MOSS Online Supplement

Appendix Sections E, F, G, H, I

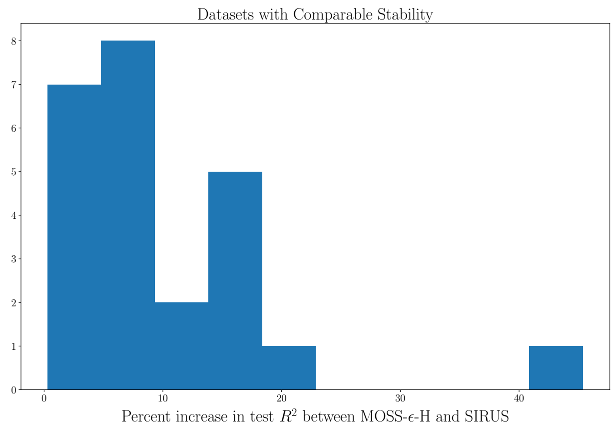

Figure 10: Percent increase in test R^2 between MOSS-ϵ-H and SIRUS on datasets where the stability of the both methods is comparable.

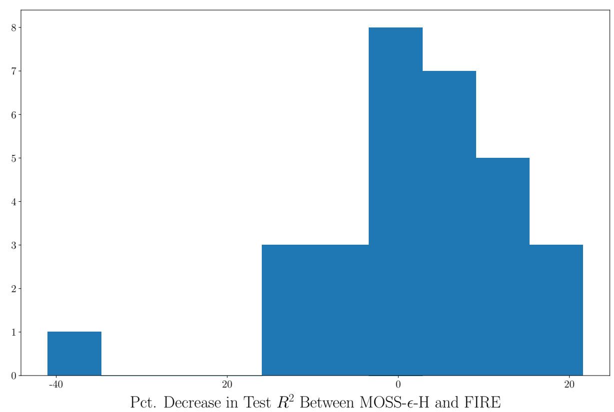

Figure 11: Percent decrease in test R^2 between MOSS-ϵ-H and FIRE.

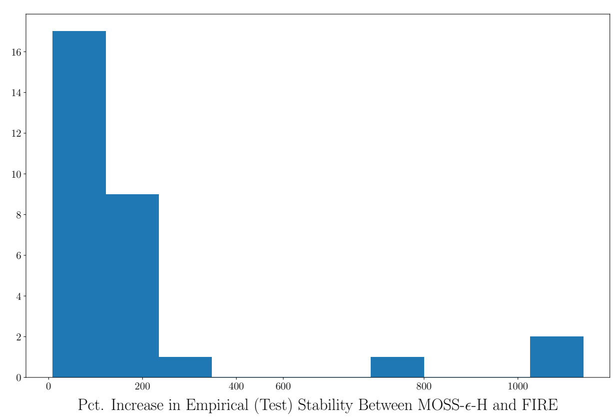

Figure 12: Percent increase in empirical (test) stability between MOSS-ϵ-H and FIRE.

E Experiment Results: Detailed Results

In the pages below, we show tables (3,4,5,6) of our detailed experiment results by dataset. We report out-of-sample R^2 to measure predictive accuracy and average pairwise DSC to measure empirical stability. We report both the means and standard errors of both metrics over the 10-fold CV.

F Experiment Results: Comparisons

One 24 out of 30 of the datasets in our experiment, *MOSS*-ϵ-H and SIRUS have empirical stabilities that are within 1 standard error of each other. These datasets are:

- ESL
- mercedes
- Moneyball
- abalone
- autoMpg
- auto_price
- bank32nh
- bodyfat
- elevators
- kin8nm
- mtp
- no2
- pol
- pollen
- puma32H
- satellite_image
- socmob
- space_ga
- tecator
- us_crime
- vinnie
- visualizing_galaxy
- wind
- wine_quality

On these datasets, we observe than MOSS-ϵ-H produces much more accurate rule sets, with an 10 percent increase of $10R^2$. We show the distribution of this percent increase in test performance, between MOSS-ϵ-H and SIRUS, for these datasets where the rule sets produced have comparable empirical stability, Figure 10.

We also compare MOSS-ϵ-H against FIRE, the competing method with the best accuracy. We observe that the on average, $MOSS-\epsilon$-H has a 2 percent decrease in test R^2 compared to FIRE, we show the distribution in Figure 11.

However, the models produced by *MOSS*-ϵ-H are much more stable, with an average percent increase in empirical stability (our average pairwise DSC metric) of 190 percent. We show the distribution of this if Figure 12.

