Evaluating Ensemble Methods for Algorithmic Stability Under Data Perturbations

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we assess methods that can be utilized to induce selection stability in classification problems, such that predictions remain consistent under perturbations to the data. Standard classification algorithms often exhibit sensitivity near decision boundaries, which can compound instability that already exists from overfitting to a subset of high-influence samples. The recently proposed inflated argmax and subbagging framework — which takes an ensemble of predictions from models trained on different subsamples of the data combined with a selection rule that allows for multi-class predictions in ambiguous cases — addresses this and can yield substantial improvements in stability, as we validate in the cause-of-death classification setting. However, this approach still requires careful consideration of the stability of the base algorithm. In our experiments, the subbagging ensemble technique is not universally beneficial, as subbagging an already stable classifier can be detrimental to stability. We further examine the extent to which the Rashomon set — the collection of distinct models that achieve near-optimal predictive performance — can serve as a substitute for subbagging by leveraging predictive multiplicity of the Rashomon effect rather than resampling. We find that it is possible for the Rashomon ensemble to stabilize predictions, though this effect is by no means guaranteed. Our results are consistent with an influence equalization interpretation of ensemble techniques, which helps provide a natural explanation for both when subbagging fails and why the Rashomon ensemble, despite its theoretical appeal, can struggle in practice without a significant multiplicity effect.

1 Introduction

Stability is the degree to which the outputs, or statistical conclusions more broadly, of an algorithm remain consistent under perturbations to the training data (Yu, 2013). It shapes the level of trust we can place in the algorithm. When predictions are highly sensitive to the minute details of which data happened to be collected, then conclusions drawn from them risk reflecting noise in the sample rather than a real pattern in the world, undermining reproducibility and the fairness of subsequent decisions. The notion of stability has been well explored in the literature and has been shown to have significant downstream benefits in statistical analysis. Shalev-Shwartz et al. (2010) pinpoint stability as a necessary and sufficient condition for learnability. Likewise, Bousquet & Elisseeff (2002) and Poggio et al. (2004) relate stability to generalization, using stability to derive generalization error bounds and identifying stability as a sufficient condition for the generalization of learning algorithms. Yu (2013) shows that stability is important for interpretable and reliable scientific models and plays a key role in sensitivity analysis and effect detection. Murdoch et al. (2019) go so far as to say that "one should not interpret parts of a model which are not stable to appropriate perturbations to the model and data."

This paper serves to evaluate and compare key methods that can help achieve algorithmic stability and to demonstrate why and under what conditions they succeed or fail. The mechanisms behind these methods can be hard to decipher and the existing literature is especially sparse when it comes to the shortcomings of these methods. We aim to outline the potential consequences of adopting each technique and provide a unified account that helps practitioners decide which approach is most appropriate for a given setting, since there is no method that dominates in every scenario.

There are many ways to measure the stability — or conversely the instability — of a learning algorithm (Bousquet & Elisseeff, 2002), though a straightforward one is the difference between the predictions of the model trained on the full dataset and those of models trained on leave-one-out subsamples, which we refer to interchangeably as jackknife perturbations. Bootstrap and other resampling approaches can also work, depending on the desired level of data perturbation. The ensemble approach of subbagging, combined with the inflated argmax selection rule, has been shown to significantly reduce instability (Soloff et al., 2024a). Subbagging provides a stabilization effect by equalizing the influence of training samples, according to the interpretation of Grandvalet (2004), and the inflated argmax smooths hard decision boundaries in the selection process, further stabilizing predictions but requiring multi-class predictions for ambiguous cases. We consider using the Rashomon ensemble as a novel alternative, which relies on the Rashomon set to build an ensemble. Despite relying on different mechanisms — resampling versus predictive multiplicity — both approaches achieve stability through the same underlying influence equalization effect. However, as we will see, these methods do not work universally, as stable and unstable base algorithms react differently to ensembling.

Our findings are demonstrated primarily in the context of using verbal autopsy data for cause-of-death assignment, a setting where prediction consistency is critical. Verbal autopsy is a method used to determine the cause of death through interviews with the next of kin or caregivers of the deceased, using a standardized questionnaire to gather details on symptoms, medical history, and the circumstances leading to the death. Throughout much of the world, full autopsies are rare, and so this methodology has become increasingly important for assigning causes to otherwise uncertified deaths. Mortality is a continuous process, and the prominent causes of death shift over time as diseases or other risk factors emerge, decline, or spread across populations. Furthermore, as our understanding of new and even existing diseases progresses, such as with COVID-19, we get a clearer picture of the true underlying causes of death for certain symptoms, further altering the training data. Moreover, the models are trained and deployed across populations from different regions around the globe, making generalization a key component.

2 Background

There is a distinction to be made between traditional output stability and selection stability. The former deals with the real-valued output of the model — i.e. predicted class probabilities — while the latter deals with the predicted label, often determined by some selection criteria. These are related but not the same, and output stability does not imply selection stability (Soloff et al., 2024a). Both are worth considering, but in classification settings we usually ultimately care about selection stability as the predicted classes are the end result used to make decisions. In the multi-class prediction setting, it is important to understand the different ways we can define selection stability and how that choice will affect model selection.

2.1 Definitions of Instability

We consider a training dataset $\mathcal{D} = \big((X_i, Y_i)\big)_{i\in[n]}$ with n observations of features $X \in \mathcal{X}$ and labels $Y \in [L] = \{1, \ldots, L\}$. Borrowing notation from Soloff et al. (2024a), we say that a classification algorithm $\mathcal{C}$ has selection stability δ at sample size n if, for all datasets $\mathcal{D} \in (\mathcal{X} \times [L])^n$ and all test features $x \in \mathcal{X}$,

$$\frac{1}{n}\sum_{i=1}^{n} \phi(\hat{S}, \hat{S}^{\setminus i}) \leq \delta,$$

where $\hat{S} = \mathcal{C}(\mathcal{D}, x)$ and $\hat{S}^{\setminus i} = \mathcal{C}(\mathcal{D}^{\setminus i}, x)$ are the prediction sets on the test feature vector $x \in \mathcal{X}$ for the classifier $\mathcal{C}$ trained on the full training dataset $\mathcal{D}$ and the leave-one-out dataset $\mathcal{D}^{\setminus i}$, respectively. The $\phi(\cdot) \in [0, 1]$ represents the instability metric as a function of the two prediction sets.

We focus on the mean and maximum instability across the samples in the test set, where the mean reflects typical instability and the maximum captures the worst-case behavior over the most ambiguous test points. A held-out test set is preferred, as we are primarily concerned with the stability of predictions on unseen data. Each test point will have a measure of instability calculated across the leave-one-out datasets. The instability for a single sample in the test set is the empirical mean of the instability score comparing the

prediction sets of each leave-one-out model to the prediction set of the full model. Then the final measures for a given model are the mean and maximum instability across the test set. In this paper, we will use the following three definitions of instability:

$$\delta_j^{\text{Disjoint}} := 1 - \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left\{\hat{S}_j \cap \hat{S}_j^{\setminus i} \neq \varnothing\right\} \tag{1}$$

$$\delta_j^{\text{Overlap}} := 1 - \frac{1}{n}\sum_{i=1}^{n} \frac{|\hat{S}_j \cap \hat{S}_j^{\setminus i}|}{\min(|\hat{S}_j|, |\hat{S}_j^{\setminus i}|)} \tag{2}$$

$$\delta_j^{\text{Jaccard}} := 1 - \frac{1}{n}\sum_{i=1}^{n} \frac{|\hat{S}_j \cap \hat{S}_j^{\setminus i}|}{|\hat{S}_j \cup \hat{S}_j^{\setminus i}|} \tag{3}$$

Here we denote δ_j as the instability for the j-th element of the test set. The term inside the summation is a stability score comparing two prediction sets, ranging from 0 (fully unstable) to 1 (fully stable). Each of these definitions is symmetric in $\hat{S}$ and $\hat{S}^{\setminus i}$ — swapping the two prediction sets leaves the instability value unchanged — which aids interpretation, since stability does not depend on an arbitrary choice of which set is treated as the reference. To say that a classification algorithm $\mathcal{C}$ has selection stability δ is to say that $\max_j(\delta_j) \leq \delta$.

The disjoint definition says that two prediction sets are stable if they share any common elements — i.e. the sets are not disjoint. This results in a simple proportion of perturbation prediction sets that are stable. Under this definition, when moving from single to multi-class prediction sets, the stability can only increase or stay the same. The overlap definition is a little stricter, where prediction sets have a stability rating equal to the size of the intersection of the sets divided by the size of the smaller set. Like the disjoint definition, if the sets are equivalent or disjoint, then the stability is equal to one or zero, respectively. However, if the prediction sets share some common elements, then the stability will fall in $(0, 1]$. The Jaccard definition is the strictest, under which two prediction sets have a stability rating equal to the size of the intersection of the sets divided by the size of the union of the sets. The stability will still equal one for equivalent sets and zero for disjoint sets, but will fall in $(0, 1)$ for sets that are neither equivalent nor disjoint. Whereas the overlap stability can still equal one when the two sets share common elements but are not equivalent, the Jaccard stability will be strictly less than one in these cases.

The reason for outlining these similar, yet distinct, definitions is that each may favor a different type of model. The disjoint definition always favors a model that makes multi-class predictions, as the inflated argmax does, because it treats overlapping sets (i.e. sets that are neither equivalent nor disjoint) as fully stable. Thus, it does not capture the full extent of the similarity, or dissimilarity, of the prediction sets. The overlap and Jaccard definitions alter this to say that an equivalent set is more stable than an overlapping set, which is more stable than a disjoint set. Unlike the Jaccard definition, the overlap definition gives more leeway to different-sized sets, since the larger set need only contain each element of the smaller set in order to be fully stable.

Ultimately, the best definition is the one that fits the goals of the user, with specific consideration given to how reasonable and useful multi-class prediction sets are. In most cases, we recommend the overlap definition because it addresses the issue with the disjoint definition while not punishing dissimilarity solely due to differing set sizes. The reasoning is that the ordering of the predicted classes may be exactly the same, except that one more class made it into the prediction set, in which case we lean toward deeming that as generally stable. The set size is a separate measure that we would ideally minimize, so it makes sense to handle stability and set size somewhat separately. In practice, we found that the disjoint and overlap measures of stability rarely produce different results, while the Jaccard measure is noticeably more conservative.

