, indicator, preference, and dimensionality reduction [10]. For our model, we opted to go for a simplistic version of Zitzler's indicator based algorithm to determine the best value given multiple objectives [13].

Explanation Algorithms Not only do the Deep Learning models which have been known to be "black boxes" have little transparency, it is also hard to determine this with most common models today. With the ability to use libraries such as sklearn [3], users don't always know what is going on under the hood and will not be able to easily determine what the model is extracting. Thus, there is a need for explainability of these models which has been dubbed "XAI" by current research. There are different kinds of explainability and these have been covered in surveys of XAI [5]. For the explainability of our model we opted to use a random forest model to train on our data and then extracted the feature information that effected the decisions from that tree.

III. Methods

A. Algorithms

Zitlers Predicate: As mentioned above, within the Sway1 model we have implemented a simpler version of Zitzler's IBEA known as the continuous domination predicate [13]. Our algorithm compares the forward and backward jumps and the jump that wins is the one that loses the least. The forward jump is calculated by $s_1 = \sum_i e^{(max||min)*(row1_i - row2_i)/numGoals}$ and the backward jump is calculated by $s_2 = \sum_i e^{(max||min)*(row2_i - row1_i)/numGoals}$. Then we compare if s1 / length of the ys is less than s2 / length of ys. If it is, then the loss is less for row1 than it is for row2 so row1 is better in this case. Instead of implementing two seperate evaluation methods for Sway1 and Sway2, we opted to keep this for both so they are equivalent in their evaluation process.

Clustering Algorithm - BisectingKMeans: To simulate as closely as possible to the Sway1 using current model techniques, we opted to use the 'BisectingKMeans' cluster of sklearn [1]. Our implementation of Sway1 operates by clustering the data into two halves then recursively clustering on the best half determined by the Zitzler's predicate described above. For Sway2, we tried to cluster into groups first and then evaluate the centroids of those clusters to find the best out of all of the created centroids. The cluster connected to the best centroid is the one that we would keep. We stuck to using the default hyperparameters of the model to simulate as much as possible the "Off the Shelf" version of the model. This does not mean we did not make any modifications. We added data processing and pruning functionality to these models which allowed for the comparison of Sway1 and Sway2. We also changed the number of clusters allowed when doing the budget test, as the number of evaluations was directly tied to the number of clusters in the model.

Explanation Algorithm - RandomForests: Recreating a similar explanation algorithm was tricky. The custom explanation algorithm for Sway1 was able to capture within rules the groups of values that worked the best for describing what was captured in the clustering. There were no algorithms within the sklearn library that we found that were similar to our custom algorithm so we opted to go for a simpler feature explanation. We used a RandomForest classifier to classify the best against the rest of the data [2]. The random forest models work by having a multitude of descision trees that focus on specific features and only those features, then casts their vote to determine the prediction. RandomForest models within sklearn have a useful attribute which allowed us to extract the features that were important to it. We used the internal feature extraction functionality of this model to get percentages of feature importance. Once we had this, we matched the percentages to their respected features and ranked them based on their importance. If a feature was important throughout the 20 runs, it was saved based on its place and then presented in the top ten form. This allowed us to still extract the features that were most important in the classification and also allowed for us to use the classifier to gather best and rest data. We opted to use the defaults here as well since their accuracy was quite high as can be seen by the effect size tests below.

B. Data

There are 11 data sets we have chosen to look at for comparison of these tools. The data sets themselves included information on what to maximize and what to minimize denoted by "+" and "-" respectively at the end of the column names. Some of the data also included rankings, but these were ignored as it would skew the data

