Appendices for "An empirical study of data sampling techniques for just-in-time software defect prediction"

Zhiqiang Li[1*], Qiannan Du[1], Hongyu Zhang[2], Xiao-Yuan Jing[3,4], Fei Wu[5]

[1]School of Computer Science, Shaanxi Normal University, Xi'an, 710119, China.
[2]School of Information and Physical Sciences, The University of Newcastle, NSW, 2308, Australia.
[3]School of Computer Science, Wuhan University, Wuhan, 430072, China.
[4]School of Computer, Guangdong University of Petrochemical Technology, Maoming, 525000, China.
[5]College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, 210023, China.

*Corresponding author(s). E-mail(s): lizq@snnu.edu.cn;
Contributing authors: duqiannan99@outlook.com; hongyu.zhang@newcastle.edu.au; jingxy_2000@126.com; wufei_8888@126.com;

1 Comparison results of LApredict with random forest classifier

Previous defect prediction work (Lessmann et al, 2008; Ghotra et al, 2015; Tantithamthavorn et al, 2019) point out random forest (RF) tends to be the top-performing classification techniques, tends to produce stable performance estimates, and is insensitive to parameter settings. Hence, we consider *LApredict* using the RF classifier to build JIT-SDP models.

Figure 1 shows the absolute performance of *LApredict* with RF when applying data sampling techniques to defect prediction models for each of the 6 non-effort-aware measures in the context of defect classification. In this figure, a red dotted line indicates

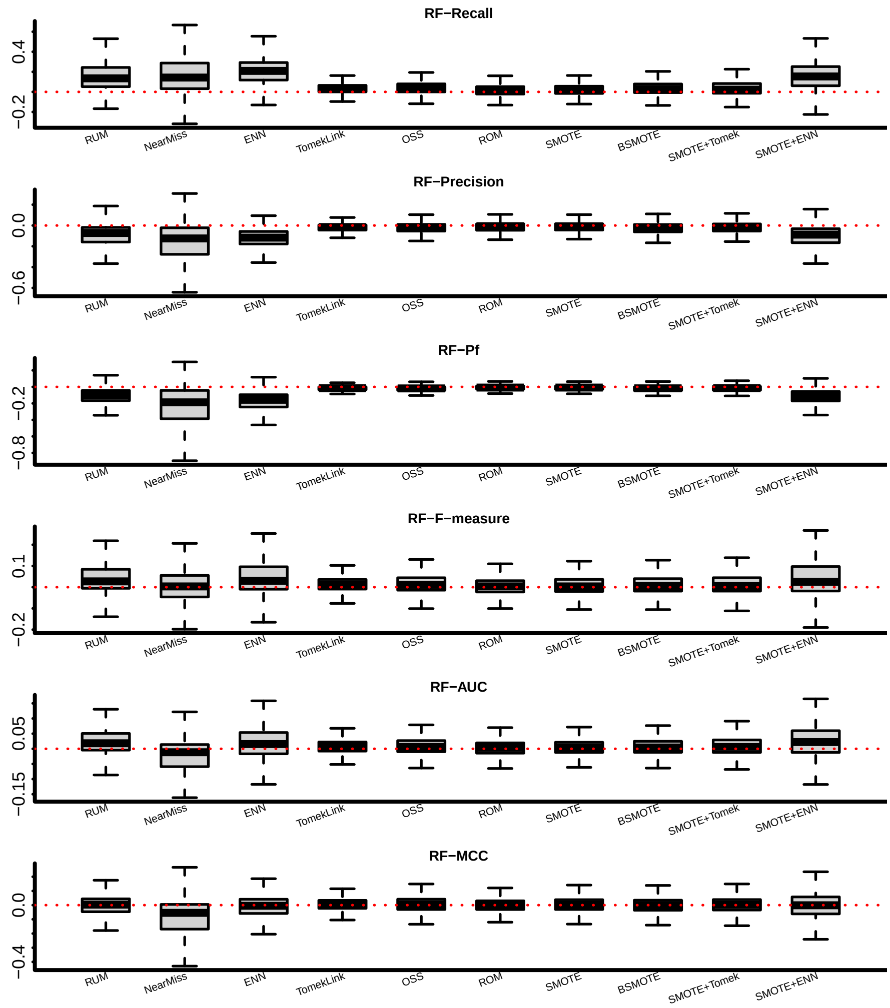

Fig. 1 The absolute performance difference of all sampling algorithms for each of the 6 non-effort-aware performance measures across 10 projects

a performance difference of zero (i.e., no improvement). Figure 2 shows the results of Scott-Knott ESD test concerning the 6 non-effort-aware performance measures. In this test, different colors denote different groups with statistical significance. The lower the value, the better the model performance. As shown in these figures, we have the following observations:

① In terms of *Recall*, almost all the data sampling algorithms exhibit better than NONE with statistical significance, expect for the ROM algorithm, which has a performance similar to NONE. Especially, ENN achieves the best performance since it lies in the first group. Following it are NearMiss and RUM.