2.2 The Inflated Argmax

The obvious issue that arises specifically in classification problems is that decision boundaries are inherently unstable. When a model decides between multiple classes, it must implement a hard selection rule — say taking the highest probability class. However, samples near the decision boundary are highly susceptible to being pushed over said boundary into a different class under perturbation. Hence, Soloff et al. (2024a) introduce the inflated argmax approach to address these issues that arise in classification settings.

Intuitively, the inflated argmax works by softening the hard boundaries between classes. Instead of returning just the highest probability class as a singleton set, it returns the set of classes that could plausibly be the top class under some level of data perturbation. More specifically, it returns the set of all classes j such that the predicted probability vector is within Euclidean distance ε of some probability vector where class j wins by a clear margin. The full definition from Soloff et al. (2024a) is as follows:

Definition 1 (Inflated Argmax) *For any $w \in \mathbb{R}^L$, the inflated argmax is defined as*

$$\operatorname{argmax}^{\varepsilon}(w) := \left\{ j \in [L] : \operatorname{dist}(w, R_j^{\varepsilon}) < \varepsilon \right\}, \tag{4}$$

where $\operatorname{dist}(w, R_j^{\varepsilon}) = \inf_{v \in R_j^{\varepsilon}} \|w - v\|$*, and where*

$$R_j^{\varepsilon} = \left\{ w \in \mathbb{R}^L : w_j \geq \max_{\ell \neq j} w_\ell + \varepsilon/\sqrt{2} \right\}.$$

By hedging our bets, in a sense, on ambiguous samples, we increase stability by avoiding uncertain predictions. Whereas with standard argmax small perturbations can push ambiguous samples over hard decision boundaries, the inflated argmax makes it so that this no longer changes the prediction set because the classes on either side of the boundary are already included in the set, thus making the predictions stable. This is where the definition of instability becomes very important, since we are now allowing for multi-class predictions.

In their experiments, Soloff et al. (2024a) demonstrate that the inflated argmax approach can drastically increase stability. The caveat is that, in order to achieve these gains, it must be applied in concert with a subbagging ensemble procedure.

2.3 Resampling Ensembles

There are two primary resampling approaches that are similar in their effect but somewhat distinct in implementation: bagging and subbagging. Each approach resamples the data and refits the model on that data to generate predictions on a test set, then averages over those predictions.

With training data of sample size n:

- **Bagging (Bootstrapping):** Sample n indices uniformly with replacement B times to create the ensemble.
- **Subbagging:** Sample m indices, where $m < n$, uniformly without replacement B times to create the ensemble.

Breiman (1996) shows that bagging can be an effective way to achieve accuracy gains, claiming that the instability of the base predictor is the key that allows bagging to improve accuracy. Bühlmann & Yu (2002) describe how bagging smooths hard decision boundaries and thus decreases variance and mean squared error. They introduce subbagging as a computationally cheaper alternative to bagging that achieves essentially identical performance. Elisseeff et al. (2005) derive bounds on the stability of subbagged algorithms. More recently, Soloff et al. (2024b) have furthered this by providing additional guarantees on the stability of subbagged algorithms, though they remark that it does not necessarily imply that subbagging must improve upon the stability of the base algorithm. Buja & Stuetzle (2006) show that bagging often, but not always, decreases variance while always increasing bias, at least in the U-statistic setting.

Grandvalet (2004) explores the reasons why bagging works and why it occasionally fails. As an alternative to the threshold softening and variance reduction explanations, Grandvalet argues that bagging equalizes the influence of training samples. He refers to influence in the sense of the well-defined leverage in the simple regression setting — that is, the sensitivity of a fitted value to a perturbation of the true label — but also offers a generalized leverage statistic for non-linear models. According to this framework, it is the "goodness/badness" of highly influential samples that explains the effect of bagging on accuracy, be it an improvement or deterioration. In most situations, highly influential points are outliers to some degree and therefore bad leverage points, so the down-weighting of their influence by bagging typically reduces variance and improves performance as downstream effects. Alternatively, when influential points are good leverage points, this same mechanism can increase variance and reduce performance. Here, an unstable predictor is one that has many highly influential points. This is why perturbing the data, and thus sometimes removing those points, leads to changes. This confirmed the hypothesis of Breiman (1996) that bagging increases accuracy due to the instability of the base predictor, but clarified more thoroughly the mechanisms behind why this is the case. His results are supported by Soloff et al. (2024b), in that "the main stabilizing effect of bagging comes from the removal of high-leverage data points from a certain fraction of bags".

Grandvalet (2006) expands on this idea for classification settings, where traditional measures of leverage do not apply. Hence, in the general case, he instead measures influence of sample i as

$$\Delta(\hat{f}, \hat{f}^{\setminus i}) = \mathbb{P}_X \left(\hat{f}(X) \neq \hat{f}^{\setminus i}(X) \right) \tag{5}$$

where $\hat{f}(X)$ and $\hat{f}^{\setminus i}(X)$ are the full and jackknife predictions, respectively, on a test set X. However, when a classifier returns prediction probabilities, a better measure of influence would be

$$\Delta(\hat{p}, \hat{p}^{\setminus i}) = \mathbb{E}_X \left\| \hat{p}(X) - \hat{p}^{\setminus i}(X) \right\| \tag{6}$$

where $\hat{p}(X)$ and $\hat{p}^{\setminus i}(X)$ are the full and jackknife prediction probability vectors, respectively.

Due to the essential equivalence of bagging and subbagging (Buja & Stuetzle, 2006), our results will largely be applicable to each approach, but for simplicity we will refer primarily to subbagging from now on. In the inflated argmax framework, subbagging is necessary to reduce the raw output instability, while the inflated argmax itself deals with the selection instability that remains due to test samples that sit near decision boundaries (Soloff et al., 2024a).

3 The Rashomon Alternative

The Rashomon set offers an alternative to resampling schemes as a way to generate multiple models from a single dataset. While bagging and subbagging average over numerous models of reduced quality, due to the resampling, the Rashomon set contains only high-quality models that are supported by the entire training set. This may help us avoid the situation where the subbagging process degrades accuracy to the point where any gains in stability are rendered meaningless.

The concept of the Rashomon effect was introduced by Breiman (2001), who observed that many distinct models can achieve near-equivalent predictive performance. Biecek et al. (2024) illustrate this phenomenon with what they call a *Rashomon Quartet*, which is a set of four models of different classes with identical predictive performance that nonetheless tell entirely different stories about the underlying relationships in the data. The idea and theory behind the Rashomon effect are outlined by Parikh (2026), who provides the following formal definition of the Rashomon set:

Definition 2 (Rashomon Set) *Let $\mathcal{F}$ denote a hypothesis class of models, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ a dataset, and $L : \mathcal{F} \times \mathcal{D} \to \mathbb{R}^+$ a loss function. Let $f^* = \arg\min_{f \in \mathcal{F}} L(f, \mathcal{D})$ denote the empirical risk minimizer with optimal loss $L^* = L(f^*, \mathcal{D})$. The* ***Rashomon set*** *$\mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D})$ for a given tolerance $\epsilon > 0$ is defined as:*

$$\mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D}) = \{f \in \mathcal{F} : L(f, \mathcal{D}) \leq L^* + \epsilon\}, \tag{7}$$

where ϵ is the Rashomon bound.

In simpler terms, the Rashomon set is the collection of models that achieve nearly equivalent loss to that of the optimal model, with the loss for each model falling within ϵ of the optimal model. These models may be the optimal model under small perturbations, or they may never be the optimal model. The common trait that they all share is that their empirical risk differs negligibly from the optimal, for a small ϵ, which means that accepting the optimal model as the "truth" implies that this negligible difference is meaningful. This is a sentiment that is difficult to justify.

The Rashomon ensemble takes a consensus of the predictions from each model in the Rashomon set to generate a final prediction. The idea is to average over the predictions from the Rashomon set as opposed to subbagged models, relying on the multiplicity to generate diverse predictions while using only the high-quality models within some Rashomon bound. Nguyen et al. (2026) have shown that this approach can provide benefits for accuracy in active learning settings, so we extend this idea to stability.

If we are to leverage the Rashomon effect to create the ensemble, one potential issue is that there is no guarantee of diverse predictions from models within any given Rashomon set. Even if we have a Rashomon set of healthy size, it may be relatively homogeneous in nature and lack predictive multiplicity — defined by Marx et al. (2020) as "the ability of a prediction problem to admit competing models with conflicting predictions". We can think of a simple example where we have only two covariates with perfect collinearity, which would produce a Rashomon set that contains two models, one that relies on each variable. However, if we used these two models to make predictions, we would get the exact same result because both carry the same information. Hence, a Rashomon ensemble can ultimately be averaging over identical, or near-identical, prediction vectors, which clearly provides no benefit and just additional computational load. As Krogh & Vedelsby (1994) observe, combining the outputs of several predictors is only useful if they disagree on some inputs, and it is important for generalization that they disagree as much as possible. Melville & Mooney (2004) build on this insight in the active learning space, showing that ensembles explicitly constructed for diversity can outperform standard bagging and boosting committees. Therefore, diversity is a critical property of an effective ensemble and thus the most basic requirement is that the Rashomon set be diverse.

With resampling approaches like bagging and subbagging, the fact that we blind each model to some of the data, and thus degrade that iteration, is precisely what encourages diversity across the ensemble. While there is no guarantee of this within a Rashomon set, in many cases with a rich feature set, the Rashomon set will be relatively diverse in that it contains models that produce conflicting predictions. This diversity is what we want to leverage to garner more stability.

Naively ensembling the entire Rashomon set may result in the ensemble underutilizing the true diversity (Nguyen et al., 2026). The Rashomon set may contain a plethora of trees that, while technically different, will have equivalent prediction vectors on the test set, and thus functionally they do not contribute anything new to predictive multiplicity. Hence, we need to prune the Rashomon set to reduce it to only the unique prediction vectors so that the ensemble does not get washed out by the most common prediction vector. We do this by removing any models that have a duplicate prediction vector and then adding a clustering algorithm to further condense the prediction set to truly unique predictive models. This helps us avoid the issue of insufficient diversity in the Rashomon set, which we know will doom the ensemble approach.