to those rankings rather than re-ranking which is what we wanted for comparison. The first two data sets, 'auto93.csv' and 'auto2.csv' are data on different cars. The algorithms are looking to find higher mpg, acceleration and lower weight to make a decision on which car is the best out of all of those cars. 'china.csv', 'coc1000.csv', and 'coc10000.csv' are all data on software projects. The algorithms are looking to reduce the effort, experience needed, risk, and program complexity. 'nasa93dem.csv' is similar to the software project data sets, but this warrants the algorithms to look for less time to completion and minimal defects. 'healthCloseIsses12mths0011-easy.csv' and 'healthCloseIsses12mths0001-hard.csv' include data on predicting the software project issues closure time based on random forest optimiztion values. The model is looking to reduce the error while boosting the accuracy and prediction values of the predictions of random forests. 'pom.csv' is data on agile project management. The algorithms are looking to find the least costly and idle teams that produce the most work. 'SSM.csv' and 'SSN.csv' are both computer physics data and are the biggest data sets of our study. Algorithms clustering for SSM are looking for smaller iterations to run as well as smaller times to get the desired solution. Algorihms clustering for SSN are looking for solutions that reduce the energy used as well as the peak signal to noise ratio.

C. Performance measures

We measured performance based on how close all of our algorithms were to the top items of the data set after evaluating all items in the data set against each other based on the defined objectives. The evaluations done on all of the algorithms was the discussed Zitzler's predicate. To gather the data points for comparison, we first ran both sway models and their respective explanation functions and got a group of the top values of the data set according to the clustering at the time of running. Then, we took the median of those results out of those guesses and re-ran the test 19 more times for a total of 20. Then we took the average of those values and used those to compare against the best and total average values in the data set.

D. Summarization Methods

We also performed a non-parametric effect size test to compare the differences of the results. The test we chose to use was the CliffsDelta test. We determined that to be considered truly significant, the result of this test needed to be above 0.4 as that is in-between the medium value of .474 and insignificant of below 0.147. The data doesn't need to be wholly different, but we also didn't want to make it indifferent either as that would give us no insight.

On top of the effect test, we calculated the different taxes that each of the models accrues. There are two major taxes that we covered which are the sampling and explanation tax. The sample tax is the information lost during the clustering. For example, if the best data item in the whole set was in a bad cluster and was removed, the loss of that data value is the tax. The second tax, explanation tax, is the information lost trying to extract the feature information from the data set and use that information to try and remake the best sets found by the clusterer. Since these explanations are simpler than the clustering itself, some of the values that were in the original good set are now lost and not found with the explainer, which are the explanation tax.

We chose to calculate these taxes by finding a lump sum value for all of the data sets to get a global picture of the two models for comparison. These results are calculated by the averaged sums across all goal data extracted from all data sets normalized.

TABLE I
AUTO2 AVERAGED RESULTS

Budget	PSNR-	Energy-	Number
10	**46.08**	**1185.35**	**45038**
25	**45.21**	**682.22**	**13**
50	**41.27**	**578.46**	**6**
100	**28.75**	**428.09**	**2**
200	**26.06**	**101.49**	**1**
500	**27.04**	**160.98**	**39**

This table holds the results for the budget test run on Sway2.

IV. RESULTS

Budget Size Test As one of the key contributions that Sway1 made when it was released was its minimal evaluations, we constrained Sway2 on a harsher metric of a constant of 8 evaluations. This proved to hinder our results for some of the bigger, less easily clustered data sets. Thus, we found it appropriate to test out how good the model could be if we relaxed our constraints. We tested the relaxed constraints on the worst data set for this model, 'SSN.csv'. What we found with the relaxed constraints was incredible. As seen in I, the results significantly improved even with just 25 evaluations. Now, 25 evaluations is still 2.5 times the number of needed evaluations that sway needed, but is still small enough to be considered better that most optimizers given the size of the data set. One interesting thing to note about the data though, is that at the point of 500 evaluations, the model actually started to get worse. Not only were the goals values getting further away, but the number of "goods" grabbed from the data set also started to go up. This led us to believe that there is a limit to how many evaluations you can give to the model before it starts to get worse, a potential avenue of research for the future.

Method Comparison When comparing our model sway2 against the original sway model, we ran each method 20 different times for each of the 11 data sets. Since the returned results of both models is a group of good values, we decided that to get the most accurate representation of that group would be to take the median value. This is because the groups that were returned by the models were clustered together due to their similarities so there should not be much variance in the results across the best list. Then we took the average of those 20 values, since we would have gotten 1 median for each of the 20 runs, and used that average value to compare the models against each other.