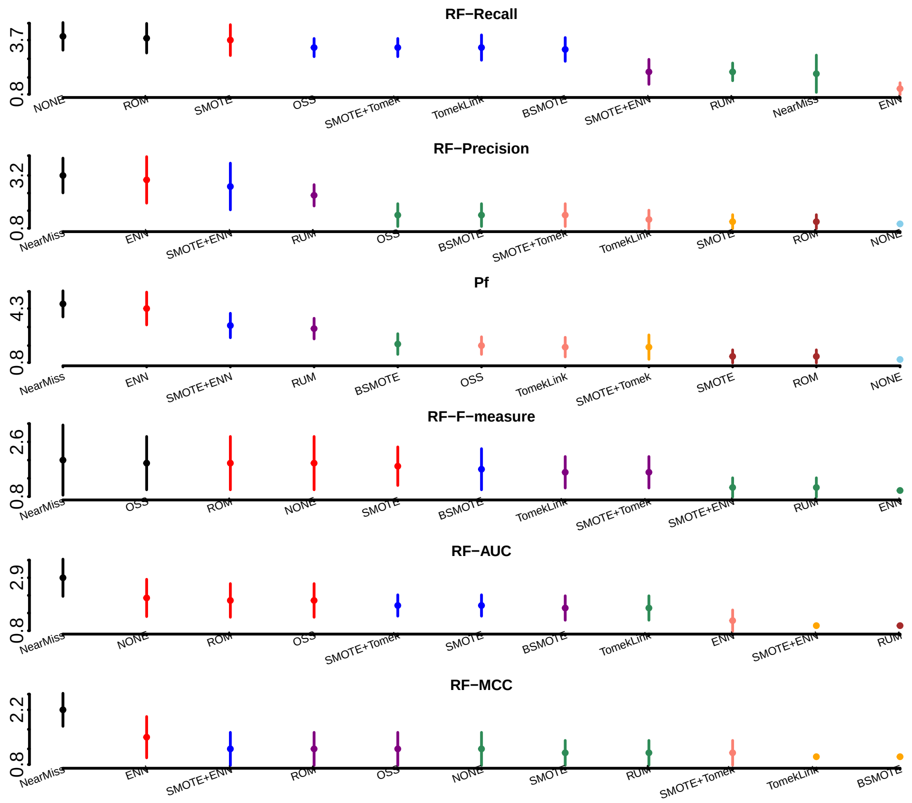

Fig. 2 The results of Scott-Knott ESD test of all sampling algorithms for each of the 6 non-effort-aware performance measures across 10 projects (the lower the better, the same below)

② With regard to *Precision* and *Pf*, NONE performs the best while NearMiss performs the worst among the data sampling algorithms, indicating that data sampling algorithms do not improve the performance in these two measures.

③ Regarding *F-measure*, most of the data sampling algorithms achieve better results than NONE with statistical significance. Specifically, ENN, RUM and SMOTE+ENN outperform other algorithms in terms of *F-measure*.

④ Regarding *AUC*, most of the data sampling algorithms show improvements. Among all the algorithms, RUM obtains the best performance in *AUC*.

⑤ In terms of *MCC*, only several data sampling algorithms show improvements. Among them, BSMOTE and TomekLink obtain the best performance, followed by SMOTE+Tomek.

In summary, the data sampling techniques usually can produce good results in terms of *Recall*, *F-measure*, *AUC*, and *MCC*, while exhibiting poor performance in *Precision* and *Pf* in the scenario of defect classification for JIT-SDP. Among all the data sampling algorithms, the RUM and ENN emerge as the best-performing algorithms overall, particularly excelling in *Recall*, *F-measure*, and *AUC*.