Dataset	MOSS-ϵ-H	MOSS-ϵ-M	MOSS-ϵ-L	FIRE	GLRM	SIRUS	RULEFIT
Ailerons_296	0.639792	0.656669	0.670918	0.675325	0.324151	0.569526	0.546867
ESL_1027	0.801289	0.828360	0.801771	0.814266	0.845445	0.756764	0.672672
mercedes_42570	0.572542	0.556255	0.576232	0.574502	0.412054	0.570803	0.566909
Moneyball_41021	0.869957	0.865513	0.871525	0.871419	0.928186	0.856857	0.683138
abalone_183	0.376287	0.393225	0.447597	0.453783	0.508449	0.360119	0.348036
autoMpg_196	0.747425	0.768764	0.772191	0.789858	0.798562	0.710980	0.610615
auto_price_195	0.863399	0.815007	0.672240	0.509488	0.747078	0.738998	0.693806
bank32nh_558	0.401371	0.399995	0.438601	0.435068	0.419061	0.370140	0.247573
bodyfat_560	0.936925	0.938313	0.849033	0.840143	0.948397	0.907751	0.858232
cpu_small_227	0.912807	0.923339	0.927361	0.922691	0.536490	0.878285	0.859981
elevators_216	0.410628	0.452588	0.494705	0.499322	0.551810	0.387546	0.355308
house_16H_574	0.365959	0.400449	0.404487	0.400990	0.334073	0.339442	0.248186
houses_537	0.534771	0.548320	0.586357	0.584077	0.563709	0.494428	0.472886
kin8nm_189	0.389164	0.405103	0.460511	0.456236	0.393324	0.340757	0.294860
mtp_405	0.318501	0.336691	0.357685	0.354267	0.359294	0.296157	0.183265
mv_344	0.914857	0.928591	0.930240	0.937472	0.963350	0.847946	0.890945
no2_547	0.428888	0.479166	0.472051	0.458165	0.516820	0.381181	0.382920
pol_201	0.750334	0.723749	0.779218	0.772502	0.198512	0.700354	0.569024
pollen_529	0.521309	0.486982	0.601252	0.591175	0.558306	0.438864	0.380181
puma32H_308	0.656846	0.651822	0.669697	0.665608	0.204270	0.451785	0.606380
satellite_image_294	0.706395	0.748535	0.744949	0.743465	0.733783	0.670141	0.452368
socmob_541	0.669874	0.611381	0.637742	0.608317	0.677353	0.582189	0.643831
space_ga_507	0.502022	0.520441	0.454429	0.445174	0.520454	0.439996	0.405508
stock_223	0.885754	0.907392	0.889649	0.905912	0.850638	0.818754	0.865049
tecator_505	0.959962	0.954073	0.827610	0.850458	0.972255	0.946702	0.857634
us_crime_315	0.578675	0.597916	0.607469	0.602808	0.351041	0.553025	0.372902
vinnie_519	0.698352	0.704060	0.608716	0.635216	0.716773	0.596746	0.706372
visualizing_galaxy_690	0.930156	0.947740	0.837415	0.871342	0.957811	0.876663	0.913459
wind_503	0.669185	0.670398	0.685995	0.679164	0.746188	0.642927	0.410033
wine_quality_287	0.278516	0.272298	0.307583	0.309223	0.291475	0.246417	0.227999
Average Rank	**3.9**	**3.4**	**2.8**	**3.1**	**3.0**	**5.7**	**6.0**

Table 3: Average Test R^2 by dataset: We show here the out-of-sample performance, measured using test R^2, for all datasets in our experiment. These results are the averages, obtained over a 10-fold CV. Standard errors are shown in Table 4

.

Dataset	MOSS-ϵ-H	MOSS-ϵ-M	MOSS-ϵ-L	FIRE	GLRM	SIRUS	RULEFIT
Ailerons_296	0.002285	0.002516	0.002836	0.002821	0.00415	0.002121	0.00287
ESL_1027	0.005362	0.004986	0.004822	0.003459	0.003859	0.00531	0.009543
mercedes_42570	0.007692	0.006812	0.00813	0.006828	0.000123	0.007624	0.007663
Moneyball_41021	0.001384	0.001677	0.001983	0.001883	0.000951	0.001561	0.005223
abalone_183	0.004065	0.003866	0.002029	0.00233	0.001936	0.00325	0.003582
autoMpg_196	0.007127	0.007724	0.004803	0.007234	0.007407	0.006754	0.005385
auto_price_195	0.004301	0.012749	0.020409	0.023252	0.011682	0.014493	0.013724
bank32nh_558	0.002263	0.002518	0.001691	0.001815	0.0034	0.002477	0.00371
bodyfat_560	0.003717	0.004767	0.008932	0.009656	0.002343	0.002949	0.005828
cpu_small_227	0.001989	0.001869	0.001205	0.001792	0.002345	0.002134	0.003018
elevators_216	0.002923	0.002512	0.002744	0.001783	0.006588	0.002986	0.002997
house_16H_574	0.002709	0.002277	0.003425	0.003237	0.001564	0.002798	0.003822
houses_537	0.001252	0.00136	0.002994	0.002617	0.002434	0.001083	0.001938
kin8nm_189	0.002086	0.001958	0.002858	0.00196	0.00322	0.002149	0.000992
mtp_405	0.005264	0.005478	0.003156	0.003498	0.003801	0.004378	0.002034
mv_344	0.000899	0.000281	0.000253	0.000357	0.000604	0.00063	0.001155
no2_547	0.014793	0.012096	0.011815	0.013442	0.010413	0.017488	0.010606
pol_201	0.001823	0.003061	0.002274	0.001908	0.002216	0.003823	0.003662
pollen_529	0.002837	0.005641	0.004749	0.004218	0.004445	0.005575	0.00268
puma32H_308	0.001736	0.002041	0.001943	0.001705	0.002937	0.009204	0.005404
satellite_image_294	0.001982	0.00186	0.00152	0.001805	0.004031	0.002496	0.006522
socmob_541	0.009706	0.009399	0.02505	0.021329	0.007091	0.01136	0.005021
space_ga_507	0.006704	0.006642	0.020675	0.020141	0.00524	0.007398	0.006961
stock_223	0.003355	0.002338	0.002685	0.001824	0.004715	0.004208	0.002727
tecator_505	0.001079	0.001199	0.0135	0.010467	0.000717	0.001254	0.00656
us_crime_315	0.005858	0.006185	0.00602	0.005627	0.006231	0.007665	0.005101
vinnie_519	0.01316	0.011878	0.010362	0.011215	0.009029	0.018975	0.011436
visualizing_galaxy_690	0.002371	0.002072	0.008285	0.006323	0.002107	0.004603	0.002173
wind_503	0.002286	0.002195	0.002025	0.002219	0.001447	0.002937	0.004641
wine_quality_287	0.002742	0.001916	0.002367	0.002378	0.00202	0.002405	0.002517

Table 4: Standard Error of Test R^2 by dataset. These standard errors are obtained over a 10-fold CV. The mean test R^2 of our performance results are shown in Table 3