The other issue with using the Rashomon set has been the lack of computational efficiency up until recently. The SORTeD method from Arslan et al. (2025) builds on previous decision tree Rashomon set approaches, like that of Xin et al. (2022), to the point that we can tractably run our experiments with full enumeration of exact Rashomon sets of decision trees. For the subbagging process, we need to generate enough subbagged models such that, for each perturbation, we have a sufficient number of subbagged models that do not include any left-out samples in the training data. For larger perturbations, this quickly becomes infeasible. The computational benefit of the Rashomon set ensemble is that we no longer need to do this. Rather, we must find the models in the Rashomon set for the full training dataset and for each perturbation resample. In this way, the Rashomon approach is actually simpler and potentially more efficient. However, this breaks down when we have many features, since the computational cost of the Rashomon set quickly increases with the number of features.

4 Experimental Design

The verbal autopsy data comes from the Population Health Metrics Research Consortium (PHMRC) and was collected in four countries: India, Mexico, Tanzania, and the Philippines (Broad Institute of MIT and Harvard et al., 2013). The data contains survey responses and keyword counts, although our analysis is limited to the survey content. We condensed the cause of death categories down to five: cancers, cardiovascular, external injuries, infectious diseases, and other. The last category is a catch-all, but mostly contains noncommunicable and chronic diseases.

Therefore, we have the number of classes $L = 5$ where $Y \in [L]$ represents the true labeled cause-of-death, determined by healthcare professionals. With Dar, Tanzania designated as the validation site, the training set has a size of $n = 6115$ and the test set contains $N = 1726$ samples.

We evaluate the performance of the following methods:

1. The argmax of the base learning algorithm $\mathcal{A}$.
2. The ε-inflated argmax of the base learning algorithm $\mathcal{A}$.
3. The argmax of the subbagged algorithm $\widetilde{\mathcal{A}}_m$, with $B = 500$ bags of size $m = n/2$.
4. The ε-inflated argmax of the subbagged algorithm $\widetilde{\mathcal{A}}_m$, with $B = 500$ bags of size $m = n/2$.
5. The argmax of the Rashomon ensemble algorithm $\widetilde{\mathcal{A}}_{\mathcal{R}}$, with a majority vote from the Rashomon set with 5,000 trees reduced to 50 distinct prediction vectors via clustering.

Using the overlap definition of instability, we calculate the instability for each test point $j = 1, \ldots, N$ across the leave-one-out perturbation models as

$$\delta_j := 1 - \frac{1}{K_{\text{loo}}} \sum_{k=1}^{K_{\text{loo}}} \frac{\left| s(\hat{p}(\tilde{X}_j)) \cap s(\hat{p}^{\setminus i_k}(\tilde{X}_j)) \right|}{\min\left(\left| s(\hat{p}(\tilde{X}_j)) \right|, \left| s(\hat{p}^{\setminus i_k}(\tilde{X}_j)) \right| \right)} , \tag{8}$$

where s is the selection rule applied to the prediction probabilities $\hat{p}$ for the j-th test point $\tilde{X}_j$ and we sample $K_{\text{loo}} = 500$ points $i_1, \ldots, i_{500}$ uniformly without replacement from $[n]$ to create each leave-one-out instance. Using this, we take the mean and maximum instability across the test set as

$$\beta_{\text{mean-instability}} := \frac{1}{N} \sum_{j=1}^{N} \delta_j , \quad \beta_{\text{max-instability}} := \max_{j \in [N]} \delta_j . \tag{9}$$

We also compute accuracy and set size measures to determine the full usefulness of each approach, which we define as

$$\beta_{\text{accuracy}} := \frac{1}{N} \sum_{j=1}^{N} \mathbf{1} \left\{ s(\hat{p}(\tilde{X}_j)) = \{Y_j\} \right\} , \quad \beta_{\text{coverage}} := \frac{1}{N} \sum_{j=1}^{N} \mathbf{1} \left\{ \{Y_j\} \subseteq s(\hat{p}(\tilde{X}_j)) \right\} , \tag{10}$$

$$\beta_{\text{size}} := \frac{1}{N} \sum_{j=1}^{N} \left| s(\hat{p}(\tilde{X}_j)) \right| . \tag{11}$$

The accuracy, or more precisely singleton accuracy, is the proportion of the validation set for which the given model returns the true single class — singleton set — and thus is always inaccurate for any multi-class predictions. Coverage, on the other hand, is the proportion of the validation set for which the given model returns a prediction set that contains the true class. As the inflated argmax ε parameter increases, we will trade off singleton accuracy for coverage while increasing the average size of the prediction sets.

In evaluating the Rashomon ensemble, we are much more severely restricted than with subbagging. Enumerating the Rashomon set for additional model classes is an ongoing research area, but for now a tree-based method is one of the few classes where exact enumeration is possible and the computation cost is reasonable. Subbagging, on the other hand, can be done with essentially any model class that is reasonably efficient.

With the SORTeD package (Arslan et al., 2025), one could use the Rashomon multiplier or the maximum number of trees to restrict the size of the Rashomon set. However, since larger perturbations can alter the Rashomon effect, setting a maximum number of trees is more stable. For example, a fixed Rashomon multiplier may return a Rashomon set with thousands of trees for the full training data but then only return hundreds, or even just tens, of trees under some significant perturbation.

In our setting, the verbal autopsy dataset contains far too many features to make for an efficient example. Thus, to evaluate and compare the Rashomon ensemble approach, we instead turn to other popular machine learning datasets and synthetic datasets, the latter of which we can manipulate to effectively induce a strong Rashomon effect and predictive multiplicity (see Appendix subsection A.3).

5 Results

5.1 Stable and Unstable Classifiers

In Figure 1, we can see the stark contrast in the effect of subbagging on stable and unstable classifiers. The stable classifiers have base mean instability near zero, and subbagging increases that slightly. The unstable classifiers start with relatively large mean instability, and subbagging provides a significant reduction, making them the perfect use case for subbagging. Illustrative examples are shown in the appendix in Table 1 and Table 2.

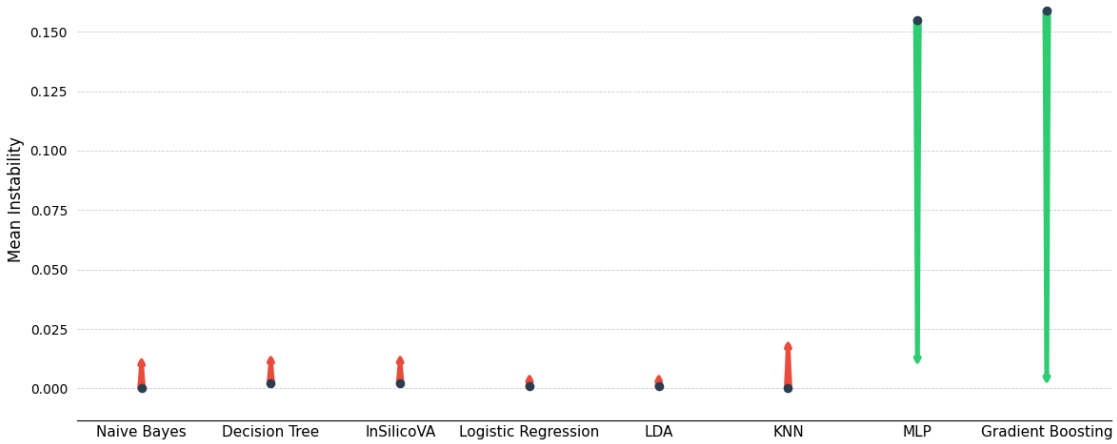


Figure 1: Comparing the effects of subbagging on mean instability ($\beta_{\text{mean-instability}}$). The dots show the mean instability of the base algorithm ($\mathcal{A}$) and arrows point to the mean instability of the subbagged algorithm ($\widetilde{\mathcal{A}}_m$).

Figure 2 shows the effect of subbagging and the inflated argmax on maximum instability. With only subbagging, there is a significant increase in the maximum instability for the stable classifiers, whereas the unstable classifiers see minor reductions. The key point is that combining subbagging with the inflated argmax *can* counteract the negative effects of subbagging a stable classifier and even bring the instability below that of the base algorithm. This will clearly not always be the case, since the inflated argmax may struggle with non-smooth classifiers and will not have much of an effect if the inflated argmax ε parameter is not large enough. Furthermore, if we consider the Jaccard definition of stability, as seen in Figure 6 of the appendix, then the effect of the inflated argmax is much less impressive. Table 13 of the appendix offers a more detailed illustrative example for how each definition changes the effect of the inflated argmax, where the Jaccard definition is significantly more conservative.

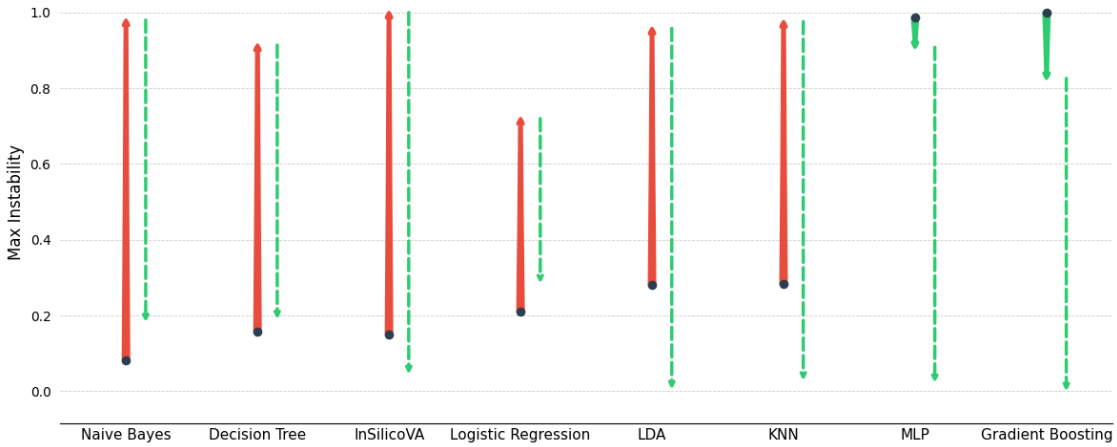


Figure 2: Comparing the effects of subbagging on maximum instability ($\beta_{\text{max-instability}}$). The dots show the maximum instability of the base algorithm ($\mathcal{A}$), solid arrows point to the maximum instability of the subbagged algorithm ($\widetilde{\mathcal{A}}_m$), and dashed arrows point to the maximum instability of the ε-inflated subbagged algorithm ($\widetilde{\mathcal{A}}_m^{\varepsilon}$).