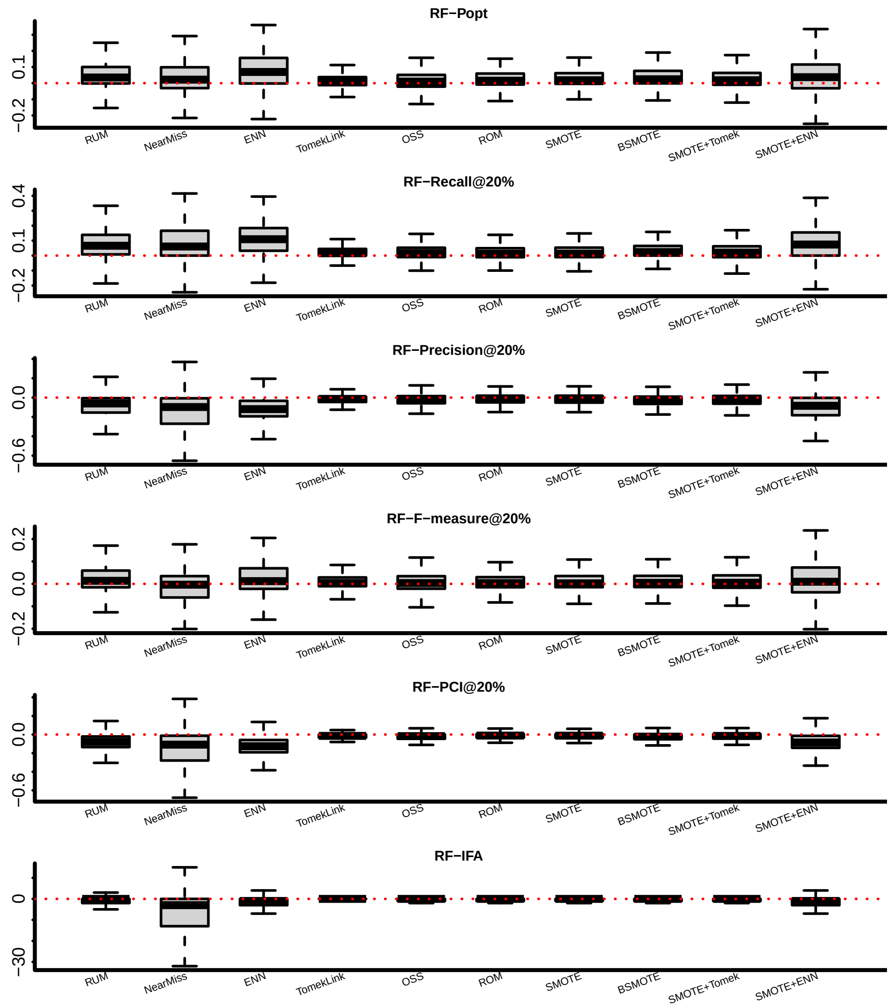

Fig. 3 The absolute performance difference of all sampling algorithms for each of the 6 effort-aware performance measures across 10 projects

> *The data sampling techniques demonstrate varying performance across different evaluation measures in the context of defect classification for JIT-SDP. Among them, the RUM and ENN algorithms emerge as the most favorable option for achieving superior results overall, especially in Recall, F-measure, and AUC.*

Figure 3 shows the absolute performance of *LApredict* with RF when applying data sampling techniques to defect prediction models for each of the 6 effort-aware measures. In this figure, a red dotted line indicates a performance difference of zero (i.e., no improvement). Figure 4 shows the results of Scott-Knott ESD test concerning the 6 effort-aware performance measures across 10 projects in the scenario of defect ranking. From these figures, we make the the following observations:

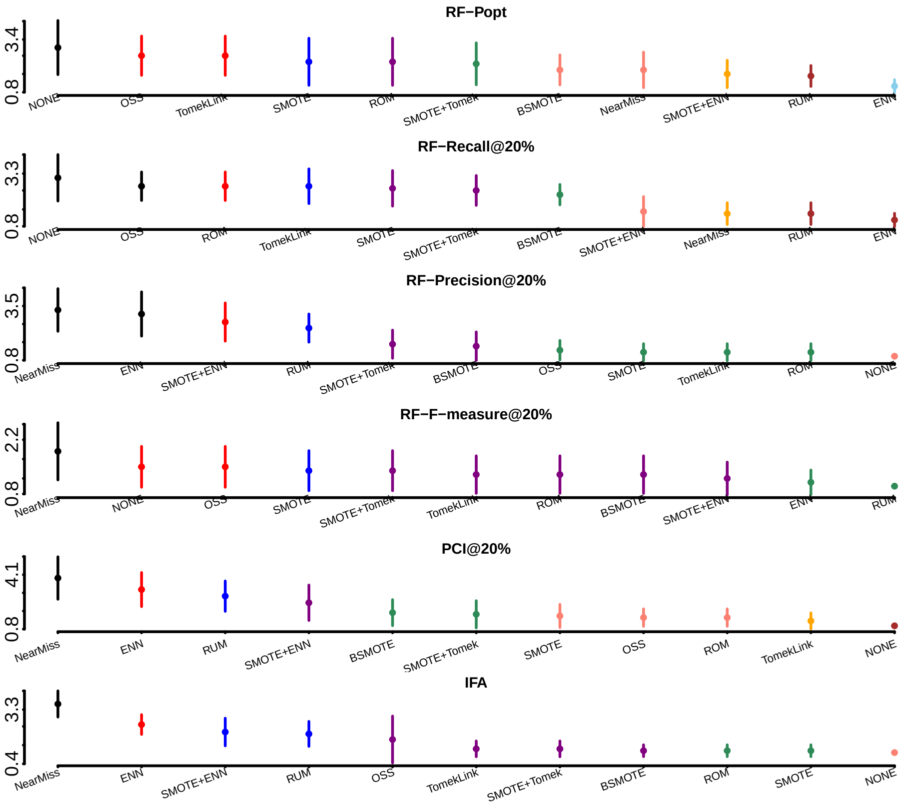

Fig. 4 The results of Scott-Knott ESD test of all sampling algorithms for each of the 6 effort-aware performance measures across 10 projects

① In terms of P_{opt}, all the data sampling algorithms demonstrate better performance than NONE with statistical significance which is in the last group. Especially, ENN achieves the best performance, followed by RUM. The *Recall@20%* measure shows similar results to those of P_{opt}.

② With respect to *Precision@20%*, *PCI@20%*, and *IFA* NONE exhibits the best performance with statistical significance among all the algorithms, indicting that data sampling algorithms have negative impact on these performance measures. This implies that the initial false alarms may negatively impact practitioners' patience and confidence in practice.

③ Regarding *F-measure@20%*, most of the data sampling algorithms show better performance than NONE with statistical significance. The RUM and ENN algorithms achieve the best performance among all the methods since they lie in the first group.

In summary, the data sampling techniques utilized in the defect ranking for JIT-SDP demonstrate favorable results in terms of P_{opt}, *Recall@20%*, and *F-measure@20%*, especially for ENN and RUM. These measures indicate that the data sampling algorithms are effective in improving the ranking performance and identifying potential defects in JIT-SDP. However, the data sampling techniques exhibit relatively poorer performance in *Precision@20%*, *PCI@20%*, and *IFA*. These measures focus on the precision of defect ranking, and the data sampling algorithms may not consistently achieve high precision or minimize the number of incorrectly ranked changes.

The effectiveness of data sampling techniques can vary depending on the specific evaluation measures in the context of defect ranking for JIT-SDP. Among them, the ENN and RUM algorithms stand out as the most favorable option for achieving superior results overall, especially in P_{opt}, Recall@20%, and F-measure@20%.

2 Comparison results of LApredict for each data sampling algorithm

Tables 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13 present detailed median values in terms of *Recall, Precision, Pf, F-measure, AUC, MCC*, P_{opt}, *Recall@20%*, *Precision@20%*, *F-measure@20%*, *PCI@20%*, and *IFA* of each sampling algorithm for each project. The overall median values across all projects are also provided. Additionally, the table highlights the best algorithm for each project in bold font.

References

Ghotra B, McIntosh S, Hassan AE (2015) Revisiting the impact of classification techniques on the performance of defect prediction models. In: Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (ICSE). IEEE, pp 789–800

Lessmann S, Baesens B, Mues C, et al (2008) Benchmarking classification models for software defect prediction: A proposed framework and novel findings. IEEE Transactions on Software Engineering 34(4):485–496

Tantithamthavorn C, McIntosh S, Hassan AE, et al (2019) The impact of automated parameter optimization on defect prediction models. IEEE Transactions on Software Engineering 45(7):683–711