Dataset	MOSS-ϵ-H	MOSS-ϵ-M	MOSS-ϵ-L	FIRE	GLRM	SIRUS	RULEFIT
Ailerons_296	0.519373	0.216654	0.252604	0.242296	0.342012	0.711917	0.230505
ESL_1027	0.539919	0.422515	0.505706	0.232644	0.277109	0.655401	0.424257
mercedes_42570	0.622377	0.493946	0.426768	0.317824	0.314512	0.673675	0.263956
Moneyball_41021	0.575336	0.341164	0.496633	0.165632	0.245471	0.667643	0.256258
abalone_183	0.414021	0.34291	0.296429	0.263169	0.365313	0.497989	0.18037
autoMpg_196	0.538893	0.260317	0.417147	0.22853	0.145074	0.577619	0.161384
auto_price_195	0.289273	0.153148	0.358583	0.025644	0.05346	0.264154	0.076101
bank32nh_558	0.603915	0.373862	0.11825	0.235648	0.622924	0.68127	0.411111
bodyfat_560	0.657985	0.316479	0.657835	0.198854	0.188848	0.708759	0.193073
cpu_small_227	0.632039	0.424135	0.366626	0.323713	0.468433	0.798771	0.583913
elevators_216	0.459524	0.399735	0.200121	0.329536	0.306148	0.591005	0.384127
house_16H_574	0.680887	0.324827	0.299862	0.271005	0.610905	0.777729	0.642267
houses_537	0.730639	0.501701	0.408506	0.580154	0.657892	0.808856	0.665739
kin8nm_189	0.34328	0.39418	0.174172	0.2484	0.189194	0.376931	0.364127
mtp_405	0.544872	0.265218	0.151811	0.168422	0.109098	0.629402	0.332087
mv_344	0.682011	0.631429	0.766772	0.546314	0.650213	0.878519	0.693439
no2_547	0.292112	0.283675	0.2304	0.170359	0.24391	0.295515	0.281209
pol_201	0.756046	0.567302	0.60816	0.698832	0.518501	0.814867	0.646029
pollen_529	0.685926	0.587831	0.331494	0.398385	0.413704	0.686772	0.477884
puma32H_308	0.632764	0.810834	0.405413	0.386241	0.119466	0.664029	0.679414
satellite_image_294	0.605291	0.334127	0.331583	0.229861	0.347707	0.691534	0.313161
socmob_541	0.695397	0.654392	0.630077	0.557368	0.170435	0.651693	0.533333
space_ga_507	0.485637	0.538771	0.204127	0.349709	0.581781	0.514082	0.363561
stock_223	0.579577	0.541905	0.471184	0.343286	0.618864	0.697354	0.430476
tecator_505	0.656032	0.253386	0.805942	0.079609	0.073071	0.746772	0.159471
us_crime_315	0.428726	0.234546	0.215438	0.034593	0.131451	0.518055	0.14915
vinnie_519	0.641058	0.566561	0.62381	0.293618	0.47808	0.699577	0.346772
visualizing_galaxy_690	0.594709	0.424127	0.526939	0.182653	0.106043	0.713228	0.288042
wind_503	0.524868	0.237302	0.326007	0.192867	0.300686	0.611111	0.311508
wine_quality_287	0.469499	0.443488	0.247871	0.345134	0.390176	0.530989	0.325507
Average Rank	**2.3**	**3.9**	**4.6**	**5.9**	**5.3**	**1.3**	**4.6**

Table 5: Average Empirical Stability (Measured using Average Pairwise DSC) by dataset: We show here the average empirical stability for all datasets in our experiments. These estimates were obtained over a 10-fold CV. Standard errors are shown in Table 6

Dataset	MOSS-ϵ-H	MOSS-ϵ-M	MOSS-ϵ-L	FIRE	GLRM	SIRUS	RULEFIT
Ailerons_296	0.118842	0.079678	0.102419	0.104818	0.123415	0.120866	0.088508
ESL_1027	0.172018	0.164152	0.117612	0.111256	0.20201	0.238199	0.169534
mercedes_42570	0.084819	0.078256	0.084641	0.137956	0.13561	0.13074	0.083186
Moneyball_41021	0.109747	0.098393	0.148511	0.094008	0.208727	0.094479	0.10799
abalone_183	0.132611	0.118806	0.12475	0.114168	0.177402	0.149801	0.090511
autoMpg_196	0.123852	0.074444	0.180301	0.132714	0.100081	0.151147	0.109536
auto_price_195	0.159446	0.091233	0.138117	0.040316	0.093504	0.16397	0.081988
bank32nh_558	0.090781	0.084361	0.116622	0.171948	0.091741	0.114701	0.104743
bodyfat_560	0.100415	0.091985	0.093507	0.10126	0.094722	0.113837	0.112233
cpu_small_227	0.107813	0.085746	0.143522	0.119897	0.223344	0.098827	0.080084
elevators_216	0.181387	0.112785	0.134456	0.142449	0.137291	0.219673	0.101059
house_16H_574	0.065372	0.105668	0.101722	0.143274	0.104602	0.087089	0.076819
houses_537	0.068961	0.107299	0.130898	0.113962	0.144677	0.070389	0.109541
kin8nm_189	0.135261	0.08342	0.09613	0.119301	0.107888	0.170825	0.117471
mtp_405	0.159357	0.095143	0.099004	0.104523	0.095361	0.172323	0.133667
mv_344	0.073298	0.099027	0.119891	0.11089	0.167417	0.061751	0.073664
no2_547	0.14432	0.092084	0.134929	0.110993	0.137453	0.144827	0.112705
pol_201	0.06828	0.078275	0.212616	0.107391	0.265527	0.082779	0.111208
pollen_529	0.078259	0.05903	0.181264	0.101756	0.170623	0.144676	0.101434
puma32H_308	0.127927	0.081662	0.108145	0.143083	0.061426	0.120889	0.110327
satellite_image_294	0.106714	0.109409	0.154514	0.123443	0.15882	0.113462	0.129678
socmob_541	0.097516	0.094177	0.12578	0.10664	0.066391	0.102278	0.107464
space_ga_507	0.112121	0.096492	0.137084	0.110447	0.186371	0.094293	0.091464
stock_223	0.090338	0.115773	0.105315	0.118745	0.176057	0.111832	0.078487
tecator_505	0.070615	0.071412	0.086415	0.07219	0.084201	0.11017	0.082183
us_crime_315	0.188118	0.097784	0.105851	0.041837	0.120541	0.185771	0.093401
vinnie_519	0.103782	0.085893	0.126296	0.152109	0.164701	0.132472	0.139373
visualizing_galaxy_690	0.124876	0.106514	0.139881	0.096474	0.156731	0.114932	0.094652
wind_503	0.120254	0.118262	0.107745	0.133936	0.183295	0.129051	0.112851
wine_quality_287	0.105566	0.098378	0.127814	0.142321	0.189997	0.146662	0.118966

Table 6: Standard Error of Empirical Stability by dataset: We show here the standard error of our stability estimates, computed across a 10-fold CV. We report average stability in Table 5

G Sensitivity Analyses

In this section, we analyze the sensitivity of MOSS to parameters γ and k.