We found, as expected, that the inflated argmax alone often struggles in addressing instability issues. Subbagging by itself can significantly decrease the mean instability, though the maximum instability often remains rather high. However, combining subbagging and the inflated argmax most consistently reduces the instability to near zero.

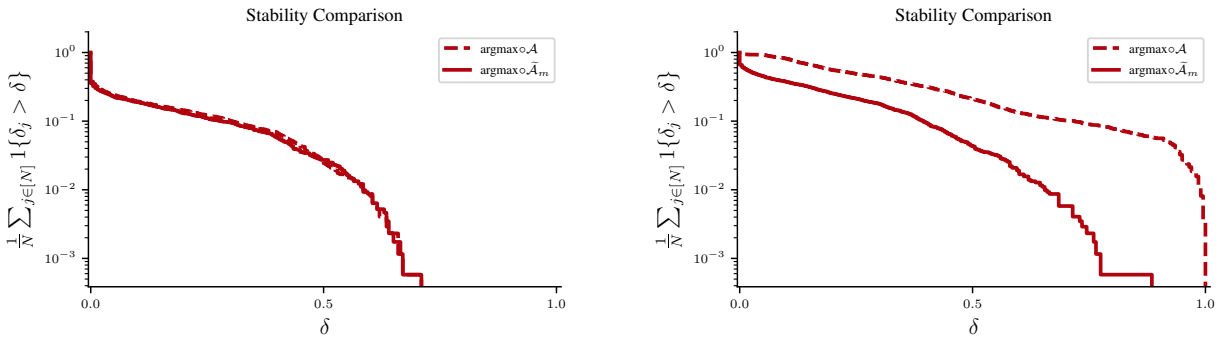
5.2 Increasing the Perturbation

The base stability is ultimately a relatively arbitrary measure, since leaving one sample out as a perturbation is an arbitrary choice. It could just as well be that we leave out more samples, thus causing larger perturbations. Then the instability of the base algorithm will inevitably be higher, yet nothing has fundamentally changed with our algorithm, so should the effect of subbagging be any different for the supposedly stable classifiers?

To investigate this question, we increase the perturbation level to see how that changes the effect of subbagging. Instead of a minor leave-one-out perturbation, we take a bootstrap resample as the perturbation. When using bootstrap perturbations, the base instability of each algorithm will inevitably be *nominally* higher since the perturbation level is higher. The interesting part is what happens with the effect of subbagging, as seen in Figure 3 below and Figure 8 in the appendix.

We use Linear Discriminant Analysis (LDA), naive Bayes, decision tree, and regularized decision tree classifiers, all of which are jackknife stable, trained on verbal autopsy data with Dar as the validation site. Each of these classifiers saw subbagging increase the instability when considering leave-one-out perturbations. For the non-tree models, when we increase the perturbation level, the instability of each approach converges, where the instability is essentially equivalent across each procedure. In these cases, our chosen perturbation level changed the *magnitude* of the effect of subbagging, but did not change the direction of that effect.

However, with these tree classifiers, this convergence is not observed, but rather the directional effect of subbagging reverses from being detrimental to becoming an improvement. It is unclear whether this effect is specific to trees. Regardless, we see that the chosen perturbation level does matter in determining the effect of subbagging. In some cases, larger perturbation levels render subbagging meaningless, while in others larger perturbation levels may allow subbagging to actually improve stability. For our decision tree classifier, subbagging increased the mean leave-one-out instability from 0.002 to 0.013 but decreased the mean bootstrap instability from 0.317 to 0.125. In scenarios like this, we may want to sacrifice a small increase in



(a) Linear Discriminant Analysis (LDA)

(b) Decision Tree

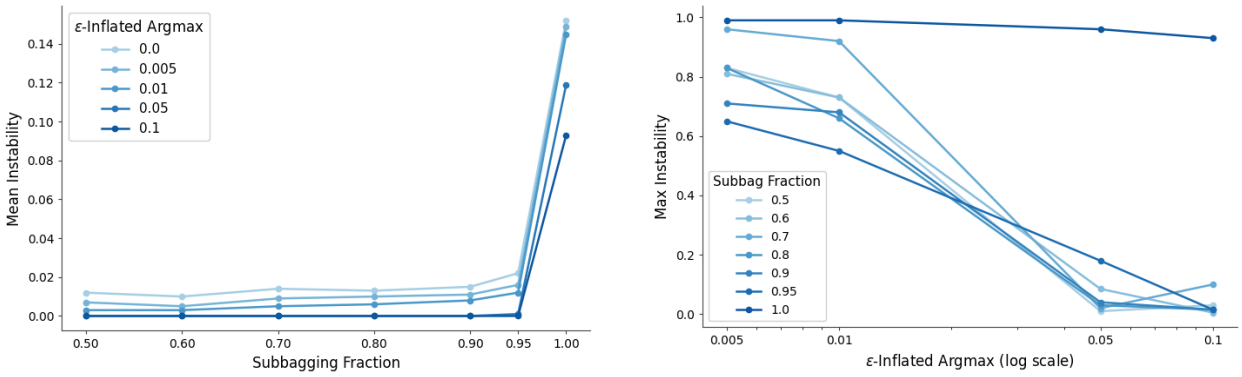
Figure 3: Instability plots for jackknife stable classifiers with bootstrap perturbations. The curves show the proportion of samples in the test set with instability greater than δ for unbagged (dashed) and subbagged (solid) algorithms.

jackknife instability if it comes with a significant enough reduction in bootstrap instability, especially if the inflated argmax can further reduce both types of instability to near zero.

5.3 Subbagging Fraction and Inflated Argmax Sensitivity

Choosing the correct subbagging fraction and ε-inflated argmax parameter is not a simple formula. In practice, the best approach is likely to be trial-and-error based. First, depending on the class of model and the dataset, computation can be quite expensive, especially as m increases. This cost makes a thorough exploration relatively infeasible. There is also no easy way to gauge the correct ε parameter without first testing values, since we cannot directly target a certain set size, for example. Finally, different scenarios are likely to present different goals. One setting may favor coverage, while another favors singleton accuracy. Ultimately, different practitioners may want to target different metrics.

In our sensitivity analysis, the first observation is that subbagging does not necessarily require significant subsampling to induce the stability benefits. Figure 4a shows that for the MLP classifier the instability is about as low with 95% of the data sampled for each bag as it is with 50% of the data. This gives significant leeway in choosing the subbagging fraction, which is especially useful in settings where the fraction has a noticeable effect on accuracy.



(a)

(b)

Figure 4: Subbagging fraction and ε-inflated argmax sensitivity analysis. The results illustrate the parameter ranges over which the subbagging fraction and the ε-inflated argmax most significantly reduce instability.

Furthermore, Figure 4b shows that while subbagging is often the significant driving force when it comes to *mean* instability, the ε parameter of the inflated argmax is more influential when it comes to *maximum* instability. The maximum instability deals with the worst-case, or most ambiguous, sample in the test set. These points are likely to be near the boundary between classes. Although subbagging alone cannot effectively handle that, the inflated argmax is built specifically to solve this issue by including all classes close enough to the boundary. We can also clearly see that the inflated argmax does not work well without any resampling.

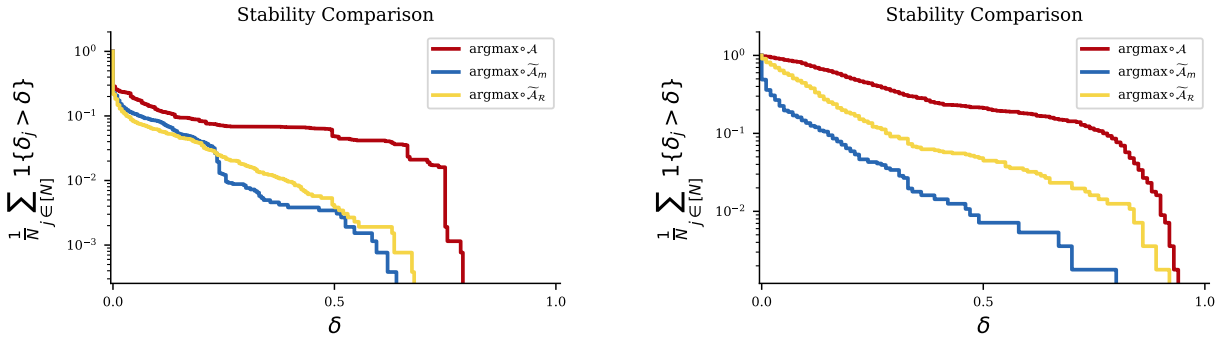
5.4 Stability of the Rashomon Ensemble

Even though we are restricted to a single model class for the Rashomon ensemble and thus cannot make comparisons across a broad range of stable and unstable classifiers, we still find promising, if not ultimately mixed, results.

In our initial experiments, leave-one-out perturbations were simply not significant enough to noticeably alter the model or the Rashomon set, given the cost complexity imposed to keep the Rashomon computation reasonable (see Figure 9 in the appendix). Given that subbagging is ineffective in this case, it is unreasonable to expect the Rashomon ensemble to perform any better. Yet, as we saw previously, specifically with tree-based methods, even if the classifier is jackknife stable, it can still be bootstrap unstable and the subbagging ensemble approach can help mitigate that. Hence, we evaluate the performance of the Rashomon ensemble method under bootstrap perturbations.