Table 1 Median results in terms of *Recall* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	0.384	0.687	0.677	0.688	0.441	0.426	0.696	0.697	**0.721**	0.714	0.685
JGroups	0.331	0.648	0.667	0.571	0.350	0.358	0.652	0.648	**0.673**	0.652	0.577
Camel	0.442	0.733	0.691	**0.782**	0.508	0.511	0.733	0.731	0.754	0.749	0.731
Tomcat	0.529	0.672	0.677	**0.834**	0.561	0.563	0.676	0.676	0.711	0.689	0.650
Brackets	0.673	0.775	0.785	**0.879**	0.689	0.691	0.783	0.776	0.818	0.783	0.765
Neutron	0.747	0.844	0.802	**0.907**	0.764	0.819	0.844	0.844	0.883	0.852	0.849
Spring	0.599	0.774	0.677	**0.878**	0.647	0.667	0.757	0.758	0.811	0.797	0.811
Broadleaf	0.469	0.750	0.668	0.760	0.512	0.512	0.747	0.750	**0.786**	0.763	0.722
Nova	0.719	0.862	0.781	**0.919**	0.733	0.766	0.860	0.860	0.894	0.862	0.867
Npm	0.288	0.667	0.667	0.667	0.333	0.358	0.667	0.662	0.686	**0.692**	0.677
Mean	0.518	0.741	0.709	**0.788**	0.554	0.567	0.741	0.740	0.774	0.755	0.733

Table 2 Median results in terms of *Precision* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	**0.610**	0.510	0.518	0.522	0.586	0.586	0.513	0.510	0.503	0.509	0.522
JGroups	**0.643**	0.457	0.452	0.533	0.625	0.636	0.456	0.456	0.438	0.455	0.487
Camel	**0.600**	0.507	0.507	0.494	0.593	0.590	0.508	0.504	0.497	0.496	0.508
Tomcat	**0.709**	0.641	0.638	0.564	0.689	0.690	0.641	0.641	0.624	0.638	0.665
Brackets	**0.732**	0.677	0.675	0.634	0.728	0.717	0.680	0.680	0.655	0.676	0.694
Neutron	0.759	0.731	0.753	0.689	**0.763**	0.741	0.731	0.731	0.717	0.722	0.731
Spring	**0.729**	0.681	0.700	0.642	0.714	0.707	0.677	0.682	0.667	0.679	0.667
Broadleaf	**0.661**	0.538	0.540	0.521	0.635	0.631	0.537	0.533	0.524	0.525	0.535
Nova	**0.747**	0.704	0.730	0.685	0.739	0.714	0.704	0.704	0.692	0.702	0.704
Npm	0.581	0.481	0.500	0.476	**0.586**	0.571	0.494	0.486	0.482	0.482	0.493
Mean	**0.677**	0.593	0.601	0.576	0.666	0.658	0.594	0.593	0.580	0.588	0.600

Table 3 Median results in terms of *Pf* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	**0.084**	0.303	0.295	0.316	0.116	0.117	0.300	0.306	0.330	0.325	0.272
JGroups	**0.061**	0.316	0.334	0.212	0.078	0.08	0.316	0.307	0.334	0.316	0.248
Camel	**0.120**	0.306	0.290	0.332	0.137	0.137	0.306	0.306	0.325	0.318	0.296
Tomcat	**0.172**	0.263	0.269	0.465	0.186	0.186	0.263	0.263	0.295	0.272	0.243
Brackets	**0.138**	0.206	0.213	0.306	0.144	0.144	0.205	0.206	0.244	0.214	0.188
Neutron	**0.122**	0.171	0.147	0.222	0.135	0.154	0.178	0.179	0.197	0.179	0.180
Spring	**0.165**	0.279	0.228	0.349	0.193	0.220	0.272	0.284	0.292	0.283	0.298
Broadleaf	**0.088**	0.249	0.194	0.259	0.103	0.098	0.246	0.252	0.287	0.258	0.218
Nova	**0.131**	0.184	0.156	0.223	0.139	0.151	0.187	0.184	0.211	0.191	0.197
Npm	**0.087**	0.313	0.313	0.342	0.097	0.119	0.319	0.319	0.333	0.339	0.316
Mean	**0.117**	0.259	0.244	0.303	0.133	0.141	0.259	0.261	0.285	0.270	0.246