G.1 ***Parameter*** γ. This parameter controls the ridge regularization penalty in accuracy objective $H_2(z)$. We follow the procedure below to assess the sensitivity of MOSS to γ.

We use the 30 OpenML datasets from our experiments in §4.2 and repeat a 10-fold CV on each dataset. On each training fold, we use random forests to generate $m \sim 10^3$ candidate decision rules. We then apply MOSS to construct rule sets with 15 decision rules. We repeat this procedure across all folds and datasets in our experiment while varying $\gamma \in \{0.0001, 0.0005, 0.001, 0.005\}$. For each value of γ we record the average predictive performance (out-of-sample R^2) and empirical stability (average pairwise Dice-Sorensen coefficient) of the rule sets. We report the results of this sensitivity analysis in Table 7. From this table, we observe that the accuracy of MOSS (out-of-sample R^2) is relatively *insensitive* to γ. Across all datasets, the rules sets constructed with varying values of γ achieve similar R^2 scores. We also note that increasing γ appears to decrease the empirical stability of the rule sets, by a slight degree.

Our results here suggest that the performance of MOSS in terms of empirical stability and accuracy is relatively insensitive to parameter γ. As an aside, we also note that γ influences the computation time of our cutting plane algorithm. When γ is large, Algorithm 1 requires more iterations to converge. As such, we recommend setting γ to a small value around 10^{-2} or 10^{-3} as a default.

G.2 ***Parameter*** k. The parameter k controls the size of the rule sets constructed by MOSS. It is important to restrict k to be small so that the rule sets remain interpretable; in fact, [5] and [21] restrict rule sets to contain < 20 rules to remain human readable. We use the procedure below to assess the sensivity of MOSS to k

We employ the same 30 OpenML datasets from our experiments in §4.2 and perform 10-fold cross-validation on each dataset. On each training fold we generate m 10^3 candidate rules and then we apply MOSS to construct rule sets. We set $\gamma = 0.001$ vary $k \in \{5, 10, 15, 20, 25\}$. For each value of k we record the average out-of-sample R^2 score and the empirical stability of the rule sets, averaged across all folds for each dataset. We report the results of this sensitivity analysis in Table 8.

From Table 8, we observe that as k increases, both the average out-of-sample R^2 and the empirical stability of the rule sets improve. Very compact rule sets ($k = 5$) exhibit significantly worse accuracy and stability compared to larger rule sets ($k > 10$). While larger rule sets perform better in terms of accuracy and stability, they are less interpretable. Based on these results, we recommend setting k to 10, 15, or 20 rules for MOSS to achieve a balance between accuracy, stability, and interpretability.

Our current experimental results in §4.2 compare MOSS against competing algorithms for $k = 15$ sized rule sets. We repeat our experiments for $k = 10$ and $k = 20$; Figures 13 and 14 show our results for these experiments. From these plots, we again see that MOSS outperforms our competing algorithms jointly in terms of both accuracy and stability, and achieves a balance between the two objectives.

Table 7: Results for sensitivity analysis over parameter γ in MOSS.

Dataset Name	γ	R^2	Stability	Dataset Name	γ	R^2	Stability
auto_price	0.0001	0.6637	0.475	space_ga	0.0001	0.4742	0.4241
	0.0005	0.6348	0.4765		0.0005	0.4776	0.4316
	0.001	0.6455	0.47		0.001	0.4781	0.4264
	0.005	0.6472	0.4763		0.005	0.4817	0.384
tecator	0.0001	0.8987	0.7186	pollen	0.0001	0.4541	0.6702
	0.0005	0.8988	0.7217		0.0005	0.4632	0.6729
	0.001	0.8988	0.7217		0.001	0.4627	0.6654
	0.005	0.8984	0.6922		0.005	0.4784	0.6423
body_fat	0.0001	0.8665	0.5647	abalone	0.0001	0.3636	0.3684
	0.0005	0.8648	0.5612		0.0005	0.3802	0.3317
	0.001	0.865	0.5559		0.001	0.3824	0.3484
	0.005	0.8678	0.5465		0.005	0.3777	0.3434
visualizing_galaxy	0.0001	0.8134	0.5652	Mercedes_Benz	0.0001	0.564	0.737
	0.0005	0.8203	0.545		0.0005	0.5724	0.6888
	0.001	0.8133	0.5275		0.001	0.5728	0.6523
	0.005	0.8295	0.4612		0.005	0.5735	0.6444
vinnie	0.0001	0.5648	0.6211	mtp	0.0001	0.2939	0.5516
	0.0005	0.5639	0.6509		0.0005	0.2966	0.5172
	0.001	0.5641	0.632		0.001	0.2979	0.5126
	0.005	0.5648	0.6039		0.005	0.3022	0.4279
autoMpg	0.0001	0.6703	0.5688	satellite_image	0.0001	0.6813	0.6698
	0.0005	0.6703	0.5654		0.0005	0.6836	0.6254
	0.001	0.6695	0.5638		0.001	0.6992	0.6063
	0.005	0.6759	0.5366		0.005	0.7006	0.6397
ESL	0.0001	0.7574	0.4205	wine_quality	0.0001	0.2578	0.5483
	0.0005	0.7566	0.4285		0.0005	0.2588	0.537
	0.001	0.7611	0.4121		0.001	0.2586	0.5281
	0.005	0.7731	0.4076		0.005	0.2564	0.5564
no2	0.0001	0.3806	0.3293	wind	0.0001	0.6522	0.5411
	0.0005	0.3787	0.3054		0.0005	0.6551	0.5262
	0.001	0.3815	0.3114		0.001	0.6598	0.5325
	0.005	0.3892	0.3158		0.005	0.6601	0.5139
stock	0.0001	0.8213	0.7335	puma32H	0.0001	0.4491	0.5895
	0.0005	0.8284	0.7116		0.0005	0.4693	0.5849
	0.001	0.8367	0.7262		0.001	0.4668	0.6016
	0.005	0.8414	0.7045		0.005	0.4943	0.6027
socmob	0.0001	0.5895	0.7789	bank32nh	0.0001	0.3444	0.6355
	0.0005	0.5837	0.7759		0.0005	0.361	0.59
	0.001	0.5951	0.7654		0.001	0.3618	0.5591
	0.005	0.613	0.7337		0.005	0.391	0.6066
Moneyball	0.0001	0.8333	0.4698	cpu_small	0.0001	0.8902	0.7885
	0.0005	0.8324	0.4547		0.0005	0.8966	0.7579
	0.001	0.8346	0.4707		0.001	0.9022	0.7471
	0.005	0.8419	0.4088		0.005	0.9057	0.744
us_crime	0.0001	0.5371	0.561	kin8nm	0.0001	0.3544	0.4219
	0.0005	0.5607	0.5553		0.0005	0.3447	0.3878
	0.001	0.5603	0.5191		0.001	0.3602	0.3835
	0.005	0.5597	0.5388		0.005	0.3801	0.3585