Figure 5 shows the results from the Home Equity Line of Credit (HELOC) dataset from FICO (left) and a synthetic dataset (right) (FICO, 2018). In the HELOC example, we see that the Rashomon ensemble can provide clear benefits in terms of stability. Both subbagging and the Rashomon ensemble achieve near identical reductions of instability, dropping mean instability from 0.058 to 0.021 and 0.019, respectively. On the full training set, the Rashomon set contained 5,497 trees (the maximum is not strict, likely accepting ties), of which 347 produced unique prediction vectors.

Although the Rashomon ensemble did very slightly edge out subbagging in the HELOC case, we could not find an example where it significantly outperformed subbagging. In our experiments with other datasets, it was more often the case that both approaches performed similarly while not providing much instability reduction, though this may be more of a reflection on the method underlying SORTeD.



(a) HELOC Dataset

(b) Synthetic Dataset

Figure 5: Instability plots from bootstrap perturbations with the Rashomon ensembles based on a maximum of 5,000 trees. The curves show the proportion of samples in the test set with instability greater than δ.

The synthetic dataset was designed to induce a strong Rashomon effect by pruning training points near the decision boundary — the samples most informative about boundary placement — leaving the boundary region sparse and allowing many distinct models to achieve near-identical empirical loss. If there existed different mechanisms driving instability, then this dataset should target a multiplicity mechanism, if it exists and is separate from the random sampling noise that subbagging addresses. On the full training set, the

Rashomon set contained 6,573 trees, of which 1,373 produced unique prediction vectors. As we see in Figure 5b, the Rashomon ensemble clearly reduces instability, from 0.298 mean instability to 0.126, but subbagging reduces instability further, down to 0.044. This provides no evidence of separate mechanisms, since even in the setting designed for the Rashomon ensemble, subbagging outperforms it.

When comparing fits on the same dataset with different SORTeD parameters, we found that increasing the predictive multiplicity, as measured by average pairwise disagreement, does not necessarily translate to a greater reduction in instability. We know that predictive multiplicity is necessary, to some degree, but there is no consistent and direct relationship between it and instability reduction.

These examples serve to show that it is *possible* for the Rashomon ensemble to be effective under favorable conditions, not that it is necessarily typical. The conclusion is that the Rashomon ensemble can reduce instability and even match the subbagging ensemble in performance. As additional Rashomon enumeration methods are developed, we believe that these results, specific to the class of sparse decision trees, will generalize to other model classes as well.

6 Discussion

6.1 When Ensembling Succeeds

The stability of a learning algorithm under leave-one-out perturbations is directly related to the influence of each of the left-out samples, under the jackknife definition of influence, since both measure the sensitivity of predictions to the leave-one-out perturbation. Thus, for any technique to improve stability, it must in aggregate reduce the influence of samples in the training set. A classifier that is jackknife unstable has many high-influence points, so, under the Grandvalet (2004) interpretation, the down-weighting of these high-influence points is the dominant effect and leads to a reduction in overall influence, which directly translates to jackknife stability. Subbagging excludes a certain proportion of data points from each bag, giving those points no influence. Thus, when these high-influence points are left out as a perturbation, the predictions from many of the bags do not change, thereby providing stability.

The Rashomon ensemble has no resampling scheme, but there is still a clear influence equalization effect, which we see in Figure 10b of the appendix. Referring to Definition 2, the hypothesis class of models $\mathcal{F}$ is fixed and the dataset $\mathcal{D}$ enters only through the loss function L to determine membership in the Rashomon set $\mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D})$. By defining $\mathcal{F}$ over the feature space $\mathcal{X}$, the hypothesis space from which ensemble membership can be chosen is invariant to data perturbations such as leave-one-out. The implication is that, unlike stochastic resampling schemes, data perturbations are not accompanied by refitting a model, but rather apply a new filter or aperture that shifts across a pre-existing hypothesis space. Any given model $f \in \mathcal{F}$ has no sensitivity to perturbations; it remains static.

Xin et al. (2022) show that the Rashomon set at ϵ'-tolerance after removing a group of instances is contained within the full-dataset Rashomon set at a larger ϵ-tolerance. Hence, for a data perturbation such as leave-one-out, denoted $\mathcal{D}^{\setminus i}$, we have $\mathcal{R}(\epsilon', \mathcal{F}, \mathcal{D}^{\setminus i}) \subseteq \mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D})$. Since Rashomon sets are nested and $\epsilon' \leq \epsilon$, we also have $\mathcal{R}(\epsilon', \mathcal{F}, \mathcal{D}^{\setminus i}) \subseteq \mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D}^{\setminus i})$. Together, these imply that $\mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D}) \cap \mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D}^{\setminus i})$ is non-empty. For all models f that are contained in the perturbed and unperturbed Rashomon sets at the ϵ-tolerance, such that $f \in \mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D}) \cap \mathcal{R}(\epsilon, \mathcal{F}, \mathcal{D}^{\setminus i})$, these committee members will contribute the same information regardless of the perturbation. Therefore, the jackknife influence of the removed point will be zero for these persistent committee members, which provides an effect akin to the exclusion of points that occurs in bagging.

In the Rashomon ensemble, for a fixed ϵ, a data perturbation shifts the membership boundary relative to the fixed elements of $\mathcal{F}$ by altering the loss $L(f, \mathcal{D})$ for every $f \in \mathcal{F}$. So, the influence of a removed point is only manifest in the fluid margin of the ensemble — the models that enter or exit the Rashomon set threshold due to the perturbation. A subset of the original Rashomon set will remain and thus the stable intersection dilutes the influence, mimicking the down-weighting effect seen in resampling schemes. Our use of bootstrap perturbations means that the theorem from Xin et al. (2022) does not apply directly, since it deals with fixed group removal, although we conjecture that it is still relevant, perhaps to a lesser degree, since the general idea is about the construction of the Rashomon set under data perturbations.

Unlike subbagging, this effect is far from guaranteed and requires not insignificant stable intersections. In our experiments, we found that a strong Rashomon effect, with sufficient predictive multiplicity, is necessary for a meaningful reduction in instability. A moderate Rashomon effect simply does not have the same power to down-weight high-influence points as subbagging, especially since predictive multiplicity of the Rashomon set alone is not enough, but rather models must also vary in which points drive their membership, so that a stable intersection can persist under perturbation. We found no evidence that the Rashomon ensemble can significantly outperform subbagging, but we are restricted to just a single tree-based class of models. In the future, a fairer comparison can be made once we are able to efficiently enumerate Rashomon sets across many different model types, which may reveal settings in which the Rashomon ensemble is far more effective than we have shown.

6.2 When Ensembling Fails

We argue, as the results of our experiments have shown, that the issue with both ensemble approaches is that they are detrimental when the base algorithm — without ensembling — is already stable. Subbagging increases bias and can sometimes increase variance (Buja & Stuetzle, 2006). The bias issue can clearly negatively affect accuracy, but does not explain the negative effects on stability since a biased subbagged model will induce similarly biased jackknife models, so the bias effect should largely cancel out in the instability comparison.

Increased variance could serve as an explanation, but the issue is that the effects of subbagging on accuracy and stability do not necessarily move in the same direction. Table 9 of the appendix shows that even when subbagging significantly increases accuracy, it can still negatively impact stability — with a minor effect on mean instability but a large effect on maximum instability. As Breiman (1996), Bühlmann & Yu (2002), and Grandvalet (2004) all agree, when subbagging aids predictive performance, it is associated with a reduction in variance. Hence, an increase in variance is also not sufficient to explain why the instability increased.

The explanation must hinge on what happens when we refit the model while leaving samples out, and likely has to do with the distribution of influence. Regardless of the state of the base algorithm, subbagging should theoretically always equalize influence, and if the influence is already relatively equal, then there should be no effect, as argued by Grandvalet (2004). However, even if they are often associated, *equalizing* influence does not necessarily equate to *reducing* influence. Figure 7 of the appendix shows that even for jackknife stable classifiers, there is still an influence equalization effect with subbagging, even though the instability increases and thus the overall influence increases as well.

A natural explanation is that subbagging simply has a higher instability floor, which is nonzero. If we take a stable classifier — where the jackknife instability is near zero — then each training point already has relatively low influence. Hence, a leave-one-out perturbation will not change the output because we have assumed the instability to be near zero. However, subbagging gives each point zero influence in the bags that exclude it but likely higher influence in the bags that include it, due to the smaller sample size. This is also the case with bagging, yet the higher influence comes from duplicate occurrences rather than a smaller sample size. Now, under a leave-one-out perturbation, the bags that did not contain the removed point do not change. However, the bags that contain this point are more likely to change because this point has higher influence in those bags, which may be just enough to change the predictions. Thus, the subbagged algorithm is more unstable because there are small changes in the predictions from some bags. In the case of an unstable base algorithm, this effect is negligible compared to the influence equalization and reduction. With little or no reduction needed, this destabilizing effect now dominates. Essentially, when the base model does not change under perturbations, the benefit of zeroing out the influence of each point in some bags is completely negated because those predictions already remain constant for the full model when each point is removed.

A similar destabilizing mechanism should be present with the Rashomon ensemble, as observed in Figure 9 of the appendix. When the algorithm for the optimal model is jackknife stable, the loss of the optimal model, L^*, will not change much under leave-one-out perturbations. However, knowing that the optimal loss L^* is relatively insensitive to perturbation does not guarantee the same invariance for $L(f, \mathcal{D})$ across all $f \in \mathcal{F}$. For a fixed $f \in \mathcal{F}$, removing point i shifts $L(f, \mathcal{D})$ by an amount governed by how f's own loss at

point i compares to f's average loss overall. So, for models $f \in \mathcal{F}$ that sit near the Rashomon membership boundary, those that fit point i exceptionally well or exceptionally poorly are those whose membership is most likely to flip. Even for a small change in the loss, if the model crosses the boundary, then the ensemble gains or loses an entire prediction vector. Because the baseline instability is already near zero in this case, even a single such entry or exit is no longer negligible. The churn that would be dwarfed by a high baseline for an unstable base algorithm instead dominates the ensemble average for a stable one, thus inducing the destabilizing effect. Ultimately, the destabilization of stable algorithms is likely a feature of any ensemble approach that equalizes influence. The same reason that they work for unstable algorithms makes them ill-suited for stable ones.