To compare the values given by these results, we used two metrics. First we compared the composition of the results themselves to determine if they were statistically different using the non-parametric effect size test known as CliffsDelta. Results for this are in Table II. The first part of the table displaying the results for 'auto2.csv' show the expected values for all of these tests and the real result for that dataset. As it can be seen, the data sets should be different from the average of all of the data, different from each other (since we want to find the better one), and more than likely different than the best since it is hard to get the best due to the different taxes. Below that first group are the interesting results. In some data sets, the explanation algorithms were different than the cluster models which leads to the conclusion that these models are harder to cluster and thus harder to extract the exact features that set them apart. Another interesting find was that sometimes sway1 was the same as the average of all of the data as seen in coc1000. This can be taken as a bad thing, but I like to think that it means that it was able to make a smaller scale version of the data while keeping the average results the same.

The second metric used for comparison was the amount of tax lost during the models clustering and explaining. In table III it shows the values for each of the different taxes and models. As we found,

TABLE II
NON-PARAMETRIC EFFECT TEST RESULTS

auto2.csv	CityMPG+	Highway MPG+	Weight-	Class-	
All vs All	=	=	=	=	
All vs Sway1	!=	!=	!=	!=	
All vs Sway2	!=	!=	!=	!=	
Sway1 vs Sway1	!=	!=	!=	!=	
Sway1 vs Xpln1	=	=	=	=	
Sway2 vs Xpln2	=	=	=	=	
Sway1 vs Top	!=	!=	!=	!=	
Sway2 vs Top	!=	!=	!=	!=	
auto93.csv	**Lbs-**	**Acc+**	**Mpg+**		
Sway1 vs Xpln1	!=	!=	!=		
Sway1 vs Sway2	!=	!=	=		
coc1000	**LOC+**	**AEXP-**	**PLEX-**	**RISK-**	**E-**
All vs Sway1	=	=	=	=	=
Sway2 vs Xpln2	!=	=	!=	!=	!=
healthCloseIsses12mths0011-easy.csv	**MRE-**	**ACC+**	**PRED40+**		
Sway1 vs Top	=	=	!=		
Sway2 vs Top	=	=	!=		

This table holds the results for the chosen effect test run on all data sets and contains specific different values

TABLE III
AVERAGE NORMALIZED TAX RESULTS

	Avg(Norm) Sample Tax	Avg(Norm) Explanation Tax
Sway/Xpln1	**-0.26**	**-0.10**
Sway/Xpln2	**-0.31**	**0**

This table holds the results for the average of the sum of the normalized difference between all of the min max goals for each data set

Sway1 had a better sampling tax with a value of -0.26 vs -0.31, but had a worse explanation tax of -0.1 vs 0. It was interesting to see that Xpln2 was able to fully capture the items in the clustered data set. This comes at a cost though as the features extracted from thses data sets were wildly different and Xpln2 was less informative than Xpln1 even though it had a worse tax.

Tables **??** all holds the different averaged results for all of the data sets for both Sway2 and Sway1.

February Study When creating a model like we have, the idea is that in the future you can use the data it produces to help you in some way. However, if you have no way of quickly getting that information once again at a later date, this makes the model a lot more difficult to use efficiently. Thus, we have conducted a February study to determine if we can use the information given as a result to speed up the process of finding useful information on the second run. For our process, this is in fact really easy due to how similar Xpln2 is to results of Sway2. This allows us to get all of the data first gathered by the clusterer again without having to re-evaluate anything taking our original evaluations to zero the following run. IV shows the results of the rerun named ReXpln2.

TABLE IV
FEBRUARY STUDY RESULTS

	Lbs-	Acc+	Mpg+	N	Evals
Sway2	**1965**	**17**	**30**	**60**	**8**
Xpln2	**1965**	**17**	**30**	**60**	**NA**
ReXpln2	**1965**	**17**	**30**	**60**	**0**

This table holds the results for the February test run on a 'auto93.csv'

TABLE V
ABLATION STUDY RESULTS

Removed Feature	Xpln2 Chosen Features
Clndrs	**Volume, origin, Model**
Volume	**origin, Clnders, Model**
Model	**Volume, origin, Clnders**
origin	**Volume, Clndrs, Model**

Included within are the results from the ablation study done on the 'auto93.csv' dataset. The removed feature column shows the feature removed from the file and the results of the Xpln2 trying to pull feature explanations out of it.