Dataset Name	γ	R^2	Stability
Ailerons	0.0001	0.5956	0.6722
	0.0005	0.606	0.6721
	0.001	0.6114	0.6802
	0.005	0.6119	0.6747
pol	0.0001	0.6868	0.8923
	0.0005	0.7077	0.8658
	0.001	0.7345	0.818
	0.005	0.7385	0.8203
elevators	0.0001	0.4042	0.5044
	0.0005	0.4056	0.4557
	0.001	0.4085	0.4521
	0.005	0.4185	0.4524
houses	0.0001	0.5025	0.7706
	0.0005	0.5281	0.7889
	0.001	0.5304	0.7643
	0.005	0.5313	0.7493
house_16H	0.0001	0.3652	0.7906
	0.0005	0.3683	0.7537
	0.001	0.3674	0.7137
	0.005	0.3669	0.677
mv	0.0001	0.8932	0.8677
	0.0005	0.9156	0.8006
	0.001	0.9171	0.8129
	0.005	0.9178	0.8049

Table 8: Results for sensitivity analysis over parameter k in MOSS.

Dataset Name	k	R^2	Stability	Dataset Name	k	R^2	Stability
auto_price	5	0.305	0.4418	Moneyball	5	0.6956	0.3833
	10	0.6095	0.5088		10	0.8019	0.501
	15	0.6637	0.475		15	0.8314	0.4918
	20	0.6379	0.5171		20	0.8488	0.5511
	25	0.665	0.5274		25	0.8535	0.5291
tecator	5	0.697	0.5608	us_crime	5	0.4951	0.4
	10	0.8939	0.7066		10	0.5301	0.5551
	15	0.8987	0.7186		15	0.5371	0.561
	20	0.9036	0.6572		20	0.5603	0.5638
	25	0.9075	0.6601		25	0.5646	0.5597
bodyfat	5	0.809	0.6278	space_ga	5	0.3851	0.3594
	10	0.8583	0.6048		10	0.4535	0.4059
	15	0.8665	0.5647		15	0.4742	0.4241
	20	0.8655	0.5126		20	0.475	0.4395
	25	0.8674	0.52		25	0.4814	0.4664
visualizing_galaxy	5	0.6876	0.3067	pollen	5	0.3377	0.5758
	10	0.8018	0.5083		10	0.3896	0.6755
	15	0.8134	0.5652		15	0.4541	0.6702
	20	0.8447	0.5144		20	0.4986	0.6835
	25	0.8344	0.4785		25	0.5001	0.6815
vinnie	5	0.4535	0.2411	abalone	5	0.2971	0.3056
	10	0.5683	0.4605		10	0.3518	0.3159
	15	0.5648	0.6211		15	0.3636	0.3684
	20	0.5766	0.6514		20	0.3725	0.4283
	25	0.593	0.7375		25	0.3749	0.4625
autoMpg	5	0.585	0.3689	Mercedes_Benz	5	0.4533	0.622
	10	0.6649	0.5461		10	0.5139	0.6763
	15	0.677	0.5721		15	0.5596	0.7789
	20	0.667	0.5395		20	0.5744	0.7735
	25	0.6647	0.5604		25	0.5744	0.7741
ESL	5	0.6378	0.2533	mtp	5	0.2438	0.4286
	10	0.7222	0.3706		10	0.2849	0.4848
	15	0.7574	0.4205		15	0.2939	0.5516
	20	0.7609	0.3887		20	0.2967	0.5235
	25	0.7832	0.4377		25	0.305	0.5252
no2	5	0.275	0.212	satellite_image	5	0.5849	0.413
	10	0.3549	0.2908		10	0.6647	0.6721
	15	0.3806	0.3293		15	0.6813	0.6698
	20	0.3563	0.3614		20	0.6927	0.6854
	25	0.3625	0.3922		25	0.6929	0.7176
stock	5	0.7265	0.668	wine_quality	5	0.223	0.5164
	10	0.8007	0.6531		10	0.2512	0.5585
	15	0.8213	0.7335		15	0.2578	0.5483
	20	0.8445	0.7309		20	0.2676	0.5849
	25	0.8417	0.715		25	0.2745	0.6586
socmob	5	0.452	0.4278	wind	5	0.5662	0.4426
	10	0.5742	0.6578		10	0.6365	0.481
	15	0.5895	0.7789		15	0.6522	0.5411
	20	0.6511	0.8299		20	0.661	0.5683
	25	0.6516	0.8772		25	0.6674	0.6011