This explanation fits with the results of subsection 5.2, where we saw that, for jackknife stable classifiers, the stability of a subbagged algorithm can converge to that of the base algorithm as the perturbation level increases, if not reverse the directional effect entirely. This is consistent with the interpretation that when the perturbation level increases, the nominal instability of the base algorithm increases above zero, and thus the zeroing out of influence in some bags once again provides a benefit that counteracts the instability effect of higher influence in the other bags. However, since the base algorithm is still inherently jackknife stable, no single point has outsized influence, and thus the benefit may be limited. The higher instability is induced by a more significant perturbation, where high-influence clusters of points are removed. As Grandvalet (2004) points out, bagging can be less effective at down-weighting clusters of leverage points because the probability of excluding the entire cluster from a bag becomes exceedingly small with larger clusters. Hence, bagging and subbagging may struggle to even affect a classifier that is jackknife stable yet bootstrap unstable because of the nature of the resampling processes. This is likely why, in some cases, we see subbagging have no effect at these higher perturbation levels.

This is where the Rashomon ensemble could have an advantage because it has no such drawback and may be able to effectively down-weight clusters of leverage points by emphasizing different features. This warrants further investigation, but again we are currently restricted to tree-based methods, which unfortunately do not represent the setting where the effect of subbagging disappears at higher perturbation levels. As such, we require a new enumeration method that relies on a different model class.

In many scenarios, achieving jackknife stability is sufficient, but this is certainly not always the case, especially with larger sample sizes. In settings where we worry about what happens under more significant perturbations, the effect of subbagging on jackknife stable classifiers can either be completely negated or actually become beneficial. When subbagging is rendered ineffective at combating significant perturbations, it is not necessarily clear that the Rashomon ensemble would suffer the same effect. On the other hand, when subbagging is detrimental to jackknife stability but beneficial to bootstrap stability, a tradeoff must be made regardless of the ensemble method.

6.3 Recovering Stability with the Inflated Argmax

In our experiments, output stability mostly mirrored selection stability, though it is important to remember that this is not always the case. Subbagging has a direct effect on *output* stability and thus an indirect effect on selection stability. The inflated argmax rule deals specifically with selection stability, as it has no effect on the raw output.

The inflated argmax approach can actually remedy the negative effect that ensembling may have on stability. We saw that even when subbagging increased the selection instability, adding the inflated argmax on top of it could reduce that instability down to essentially zero, even sending the maximum instability below that of the original base algorithm in some cases. The added instability associated with subbagging is relatively minor and is only due to small changes in influence, meaning that the perturbations likely cannot generate large shifts, and thus most of the added instability should be from points that are still near decision boundaries. Hence, the inflated argmax can help, and thus allows us to utilize subbagging on any classifier, be it unstable or stable, without ever sacrificing much stability. Granted, this may not be true with the Jaccard definition of instability, which is where it falls to the user to determine the definition of instability that fits the goals of the experiment.

7 Conclusion

Instability is a direct product of the high influence of individual data points in the training set, such that small jackknife perturbations can lead to significant changes when these points are removed. Our results are consistent with ensemble approaches — from bagging and subbagging to the Rashomon ensemble — equalizing the influence of data points. The Rashomon ensemble does not have the same diversity guarantee as bagging and thus works only in settings with predictive multiplicity and diversity of influence. When the base algorithm is jackknife unstable, these ensemble techniques provide a stabilizing effect, yet they provide a destabilizing effect when the base algorithm is already stable, likely because the influence becomes more unequal for each individual model of the ensemble. This fits with the idea that the influence equalization effect leads to an overall reduction in influence when high-influence points exist, though not necessarily so otherwise. For selection stability specifically, the inflated argmax smooths hard decision boundaries and can even counteract the destabilizing forces associated with ensembling stable classifiers, potentially making ensembling applicable and beneficial in more settings.

References

Elif Arslan, Jacobus G. M. van der Linden, Serge Hoogendoorn, Marco Rinaldi, and Emir Demirović. Sorted rashomon sets of sparse decision trees: Anytime enumeration, 2025. arXiv:2511.03344.

Przemysław Biecek, Hubert Baniecki, Mateusz Krzyziński, and Dianne Cook. Performance is not enough: The story told by a rashomon quartet. *Journal of Computational and Graphical Statistics*, 33(3):1118–1121, 2024. ISSN 1537-2715.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–215, 2001. ISSN 08834237, 21688745.

Broad Institute of MIT and Harvard, Chhatrapati Shahuji Maharaj Medical University (India), George Institute for Global Health (India), Institute for Health Metrics and Evaluation (IHME), Johns Hopkins Bloomberg School of Public Health, Muhimbili University of Health and Allied Sciences (Tanzania), National Institute of Public Health (Mexico), Nizam's Institute of Medical Sciences (NIMS), Public Health Laboratory-Ivo de Carneri (Tanzania), Research Institute for Tropical Medicine (Philippines), and University of Queensland (Australia). Population health metrics research consortium gold standard verbal autopsy data 2005–2011, 2013.

Peter Bühlmann and Bin Yu. Analyzing bagging. *The Annals of Statistics*, 30(4):927–961, 2002.

Andreas Buja and Werner Stuetzle. Observations on bagging. *Statistica Sinica*, 16(2):323–351, 2006.

Andre Elisseeff, Theodoros Evgeniou, and Massimiliano Pontil. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(3):55–79, 2005.

FICO. Explainable machine learning challenge, 2018.

Yves Grandvalet. Bagging equalizes influence. *Machine Learning*, 55(3):251–270, 2004.

Yves Grandvalet. Stability of bagged decision trees. In *Scientific Meeting of the Italian Statistical Society (SIS)*, pp. 221–230, 2006.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In G. Tesauro, D. Touretzky, and T. Leen (eds.), *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994.

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the COMPAS recidivism algorithm. ProPublica, 2016.

Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6765–6774. PMLR, 13–18 Jul 2020.

Tyler H. McCormick, Zehang Richard Li, Clara Calvert, Amelia C. Crampin, Kathleen Kahn, and Samuel J. Clark. Probabilistic cause-of-death assignment using verbal autopsies. *Journal of the American Statistical Association*, 111(515):1036–1049, 2016.

Prem Melville and Raymond J. Mooney. Diverse ensembles for active learning. In *Proceedings of 21st International Conference on Machine Learning (ICML-2004)*, pp. 584–591, July 2004.

W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116 (44):22071–22080, October 2019. ISSN 1091-6490.

Simon D. Nguyen, Hayden McTavish, Kentaro Hoffman, Cynthia Rudin, and Tyler H. McCormick. Realitrees: Rashomon ensemble active learning for interpretable trees, 2026. arXiv:2603.22750.

Harsh Parikh. Why are there many equally good models? an anatomy of the rashomon effect, 2026. arXiv:2601.06730.

Tomaso Poggio, Ryan Rifkin, Sayan Mukherjee, and Partha Niyogi. General conditions for predictivity in learning theory. *Nature*, 428:419–422, 2004.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11(90):2635–2670, 2010.

Jake A. Soloff, Rina Foygel Barber, and Rebecca Willett. Building a stable classifier with the inflated argmax. In *Advances in Neural Information Processing Systems*, volume 37, pp. 70349–70380, 2024a.

Jake A. Soloff, Rina Foygel Barber, and Rebecca Willett. Bagging provides assumption-free stability. *Journal of Machine Learning Research*, 25(131):1–35, 2024b.

Rui Xin, Chudi Zhong, Zhi Chen, Takuya Takagi, Margo Seltzer, and Cynthia Rudin. Exploring the whole rashomon set of sparse decision trees. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14071–14084, 2022.

Bin Yu. Stability. *Bernoulli*, 19(4):1484–1500, 2013.