TABLE VI
HPO STUDY RESULTS

	Par1+	Par2-	Value	Time (S)
Grid Search	**78**	**18**	**60**	**101**
Sway2	**81**	**18**	**63**	**12**

This table holds the results for the HPO test run on a simple optimization problem

Ablation Study The extra features that are not goals of the optimization are important for the clustering process. We created Xpln2 for the purpose of being able to extract important features from the data that are factors of the clustering. So, to test and see if the extracted features we found really mattered, we ran an ablation study on 'auto93.csv' to see if removing these features impeded the accuracy of the model at all. The removals can be seen in V. As it turns out, for the most part these extra features do no matter a whole lot becasue as one is taken out, another jumps into place with no variance in the overall order. This could be specific to 'auto93.csv', but no futher testing was done on the other datasets.

HPO Study Since a lot of our data sets were dealing with pre-computed values and some even containing their own rankings, we decided to perform an HPO study on a simple optimization problem. We compared our results against a simplified GridSearch algorithm that searched the entire space for a hyper parameter optimization problem and found that not only did Sway2 beat the optimizer when finding a good average group, it did it ten times faster. This once again proves the valididty of the method of clustering before evaluating. Results are shown in the table VI

Research Question 1 When looking at the results for sway2, it becomes obvious that it was able to perform well for reducing the number of evaluations. It was able to reduce the evaluations of all of the test data sets that we looked into. The evaluation reduction ranged from at its best reducing the number of items to evaluate from 10000 down to 571 for 'healthCloseIsses12mths0011-easy.csv' in table XII to its worst reducing 53662 to 52981 for 'SSN.csv' table XVII. While some of the results were hindered by high sampling and explanation tax SAMPLING AND EXPLAIN TAX HERE. ACTUALLY JUST REMAKE THE BELOW

With reducing the number of items to evaluate, there comes taxes due to the lost information. This tax represents the loss of good values

TABLE VII
AUTO2 AVERAGED RESULTS

	CityMPG+	HighwayMPG+	Weight-	Class-	N	Evals
Data Set Average	**21**	**28**	**3040**	**17.7**	**93**	**NA**
Sway1	**29.16**	**34**	**2169.21**	**8.97**	**6**	**5**
Sway2	**30.8**	**36.65**	**2100**	**8.95**	**9**	**8**
Xpln1	**29.26**	**33.42**	**2323.16**	**9.84**	**18**	**NA**
Xpln2	**30.85**	**36.25**	**2106**	**8.86**	**9**	**NA**
Top Best	**33**	**41**	**2045**	**8.6**	**6**	**93**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'auto2.csv'. This is averaged over 20 runs

TABLE VIII
AUTO93 AVERAGED RESULTS

	Lbs-	Acc+	Mpg+	N	Evals
Data Set Average	**2800**	**15.5**	**20**	**398**	**NA**
Sway1	**2111.16**	**16.33**	**33.69**	**13**	**6**
Sway2	**1965**	**17**	**30**	**60**	**8**
Xpln1	**2190.32**	**16.07**	**30**	**81**	**NA**
Xpln2	**1965**	**17**	**30**	**60**	**NA**
Top Best	**1985**	**18.8**	**40**	**12**	**398**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'auto93.csv'. This is averaged over 20 runs

clustered into other groups that were not kept. The two tax results that we found, sample and explanation, were -0.31 and 0. The sampling tax was higher than we wanted and was more than likely due to the extra values in the good group. The explanation tax however was a very good value that exceeded expectations. Since the explanation tax was zero, there was no loss of information on average when trying to explain the data being produced. This worked really well for the February study in table IV where re-running the test to get the same cluster can be done with just the explanation model and requires not re-evaluations.