Dataset Name	k	R^2	Stability
puma32H	5	0.2995	0.5611
	10	0.3882	0.6184
	15	0.4491	0.5895
	20	0.4888	0.5856
	25	0.5198	0.5791
bank32nh	5	0.2409	0.5289
	10	0.3364	0.617
	15	0.3444	0.6355
	20	0.3763	0.631
	25	0.3845	0.6459
cpu_small	5	0.7487	0.8
	10	0.8511	0.8267
	15	0.8902	0.7885
	20	0.8956	0.8143
	25	0.9017	0.8016
kin8nm	5	0.2427	0.2961
	10	0.3249	0.377
	15	0.3544	0.4219
	20	0.3698	0.4399
	25	0.3826	0.4887
Ailerons	5	0.5018	0.6537
	10	0.5707	0.6423
	15	0.5956	0.6722
	20	0.6105	0.7177
	25	0.6138	0.7105
pol	5	0.5539	1.0
	10	0.668	0.9156
	15	0.6868	0.8923
	20	0.7157	0.9148
	25	0.747	0.9127
elevators	5	0.3471	0.3426
	10	0.3945	0.5006
	15	0.4042	0.5044
	20	0.408	0.5216
	25	0.4114	0.5319
houses	5	0.4159	0.7268
	10	0.4903	0.7339
	15	0.5025	0.7706
	20	0.5198	0.7719
	25	0.5336	0.7801
house_16H	5	0.2871	0.5215
	10	0.3413	0.7106
	15	0.3652	0.7906
	20	0.3708	0.7887
	25	0.3788	0.752
mv	5	0.8005	0.816
	10	0.8467	0.8067
	15	0.9149	0.8391
	20	0.9214	0.8401
	25	0.9201	0.8424

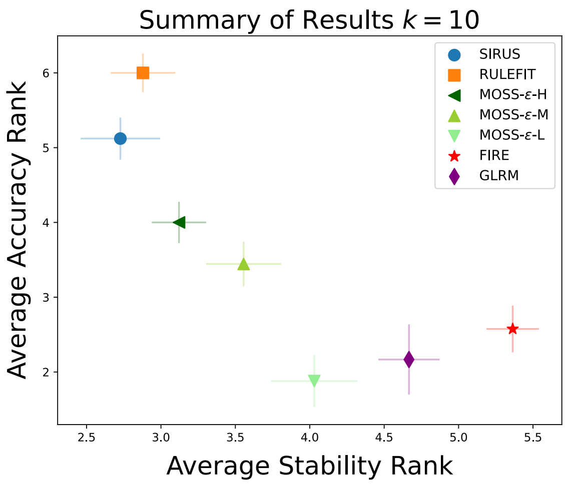

Figure 13: Experimental results for size $k = 10$ rule sets. MOSS methods are able to compute the Pareto frontier between accuracy and stability.

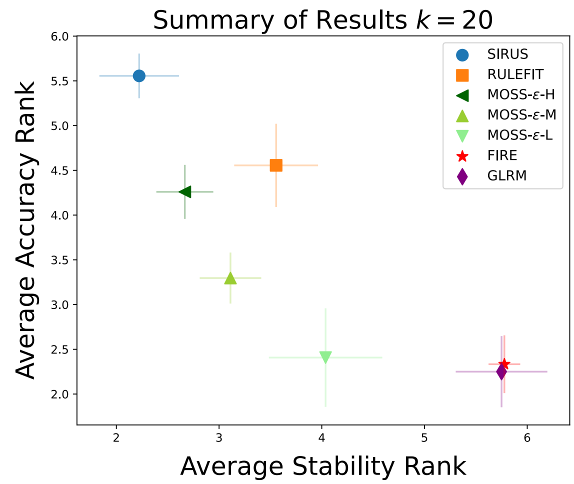

Figure 14: Experimental results for size $k = 20$ rule sets.

H Choice of Stability Measure

In §2.1 of the main text of our paper, we present our approach to asses the stability of rule algorithms. We use the Dice-Sorensens coefficient to measure the similarity between a pair of rule sets, R_i and R_j, and we take the average pairwise coefficient across all rule sets to be the empirical stability of the algorithm.

We note here that we can use various alternative measures to assess the stability of our rule algorithms, and we show that our experimental results are *insensitive* to our choice of stability measure. We evaluate the following metrics [26].

$$\text{Jaccard}(R_i, R_j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|}$$

$$\text{Ochiai}(R_i, R_j) = \frac{|R_i \cap R_j|}{\sqrt{|R_i||R_j|}}$$

$$\text{POG}(R_i, R_j) = \frac{|R_i \cap R_j|}{|R_i|}$$

For each metric, we again take the average pairwise similarity between rule sets as the empirical stability of the rule algorithm.

We repeat our experimental evaluation of MOSS (§4.2) and report our stability results using the measures discussed above. We summarize the results of this analysis in Table 9.

Metric	MOSS-H	MOSS-M	MOSS-L	FIRE	GLRM	SIRUS	RuleFit
DSC	2.3	3.9	4.6	5.9	5.3	1.3	4.6
Jaccard	2.5	4.0	4.5	5.8	5.2	1.4	4.4
Ochiai	2.5	3.8	4.6	5.8	5.3	1.4	4.3
POG	2.4	3.9	4.6	5.9	5.2	1.4	4.3

Table 9: Average stability ranking for each method in our experimental evaluation of MOSS using different stability measures.

In this table, we present the average empirical stability ranking of each algorithm considered in our experiment, across all datasets. The first row reports stability measured using our Dice-Sorensen coefficient metrics, which represents the main result of our paper as shown in §7. The subsequent rows display the stability rankings measured using our alternative metrics. This table demonstrates that our findings remain consistent regardless of the metrics used.

We report dataset-level stability results for each metric in the pages below (Tables 10,11,12).