A Appendix

A.1 Figures

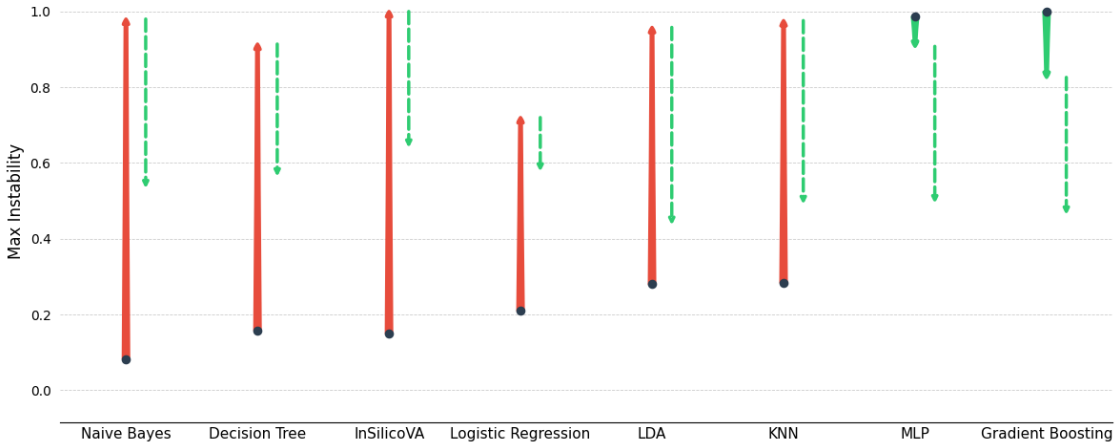
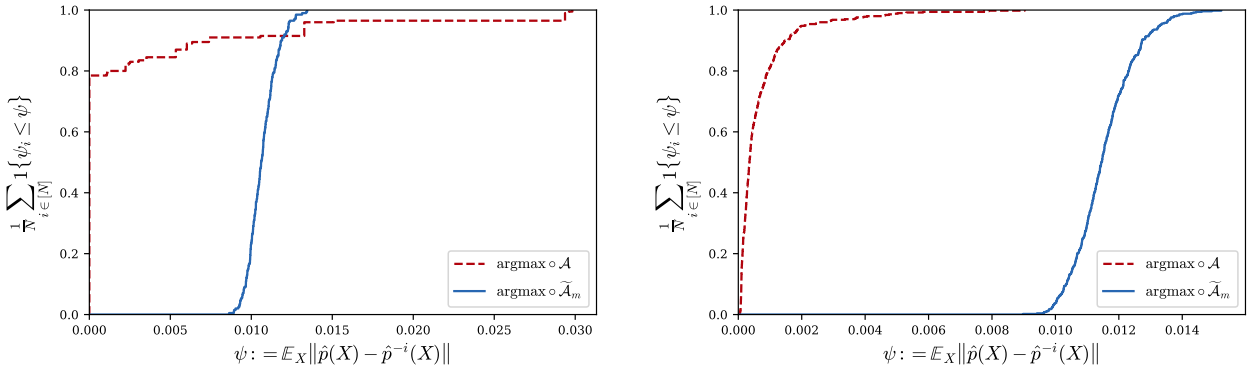
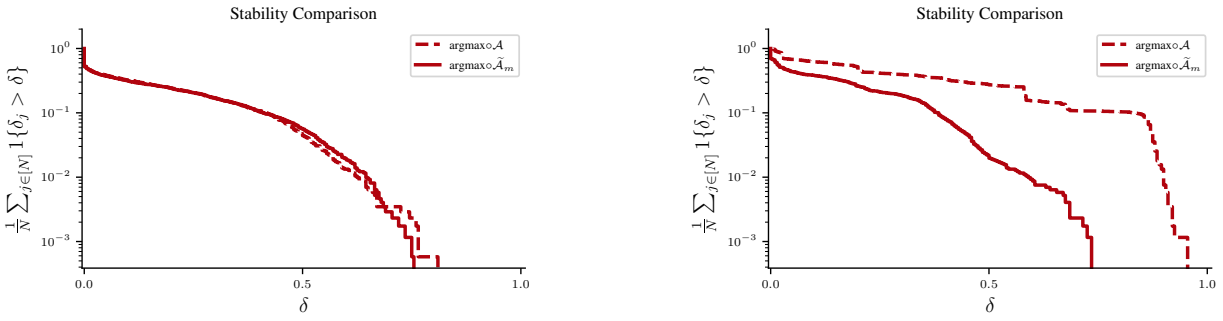


Figure 6: Comparing the effects of subbagging on maximum instability ($\beta_{\text{max-instability}}$) with Jaccard instability. The dots show the maximum instability of the base algorithm ($\mathcal{A}$), solid arrows point to the maximum instability of the subbagged algorithm ($\widetilde{\mathcal{A}}_m$), and dashed arrows point to the maximum instability of the ε-inflated subbagged algorithm ($\widetilde{\mathcal{A}}_m^{\varepsilon}$). Under the Jaccard definition of instability, the inflated argmax does not provide as significant a reduction in instability as it does with the disjoint or overlap definitions.



(a) InSilicoVA (b) Naive Bayes

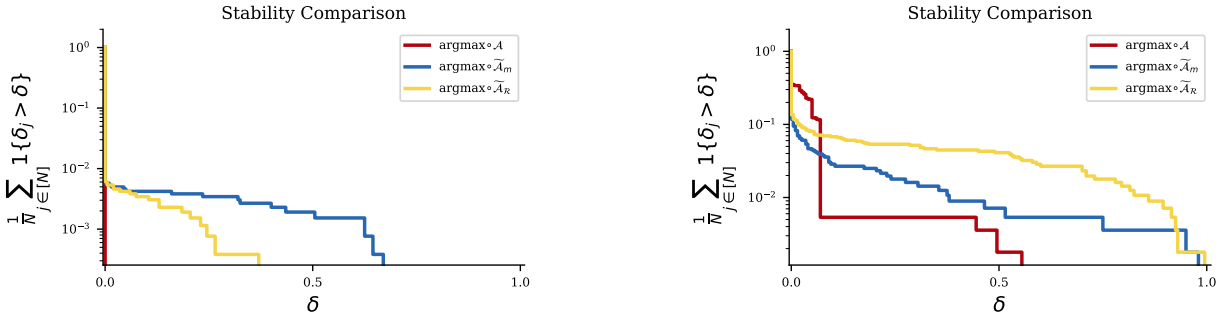
Figure 7: Empirical cumulative distribution function of influence as defined by $\mathbb{E}_X\|\hat{p}(X) - \hat{p}^{-i}(X)\|$ for unbagged (dashed) and subbagged (solid) algorithms. Influence equalization does not imply influence reduction since these are examples where the instability increases with subbagging even though there is a clear equalization effect happening.



(a) Naive Bayes

(b) Regularized Decision Tree

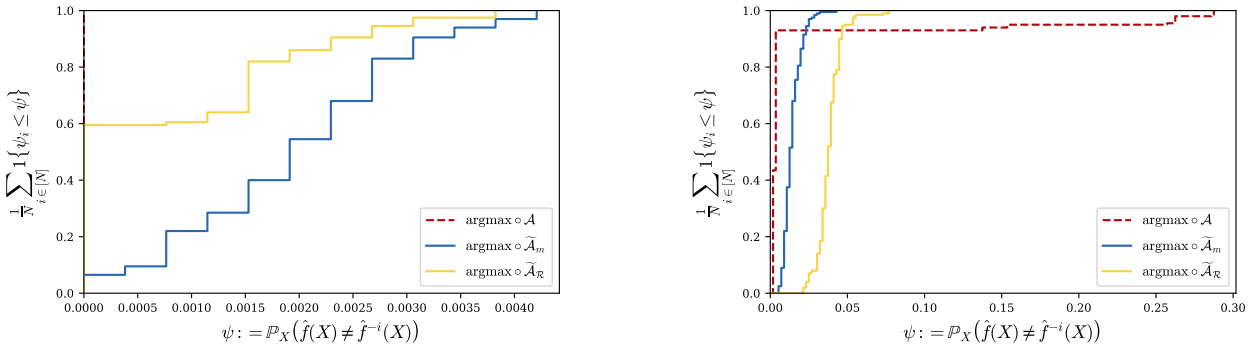
Figure 8: Instability plots for jackknife stable classifiers with bootstrap perturbations. The curves show the proportion of samples in the test set with instability greater than δ for unbagged (dashed) and subbagged (solid) algorithms.



(a) HELOC Dataset

(b) Synthetic Dataset

Figure 9: Instability plots from leave-one-out perturbations with the Rashomon ensembles based on 5,000 trees. The curves show the proportion of samples in the test set with instability greater than δ.



(a) HELOC Dataset

(b) Synthetic Dataset

Figure 10: Empirical cumulative distribution function of influence as defined by $\mathbb{P}_X(\hat{f}(X) \neq \hat{f}^{-i}(X))$ for unbagged (red dashed), subbagged (blue solid), and Rashomon ensemble (yellow solid) algorithms for leave-one-out perturbations. The Rashomon ensemble generates an equalization effect similar to that of subbagging. However, this does appear to break down for the HELOC example that is fully jackknife stable, though it may be that this jackknife definition of influence is simply a poor proxy in this scenario.

A.2 Tables

We can also measure the output stability more directly by considering the following.

$$\beta_{\text{eucl-instability}} := \frac{1}{N} \sum_{j=1}^{N} \frac{1}{K_{\text{loo}}} \sum_{k=1}^{K_{\text{loo}}} \left\| \hat{p}(\tilde{X}_j) - \hat{p}^{\setminus i_k}(\tilde{X}_j) \right\| , \tag{12}$$

which is the average Euclidean distance between the prediction probability vector from the model trained on the full dataset and that of the model trained on each leave-one-out dataset. In our experiments, this output instability roughly mirrored the selection stability, although this will not always be the case.

A.2.1 Unstable Classifiers

Table 1: Summary results from the verbal autopsy dataset and Multi-Layer Perceptron (MLP) classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.581 (0.012)	0.581 (0.012)	0.156 (0.003)	0.155 (0.006)	0.986	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.590 (0.012)	0.590 (0.012)	0.010 (0.000)	0.011 (0.002)	0.908	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.570 (0.012)	0.601 (0.012)	0.156 (0.003)	0.122 (0.005)	0.952	1.046 (0.005)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.570 (0.012)	0.610 (0.012)	0.010 (0.000)	0.000 (0.000)	0.022	1.064 (0.006)

Table 2: Summary results from the verbal autopsy dataset and gradient boosting classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.542 (0.012)	0.542 (0.012)	0.080 (0.001)	0.159 (0.008)	1.000	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.568 (0.012)	0.568 (0.012)	0.002 (0.000)	0.003 (0.001)	0.826	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.483 (0.012)	0.594 (0.012)	0.080 (0.001)	0.068 (0.005)	1.000	1.203 (0.011)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.505 (0.012)	0.617 (0.012)	0.002 (0.000)	0.000 (0.000)	0.000	1.155 (0.009)

A.2.2 Stable Classifiers

Table 3: Summary results from the verbal autopsy dataset and InSilicoVA (McCormick et al., 2016) classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.1$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.541 (0.012)	0.541 (0.012)	0.003 (0.000)	0.002 (0.000)	0.150	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.548 (0.012)	0.548 (0.012)	0.018 (0.000)	0.013 (0.002)	1.000	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.525 (0.012)	0.552 (0.012)	0.003 (0.000)	0.001 (0.000)	0.150	1.040 (0.005)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.536 (0.012)	0.568 (0.012)	0.018 (0.000)	0.000 (0.000)	0.045	1.045 (0.005)

Table 4: Summary results from the verbal autopsy dataset and decision tree classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.568 (0.012)	0.568 (0.012)	0.004 (0.000)	0.002 (0.000)	0.158	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.547 (0.012)	0.547 (0.012)	0.018 (0.000)	0.013 (0.002)	0.914	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.564 (0.012)	0.570 (0.012)	0.004 (0.000)	0.001 (0.000)	0.158	1.010 (0.003)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.535 (0.012)	0.565 (0.012)	0.018 (0.000)	0.000 (0.000)	0.190	1.035 (0.005)