Research Question 2 A problem that has plagued many computer science researchers in the past is the problem of reproducibility [9]. It is hard to read a scientific paper and glean all of the details of a complex algorithm and or model and reproduce that model from scratch. In some cases there are software artifacts that you can download and try to run, however this does not always mean that they will work. Dependencies used could be no longer available, versions of the language could no longer be supported or distributed, as well as the operating system these tools were first run on to name a few problems. Thus it has been a task for researchers who wish to build off of the ideas of other projects to re-implement these ideas with existing tools.

This in essence was the point for Sway2. If the same results could be collected without as much manual work on the user adding complexity, then Sway2 could serve as a replacement for the current Sway1. However, after conducting the experiments and gathering the results the answer to this question is no. There were some cases where Sway2 did really well and outperformed Sway1. However, as the data sets got larger and more complicated, the clustering lost a lot of its effectiveness. Specifically for the 'SSM.csv' data set. It can be seen in the XVII results that this clustering method didn't reduce the evaluations needed much as well as it almost did worse than the average of the whole data set. The other results followed similar paths with only **??** being the only table with better results with little additional evaluations. With what we have stated in mind, we have reached a final conclusion that Sway2 is not better than Sway1 and that it needs further work and tweaking to make it so.

TABLE IX
COC1000 AVERAGED RESULTS

	LOC+	AEXP-	PLEX-	RISK-	Effort-	N	Evals
Data Set Average	**1078**	**3**	**3**	**5**	**19198**	**1000**	**NA**
Sway1	**1034.5**	**2.94**	**2.81**	**3.25**	**14783.38**	**19**	**7**
Sway2	**1575**	**3**	**2**	**8**	**67244.5**	**75**	**8**
Xpln1	**1084.56**	**2.94**	**3**	**4.56**	**21334.88**	**242**	**NA**
Xpln2	**1598.55**	**3**	**2.35**	**7.3**	**67617**	**66**	**NA**
Top Best	**1543**	**2.0**	**1.0**	**2.0**	**31323**	**19**	**1000**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'coc1000.csv'. This is averaged over 20 runs

TABLE X
HEALTHCLOSEISSES12MTHS0001-HARD AVERAGED RESULTS

	MRE-	ACC+	PRED40+	N	Evals
Data Set Average	**75.41**	**7.19**	**25**	**10000**	**NA**
Sway1	**73.7**	**7.57**	**25**	**79**	**8**
Sway2	**75.23**	**7.11**	**25**	**1704**	**8**
Xpln1	**73.77**	**7.56**	**25**	**2932**	**NA**
Xpln2	**75.31**	**7.44**	**25**	**1704**	**NA**
Top Best	**64.91**	**11.4**	**25**	**79**	**10000**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'healthCloseIsses12mths0001-hard.csv'. This is averaged over 20 runs

TABLE XI
COC10000 AVERAGED RESULTS

	Loc+	Risk-	Effort-	N	Evals
Data Set Average	**1030**	**5**	**18405**	**10000**	**NA**
Sway1	**1015**	**4.28**	**16777.72**	**79**	**8**
Sway2	**1440.25**	**6.95**	**47230.75**	**1148**	**8**
Xpln1	**1022.83**	**5.39**	**19355.5**	**2293.17**	**NA**
Xpln2	**1438.5**	**6.9**	**47301.15**	**1147**	**NA**
Top Best	**1959**	**0**	**17752**	**79**	**10000**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'coc10000.csv'. This is averaged over 20 runs

V. DISCUSSION

Threats to Validity Sway2 is far from perfect. We were unable to completely match the exact same structure as in the original Sway1 model. We opted for the closest similar model available at the time of writing for comparison, BisectingKMeans. While we originally wished to leave the model mainly untouched, we had to make adjustments to accurately extract the needed information for comparison. We believe that these adjustments were justified as without them no conclusions could be drawn from this study. The adjustments made for the different case studies were also technical