Dataset Name	MOSS-H	MOSS-M	MOSS-L	FIRE	GLRM	SIRUS	RuleFit
Ailerons_296	0.37991	0.129643	0.159893	0.142019	0.13491	0.601258	0.13888
ESL_1027	0.412596	0.297479	0.3629	0.136341	0.177267	0.564619	0.299636
Mercedes_Benz_42570	0.478343	0.354673	0.296266	0.196763	0.183194	0.548374	0.160499
Moneyball_41021	0.423393	0.214538	0.366631	0.093294	0.158887	0.523579	0.155207
abalone_183	0.296459	0.231639	0.206711	0.156669	0.238788	0.382027	0.110963
autoMpg_196	0.405238	0.155535	0.299937	0.135751	0.081399	0.457141	0.096783
auto_price_195	0.197105	0.09418	0.251944	0.013422	0.030097	0.178004	0.044721
bank32nh_558	0.451013	0.238251	0.071243	0.14544	0.459234	0.543302	0.270109
bodyfat_560	0.536347	0.202649	0.54129	0.114032	0.107355	0.604153	0.117933
cpu_small_227	0.495879	0.285403	0.244834	0.199628	0.332819	0.716664	0.432053
elevators_216	0.336396	0.27418	0.129023	0.206686	0.188657	0.484291	0.262566
house_16H_574	0.547808	0.207961	0.187306	0.164832	0.448287	0.682481	0.50674
houses_537	0.617404	0.360401	0.282309	0.418224	0.507125	0.728404	0.537911
kin8nm_189	0.222827	0.26181	0.108051	0.147727	0.108419	0.253536	0.24064
mtp_405	0.414122	0.164794	0.089602	0.095773	0.060559	0.515025	0.217215
mv_344	0.542536	0.479759	0.657076	0.384421	0.505367	0.82421	0.557431
no2_547	0.189344	0.178384	0.154659	0.097312	0.146761	0.193289	0.177325
pol_201	0.686128	0.442294	0.508667	0.5478	0.396808	0.782373	0.538459
pollen_529	0.540905	0.428081	0.224378	0.253962	0.275593	0.553185	0.325646
puma32H_308	0.509926	0.747302	0.275336	0.249453	0.064664	0.549834	0.568629
satellite_image_294	0.475034	0.219235	0.221151	0.135448	0.222828	0.584153	0.204638
socmob_541	0.585336	0.524908	0.510845	0.393824	0.094648	0.53146	0.400179
space_ga_507	0.353204	0.402592	0.136382	0.21755	0.432612	0.379037	0.243522
stock_223	0.438271	0.400944	0.336365	0.213751	0.471414	0.580025	0.291354
tecator_505	0.524734	0.154821	0.717786	0.043018	0.04003	0.653023	0.093625
us_crime_315	0.306967	0.143614	0.134752	0.018068	0.139102	0.391001	0.086693
vinnie_519	0.49837	0.414396	0.488711	0.181717	0.330411	0.575668	0.224538
visualizing_galaxy_690	0.456523	0.285521	0.402726	0.1038	0.064846	0.598021	0.178512
wind_503	0.399667	0.150179	0.207135	0.113408	0.190706	0.492619	0.208961
wine_quality_287	0.350762	0.324142	0.156606	0.217912	0.262628	0.42454	0.223012

Table 10: Average empirical stability measured using Jaccard metric.

Dataset	MOSS-H	MOSS-M	MOSS-L	FIRE	GLRM	SIRUS	RuleFit
Ailerons_296	0.541128	0.225864	0.269797	0.243849	0.230194	0.741071	0.239569
ESL_1027	0.562471	0.440303	0.523019	0.232895	0.280307	0.683056	0.442317
Mercedes_Benz_42570	0.642431	0.520788	0.453549	0.318953	0.321940	0.695512	0.272624
Moneyball_41021	0.586868	0.347744	0.521799	0.166219	0.255783	0.681284	0.261877
abalone_183	0.446886	0.368871	0.335289	0.265716	0.36733	0.537703	0.195091
autoMpg_196	0.568815	0.266127	0.439245	0.22938	0.145534	0.611416	0.169293
auto_price_195	0.313874	0.168083	0.390849	0.025765	0.054132	0.286174	0.082039
bank32nh_558	0.615667	0.38057	0.124233	0.235813	0.624428	0.694269	0.418547
bodyfat_560	0.691549	0.332096	0.69787	0.199768	0.190693	0.744134	0.203302
cpu_small_227	0.655048	0.439824	0.379241	0.324052	0.469916	0.828172	0.599613
elevators_216	0.485825	0.425721	0.217833	0.332422	0.310246	0.624132	0.412466
house_16H_574	0.706036	0.337601	0.30926	0.271507	0.612085	0.806731	0.668938
houses_537	0.760586	0.522144	0.42876	0.580846	0.659909	0.84107	0.691953
kin8nm_189	0.353694	0.410896	0.190603	0.249942	0.191684	0.387209	0.37927
mtp_405	0.568483	0.27769	0.159129	0.168782	0.109426	0.657098	0.345234
mv_344	0.699932	0.640502	0.781384	0.547047	0.651239	0.901553	0.711976
no2_547	0.305845	0.297938	0.258686	0.173698	0.244973	0.31061	0.294099
pol_201	0.815246	0.611902	0.639754	0.702624	0.522728	0.878315	0.695217
pollen_529	0.697261	0.597242	0.347057	0.399166	0.416767	0.697328	0.484955
puma32H_308	0.665015	0.852149	0.425041	0.387022	0.120502	0.698762	0.715348
satellite_image_294	0.639354	0.35289	0.347273	0.230399	0.349314	0.731184	0.330443
socmob_541	0.73334	0.686206	0.664807	0.558244	0.170639	0.687682	0.565676
space_ga_507	0.514504	0.5696	0.228181	0.352075	0.584631	0.544893	0.386662
stock_223	0.604512	0.564316	0.496177	0.344409	0.623711	0.726325	0.448297
tecator_505	0.686373	0.265458	0.830009	0.079659	0.073141	0.781897	0.167463
us_crime_315	0.448203	0.245703	0.231679	0.034702	0.094120	0.541423	0.155004
vinnie_519	0.657658	0.581303	0.645009	0.294261	0.488865	0.717735	0.354748
visualizing_galaxy_690	0.615562	0.43737	0.562095	0.183542	0.106421	0.738236	0.297596
wind_503	0.564711	0.253707	0.335764	0.192989	0.301108	0.653679	0.339413
wine_quality_287	0.514151	0.48525	0.260365	0.345662	0.393642	0.581621	0.356523