Table 5: Summary results from the verbal autopsy dataset and naive Bayes classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.517 (0.012)	0.517 (0.012)	0.001 (0.000)	0.000 (0.000)	0.082	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.516 (0.012)	0.516 (0.012)	0.012 (0.000)	0.012 (0.002)	0.980	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.517 (0.012)	0.517 (0.012)	0.001 (0.000)	0.000 (0.000)	0.024	1.001 (0.001)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.509 (0.012)	0.523 (0.012)	0.012 (0.000)	0.000 (0.000)	0.182	1.027 (0.004)

Table 6: Summary results from the verbal autopsy dataset and K-Nearest Neighbors (KNN) classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.385 (0.012)	0.385 (0.012)	0.000 (0.000)	0.000 (0.000)	0.284	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.395 (0.012)	0.395 (0.012)	0.011 (0.000)	0.019 (0.002)	0.976	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.336 (0.011)	0.432 (0.012)	0.000 (0.000)	0.000 (0.000)	0.223	1.215 (0.011)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.368 (0.012)	0.421 (0.012)	0.011 (0.000)	0.000 (0.000)	0.028	1.114 (0.008)

Table 7: Summary results from the verbal autopsy dataset and Linear Discriminant Analysis (LDA) classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.645 (0.012)	0.645 (0.012)	0.001 (0.000)	0.001 (0.000)	0.280	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.646 (0.012)	0.646 (0.012)	0.005 (0.000)	0.005 (0.001)	0.958	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.635 (0.012)	0.655 (0.011)	0.001 (0.000)	0.000 (0.000)	0.004	1.028 (0.004)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.638 (0.012)	0.657 (0.011)	0.005 (0.000)	0.000 (0.000)	0.005	1.030 (0.004)

Table 8: Summary results from the verbal autopsy dataset and logistic regression classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.593 (0.012)	0.593 (0.012)	0.001 (0.000)	0.001 (0.000)	0.210	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.604 (0.012)	0.604 (0.012)	0.006 (0.000)	0.005 (0.001)	0.720	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.584 (0.012)	0.609 (0.012)	0.001 (0.000)	0.000 (0.000)	0.004	1.044 (0.005)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.593 (0.012)	0.625 (0.012)	0.006 (0.000)	0.000 (0.000)	0.285	1.053 (0.006)

Table 9: Summary results from the verbal autopsy dataset and a heavily regularized decision tree classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.414 (0.012)	0.414 (0.012)	0.000 (0.000)	0.000 (0.000)	0.004	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.499 (0.012)	0.499 (0.012)	0.010 (0.000)	0.003 (0.001)	0.794	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.409 (0.012)	0.420 (0.012)	0.000 (0.000)	0.000 (0.000)	0.004	1.011 (0.003)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.496 (0.012)	0.509 (0.012)	0.010 (0.000)	0.000 (0.000)	0.000	1.016 (0.003)

A.2.3 Rashomon

Table 10: Summary results from the synthetic dataset and SORTeD decision tree classifier with inflated argmax $\epsilon = 0.1$ and bootstrap perturbations. On the full training dataset the Rashomon set had 6,573 trees, of which 1,373 produced unique prediction vectors, and those were clustered down to 50. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.779 (0.018)	0.779 (0.018)	0.424 (0.014)	0.298 (0.011)	0.940	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.970 (0.007)	0.970 (0.007)	0.108 (0.002)	0.044 (0.004)	0.800	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_{\mathcal{R}}$	0.938 (0.010)	0.938 (0.010)	0.223 (0.004)	0.126 (0.007)	0.920	1.000 (0.000)
argmax$^\varepsilon$	$\mathcal{A}$	0.779 (0.018)	0.779 (0.018)	0.424 (0.014)	0.298 (0.011)	0.940	1.000 (0.000)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_m$	0.957 (0.009)	0.975 (0.007)	0.108 (0.002)	0.024 (0.002)	0.480	1.018 (0.006)
argmax$^\varepsilon$	$\widetilde{\mathcal{A}}_{\mathcal{R}}$	0.920 (0.011)	0.959 (0.008)	0.223 (0.004)	0.096 (0.005)	0.840	1.039 (0.008)

Table 11: Summary results from the FICO HELOC dataset and SORTeD decision tree classifier with inflated argmax $\epsilon = 0.1$ and bootstrap perturbations. On the full training set, the Rashomon set contained 5,497 trees of which 347 produced unique prediction vectors, and those were clustered down to 50. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.858 (0.007)	0.858 (0.007)	0.089 (0.003)	0.058 (0.003)	0.790	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.857 (0.007)	0.857 (0.007)	0.044 (0.001)	0.021 (0.001)	0.640	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_{\mathcal{R}}$	0.858 (0.007)	0.858 (0.007)	0.042 (0.001)	0.019 (0.001)	0.680	1.000 (0.000)
argmax$^{\varepsilon}$	$\mathcal{A}$	0.858 (0.007)	0.858 (0.007)	0.089 (0.003)	0.058 (0.003)	0.790	1.000 (0.000)
argmax$^{\varepsilon}$	$\widetilde{\mathcal{A}}_m$	0.855 (0.007)	0.858 (0.007)	0.044 (0.001)	0.015 (0.001)	0.530	1.002 (0.001)
argmax$^{\varepsilon}$	$\widetilde{\mathcal{A}}_{\mathcal{R}}$	0.857 (0.007)	0.860 (0.007)	0.042 (0.001)	0.014 (0.001)	0.570	1.003 (0.001)

Table 12: Summary results from the COMPAS Recidivism dataset (Larson et al., 2016) and SORTeD decision tree classifier with inflated argmax $\epsilon = 0.1$ and bootstrap perturbations. On the full training dataset the Rashomon set had 3,660 trees, of which 340 produced unique prediction vectors, and those were clustered down to 50. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired.

Sel. rule	Algo	$\beta_{\text{accuracy}} \nearrow$	$\beta_{\text{coverage}} \nearrow$	$\beta_{\text{eucl-instability}} \searrow$	$\beta_{\text{mean-instability}} \searrow$	$\beta_{\text{max-instability}} \searrow$	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.664 (0.013)	0.664 (0.013)	0.070 (0.002)	0.091 (0.004)	0.620	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.669 (0.013)	0.669 (0.013)	0.042 (0.001)	0.067 (0.003)	0.570	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_{\mathcal{R}}$	0.662 (0.013)	0.662 (0.013)	0.075 (0.002)	0.052 (0.004)	0.690	1.000 (0.000)
argmax$^{\varepsilon}$	$\mathcal{A}$	0.664 (0.013)	0.664 (0.013)	0.070 (0.002)	0.091 (0.004)	0.620	1.000 (0.000)
argmax$^{\varepsilon}$	$\widetilde{\mathcal{A}}_m$	0.660 (0.013)	0.676 (0.013)	0.042 (0.001)	0.035 (0.002)	0.270	1.016 (0.004)
argmax$^{\varepsilon}$	$\widetilde{\mathcal{A}}_{\mathcal{R}}$	0.657 (0.014)	0.666 (0.013)	0.075 (0.002)	0.040 (0.003)	0.610	1.008 (0.003)

A.2.4 Miscellaneous

Table 13: Summary results from the verbal autopsy dataset and Multi-Layer Perceptron (MLP) classifier validated on Dar, Tanzania holdout data with inflated argmax $\epsilon = 0.05$. Standard errors are shown in parentheses and arrow direction denotes higher/lower values are desired. This example demonstrates how the chosen instability definition can alter the effect of the inflated argmax, with the Jaccard definition being noticeably more conservative.

		Mean Instability $\searrow$			Max Instability $\searrow$			
Sel. rule	Algo	Disjoint	Overlap	Jaccard	Disjoint	Overlap	Jaccard	$\beta_{\text{size}} \searrow$
argmax	$\mathcal{A}$	0.155 (0.006)	0.155 (0.006)	0.155 (0.006)	0.986	0.986	0.986	1.000 (0.000)
argmax	$\widetilde{\mathcal{A}}_m$	0.011 (0.002)	0.011 (0.002)	0.011 (0.002)	0.908	0.908	0.908	1.000 (0.000)
argmax$^{\varepsilon}$	$\mathcal{A}$	0.122 (0.005)	0.122 (0.005)	0.156 (0.006)	0.952	0.952	0.960	1.046 (0.005)
argmax$^{\varepsilon}$	$\widetilde{\mathcal{A}}_m$	0.000 (0.000)	0.000 (0.000)	0.012 (0.001)	0.002	0.022	0.493	1.064 (0.006)

Table 14: Effect of subbagging on instability for different perturbation levels for a stable classifier. As the level of perturbation increases, the multiplicative effect of subbagging on instability decreases.

Val. Site	n Left Out	Base Instability	Subbagged Instability	Ratio
Bohol	1	0.002	0.029	14.5
Bohol	3	0.007	0.035	5.0
Bohol	5	0.012	0.041	3.4
Dar	1	0.015	0.035	2.3
Dar	3	0.025	0.041	1.6
Dar	5	0.033	0.045	1.4

A.3 Synthetic Dataset

The dataset consists of $n = 300$ samples and $p = 15$ features generated via `make_classification` with 4 informative features, 6 redundant features (linearly correlated with the informative features), and 1 repeated feature. The outcome is binary with class weights $[0.85, 0.15]$, class separability `class_sep` $= 0.25$, and label noise of 10% applied via random flips. Features are standardized to zero mean and unit variance.

To expand the Rashomon set, the training set is pruned to remove samples near the decision boundary. A logistic regression model $\hat{f}$ is fit on the raw training data, and any sample with predicted probability within 0.2 of the boundary is discarded:

$$\mathcal{D}_{\text{train}} = \left\{ (X_i, Y_i) : \left| \hat{f}(X_i) - 0.5 \right| > 0.2 \right\}.$$

The test set is augmented with $n_{\text{ambig}} = 500$ synthetic points drawn uniformly from $[-1, 1]^p$ and labeled by the decision function of $\hat{f}$:

$$X_i^{\text{ambig}} \overset{iid}{\sim} \text{Uniform}([-1, 1]^p), \quad Y_i^{\text{ambig}} = \mathbf{1}\{\hat{f}(X_i^{\text{ambig}}) > 0\}.$$