TABLE XII
HEALTHCLOSEISSES12MTHS0011-EASY AVERAGED RESULTS

	MRE-	ACC+	PRED40+	N	Evals
Data Set Average	**119.33**	**-12.2**	**0**	**10000**	**NA**
Sway1	**10.05**	**-0.05**	**75**	**79**	**8**
Sway2	**0**	**0**	**83.33**	**571**	**8**
Xpln1	**0**	**0**	**83.33**	**3302**	**NA**
Xpln2	**0**	**0**	**83.33**	**571**	**NA**
Top Best	**64.91**	**11.4**	**25**	**79**	**10000**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'healthCloseIsses12mths0011-easy.csv'. This is averaged over 20 runs

TABLE XIII
CHINA AVERAGED RESULTS

	N_effort-	N	Evals
Data Set Average	**2098**	**499**	**NA**
Sway1	**907**	**16**	**6**
Sway2	**758.1**	**244**	**8**
Xpln1	**1634.95**	**130**	**NA**
Xpln2	**758.1**	**243**	**NA**
Top Best	**145**	**16**	**499**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'china.csv'. This is averaged over 20 runs

TABLE XIV
NASA93DEM AVERAGED RESULTS

	Kloc+	Effort-	Defect-	Months-	N	Evals
Data Set Average	**47.5**	**252**	**2007**	**21.4**	**93**	**NA**
Sway1	**16.82**	**88.89**	**609.61**	**14.09**	**6**	**5**
Sway2	**12.61**	**61.2**	**542.7**	**13.90**	**39**	**8**
Xpln1	**18.57**	**106**	**750.22**	**14.72**	**18**	**NA**
Xpln2	**12.675**	**60.7**	**542.35**	**13.895**	**39**	**NA**
Top Best	**3.5**	**10.8**	**109**	**7.8**	**6**	**93**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'nasa93dem.csv'. This is averaged over 20 runs

TABLE XV
POM AVERAGED RESULTS

	Cost-	Completion+	Idle-	N	Evals
Data Set Average	**331.89**	**0.9**	**0.24**	**10000**	**NA**
Sway1	**283.01**	**0.91**	**0.23**	**78**	**8**
Sway2	**137.952**	**0.89**	**0.24**	**1562**	**8**
Xpln1	**330.92**	**0.9**	**0.23**	**7332**	**NA**
Xpln2	**137.86**	**0.89**	**0.24**	**1562**	**NA**
Top Best	**138.75**	**1**	**0**	**79**	**10000**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'pom.csv'. This is averaged over 20 runs

TABLE XVI
SSM AVERAGED RESULTS

	NUMBER-ITERATIONS-	TIMETO-SOLUTION-	N	Evals
Data Set Average	**7**	**?**	**239360**	**NA**
Sway1	**4.8**	**?**	**468**	**10**
Sway2	**6.1**	**126.47**	**214235**	**8**
Xpln1	**5.65**	**?**	**13258**	**NA**
Xpln2	**6.1**	**126.47**	**214237**	**NA**
Top Best	**3**	**?**	**468**	**239360**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'SSM.csv'. This is averaged over 20 runs

TABLE XVII
SSN AVERAGED RESULTS

	PSNR-	Energy-	N	Evals
Data Set Average	**45.96**	**1215.58**	**53662**	**NA**
Sway1	**43.76**	**1185.25**	**210**	**9**
Sway2	**45.64**	**1271**	**52981**	**8**
Xpln1	**45.24**	**1220.85**	**6086**	**NA**
Xpln2	**45.85**	**1269.5**	**52981**	**NA**
Top Best	**26.7**	**465.94**	**209**	**53662**

This table holds the results for the average of the medians of the extracted best list seperated from the rest for 'SSN.csv'. This is averaged over 20 runs

improvements to improve the drawing of results and did not change the clustering method or the explanation method.