Table 11: Average empirical stability measured using Ochiai metric

Dataset	MOSS-H	MOSS-M	MOSS-L	FIRE	GLRM	SIRUS	RuleFit
Ailerons_296	0.534274	0.224652	0.262974	0.229377	0.224151	0.732800	0.237581
ESL_1027	0.542947	0.424558	0.521958	0.232340	0.276984	0.659365	0.427253
Mercedes_Benz_42570	0.645853	0.525220	0.437106	0.326976	0.301945	0.697070	0.273472
Moneyball_41021	0.580138	0.350952	0.525092	0.167814	0.258333	0.673179	0.257835
abalone_183	0.438307	0.363968	0.345132	0.261420	0.357088	0.530265	0.190635
autoMpg_196	0.591775	0.265136	0.449240	0.221644	0.151592	0.638246	0.173451
auto_price_195	0.310888	0.167738	0.413344	0.027336	0.052137	0.286918	0.083685
bank32nh_558	0.618307	0.382116	0.127171	0.235653	0.624348	0.697249	0.419471
bodyfat_560	0.697452	0.336048	0.675731	0.195637	0.197125	0.750574	0.205783
cpu_small_227	0.651901	0.437399	0.375417	0.320944	0.489577	0.824803	0.599135
elevators_216	0.504497	0.443254	0.212563	0.321528	0.323980	0.645238	0.432407
house_16H_574	0.705332	0.338083	0.307953	0.267354	0.605561	0.806162	0.667867
houses_537	0.759752	0.521477	0.427733	0.579271	0.684691	0.840765	0.692597
kin8nm_189	0.354045	0.410996	0.182477	0.245684	0.196083	0.386821	0.377835
mtp_405	0.573716	0.281380	0.162165	0.167150	0.107181	0.663492	0.345849
mv_344	0.700041	0.643118	0.776825	0.541083	0.660033	0.901245	0.712186
no2_547	0.300969	0.296931	0.266697	0.184595	0.258225	0.306772	0.294892
pol_201	0.814457	0.611534	0.650009	0.736759	0.516296	0.878620	0.694048
pollen_529	0.691852	0.592593	0.353193	0.402698	0.419920	0.692169	0.481270
puma32H_308	0.687810	0.878869	0.416443	0.376662	0.123243	0.723240	0.738950
satellite_image_294	0.627831	0.346085	0.342532	0.225709	0.358154	0.717143	0.323611
socmob_541	0.742848	0.687619	0.680431	0.558221	0.173757	0.694449	0.569620
space_ga_507	0.507188	0.562345	0.245779	0.366538	0.619276	0.535657	0.382008
stock_223	0.617807	0.575305	0.483883	0.344834	0.641803	0.741020	0.457672
tecator_505	0.680476	0.262857	0.816622	0.079312	0.074212	0.774074	0.164815
us_crime_315	0.456817	0.251904	0.229608	0.034443	0.140149	0.551659	0.157081
vinnie_519	0.661026	0.584681	0.638624	0.288877	0.549118	0.721775	0.357468
visualizing_galaxy_690	0.610000	0.433968	0.570468	0.182116	0.106315	0.731005	0.294392
wind_503	0.580159	0.258862	0.342267	0.190028	0.297192	0.668519	0.348710
wine_quality_287	0.502963	0.476227	0.262332	0.347411	0.386793	0.566679	0.350028

Table 12: Average empirical stability measured using POG metric.

I Additional Discussions

In the following sections, we provide additional discussions on our MOSS framework.

I.1 ***Generating Candidate Rules*.** Throughout this paper, we use random forests to generate large collections of candidate rules, on which we apply MOSS. Random forests fit decision trees on bootstrapped samples of the original data, where only a subset of features are considered at each split in each tree. The randomness injected from the bootstrap and the feature sub-setting helps create a diverse set of candidate rules.

We note that alternative forms of randomness can be injected into a random forest. For example, in the original random forest paper [7], the author explores randomizing the outputs of each decision tree. Here, we explore adding additional forms of randomness when constructing candidate rules.

We use this new procedure. Given data X and response y we add Gaussian noise to the response to generate y'. We then fit a decision tree on (X, y') while randomizing the features considered per split. We repeat until we have a large collection of candidate rules.

Using this procedure, we repeat our experimental setup from §4.2 and apply MOSS to construct stable rule sets. We show the results in Table 13.

Method	MOSS	FIRE	GLRM	SIRUS	RuleFit
Avg. Accuracy Ranking	2.4	2.6	1.8	3.9	4.1
Avg. Stability Ranking	1.9	4.3	4.4	1.8	2.6
Combined Metric	2.2	3.5	3.1	2.9	3.5

Table 13: Experimental results: injecting additional forms of randomness when generating candidate rules.

These results are consistent with the main results of our paper, and we see here that again MOSS balance predictive accuracy with empirical stability. Only GLRM beats MOSS in terms of accuracy, however, MOSS is significantly more stable; GLRM is the least stable method. Only SIRUS beats MOSS in terms of stability, however, SIRUS is much less accurate. Exploring new methods to generate candidate rules may be an interesting direction for future research.

I.2 ***Classification Tasks*.** Our current discussion of MOSS focuses on constructing stable rule sets for regression tasks. However, we can extend MOSS to classification by replacing the ridge-regularized quadratic loss function in the accuracy objective $H_2(z)$ with a ridge-regularized logistic loss function. We could then apply a cutting-plane algorithm similar to the one described in [5].