Insights The biggest underlying insight that we as authors wish to have taken from this paper is the spark of the idea of difference between custom and commercial models. There are huge trade offs in terms of complexity and understanding that we did not have the means or skills to accurately portray. While we were able to work closely with the creators of the original sway paper to produce our model of Sway1, this would have been quite difficult for someone to achieve without such guidance. While this difficulty exists, our research goes to show that there are ways to recreate these models, even crudely, that allow for better results than doing nothing at all. It speaks to the power of the idea of using clustering to optimize the evaluation space. In other research areas, using current commercial models might not be as applicable as they could be fine tuned too much to perform specific tasks. However, it can be seen that for this particular problem, there wasn't as big of a gap in what is available now to the custom Sway1 model as there was at least a starting point for programmers and researcher to jump off from. An interesting question can be asked from this insight which is that since Sway2 worked well enough to reduce the space as we have seen, why have people been so slow to adopt this methodology? Is it because it hasn't become widespread enough? Time will tell.

Future Work Since the results of the explanation algorithms are so different, a future area of improvement would be to look deeper into which specific features are important to create a baseline, to then improve the explanation based on that baseline. The ablation study came close to this, but it only gives us a general idea of what features are truly important, without future data to back up those claims. Then it can be determined better which algorithm pulls the correct features. On the flip side, one of these algorithms could have been evaluating and extracting the features incorrectly and with more time and data this could be proven one way or the other. For now, it is left as a comparison with an open ended true correct evaluation.

Another area of interest to look into would be to come up with a systematic way to determine the optimal evaluation budget to produce

the best results. As seen in the budget test, increasing the available budget up to 25 had completely different results in comparison to the static 8 evaluations as is default. However, there was a point in which the larger number of evaluations actually hindered the results and made them worse. This optimization also seems somewhat like a fishing problem so applying the Sway methodology here could be something to look into as well.

VI. Conclusions

Using Semi-Supervised Multi-Objective Clustering and Explanation algorithms improves the efficiency of optimization tasks significantly. The idea of using clustering as an optimization tool holds true even for models with default hyper parameters built upon "Off the Shelf" commercial models. While there is a significant difference in the power of a good custom tool, the defaults are still able to reduce the number of evaluations needed and have a list of better values than originally started with. With time, and the adoption of the methodology of Sway, more models will begin to be produced and may find their way into popular libraries such as sklearn.

While we tried to reproduce a similar model to Sway1, it proved to be the superior version at this time. Tweaking the budget constraints could help bridge the gap between these two tools, but at this time it is not ready to take the mantle of number one.

References

[1] Bisectingkmeans. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.BisectingKMeans.htmlsklearn.cluster.BisectingKMeans. Accessed: 04-20-23.

[2] Randomforests. https://scikit-learn.org/stable/modules/ensemble.htmlrandom-forests. Accessed: 04-20-23.

[3] Sklearn homepage. https://scikit-learn.org/stable/. Accessed: 04-20-23.

[4] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[5] A. Adadi and M. Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[6] J. Chen, V. Nair, R. Krishna, and T. Menzies. "sampling" as a baseline optimizer for search-based software engineering. *IEEE Transactions on Software Engineering*, 45(6):597–614, 2018.

[7] W. Fu and T. Menzies. Easy over hard: A case study on deep learning. In *Proceedings of the 2017 11th joint meeting on foundations of software engineering*, pages 49–60, 2017.

[8] G. S. Handelman, H. K. Kok, R. V. Chandra, A. H. Razavi, S. Huang, M. Brooks, M. J. Lee, and H. Asadi. Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods. *American Journal of Roentgenology*, 212(1):38–43, 2019.

[9] W. Raghupathi, V. Raghupathi, and J. Ren. Reproducibility in computing research: An empirical study. *IEEE Access*, 10:29207–29223, 2022.

[10] A. Ramirez, J. R. Romero, and S. Ventura. A survey of many-objective optimisation in search-based software engineering. *Journal of Systems and Software*, 149:382–395, 2019.

[11] J. E. Van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[12] S. Wang, T. Liu, and L. Tan. Automatically learning semantic features for defect prediction. In *Proceedings of the 38th International Conference on Software Engineering*, pages 297–308, 2016.

[13] E. Zitzler, S. Künzli, et al. Indicator-based selection in multiobjective search. In *PPSN*, volume 4, pages 832–842. Springer, 2004.