Table 4 Median results in terms of *F-measure* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	0.450	0.564	0.556	0.557	0.483	0.479	0.564	**0.568**	**0.568**	0.567	0.563
JGroups	0.407	0.532	0.532	0.492	0.409	0.411	**0.533**	0.527	0.530	0.524	0.521
Camel	0.514	0.599	0.582	0.600	0.533	0.537	**0.601**	0.598	0.593	0.594	0.597
Tomcat	0.610	0.653	0.648	0.657	0.617	0.618	0.648	0.648	**0.658**	0.653	0.644
Brackets	0.691	0.724	0.714	0.712	0.693	0.697	0.723	**0.725**	0.719	**0.725**	0.717
Neutron	0.748	0.781	0.774	**0.793**	0.753	0.759	0.778	0.778	0.780	0.783	0.781
Spring	0.654	0.704	0.678	**0.713**	0.664	0.678	0.705	0.702	0.705	0.704	0.699
Broadleaf	0.527	0.600	0.598	0.592	0.549	0.553	**0.608**	**0.608**	0.604	**0.608**	0.605
Nova	0.734	0.760	0.760	**0.768**	0.741	0.745	0.760	0.760	0.759	0.760	0.752
Npm	0.349	0.538	**0.553**	0.527	0.411	0.411	0.545	0.544	0.547	0.545	0.524
Mean	0.568	0.646	0.640	0.641	0.585	0.589	**0.647**	0.646	0.646	0.646	0.640

Table 5 Median results in terms of AUC for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	0.632	0.702	0.704	0.692	0.651	0.652	0.706	0.706	0.704	0.706	**0.707**
JGroups	0.614	0.665	0.664	0.648	0.623	0.623	0.665	0.666	**0.667**	**0.667**	0.661
Camel	0.656	**0.722**	0.701	0.713	0.666	0.666	**0.722**	0.721	0.718	0.719	0.714
Tomcat	0.687	**0.701**	**0.701**	0.666	0.692	0.692	0.700	0.700	0.700	0.698	0.698
Brackets	0.771	0.789	**0.790**	0.785	0.771	0.773	0.789	0.789	0.788	0.789	0.780
Neutron	0.809	0.833	0.823	**0.840**	0.816	0.825	0.831	0.833	0.839	0.835	0.834
Spring	0.704	0.727	0.714	0.717	0.705	0.707	0.729	0.728	**0.731**	0.730	0.728
Broadleaf	0.683	0.750	0.733	0.735	0.696	0.696	0.751	**0.754**	0.747	0.751	0.748
Nova	0.796	0.834	0.818	**0.839**	0.806	0.805	0.834	0.833	0.835	0.834	0.834
Npm	0.598	0.680	0.673	0.647	0.623	0.623	0.689	0.685	0.680	**0.694**	0.669
Mean	0.695	0.740	0.732	0.728	0.705	0.706	0.741	**0.742**	0.741	**0.742**	0.737

Table 6 Median results in terms of MCC for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	0.310	0.371	0.363	0.376	0.322	0.318	0.364	0.360	0.369	0.373	**0.380**
JGroups	0.296	0.298	0.301	0.285	0.296	0.298	0.306	0.301	0.302	0.297	**0.314**
Camel	0.342	0.397	0.385	0.393	0.354	0.352	0.400	**0.401**	**0.401**	0.394	0.392
Tomcat	0.394	0.394	0.396	0.339	0.401	0.402	0.394	0.393	0.395	0.394	**0.403**
Brackets	0.544	0.558	**0.563**	0.545	0.552	0.554	**0.563**	0.561	0.557	0.56	0.554
Neutron	0.620	0.640	0.627	**0.664**	0.627	0.642	0.638	0.643	0.645	0.643	0.644
Spring	0.414	0.442	0.408	0.439	0.429	0.429	0.439	0.437	**0.453**	0.445	0.443
Broadleaf	0.403	**0.457**	0.430	0.416	0.414	0.417	0.455	**0.457**	0.448	0.456	0.452
Nova	0.590	0.622	0.598	0.622	0.592	0.579	0.613	0.613	**0.631**	0.623	0.623
Npm	0.250	0.301	0.309	0.287	0.274	0.267	**0.313**	0.308	0.299	0.308	0.288
Mean	0.416	0.448	0.438	0.436	0.426	0.426	0.448	0.447	**0.450**	0.449	0.449

Table 7 Median results in terms of P_{opt} for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	0.490	0.725	0.709	0.736	0.541	0.554	0.721	0.721	**0.748**	0.744	0.736
JGroups	0.430	0.605	0.629	0.579	0.436	0.448	0.611	0.605	**0.632**	0.603	0.543
Camel	0.489	0.716	0.680	**0.773**	0.528	0.523	0.712	0.711	0.741	0.730	0.711
Tomcat	0.529	0.646	0.655	**0.804**	0.546	0.548	0.647	0.658	0.684	0.666	0.636
Brackets	0.639	0.761	0.772	**0.858**	0.655	0.663	0.766	0.761	0.808	0.771	0.749
Neutron	0.740	0.832	0.809	**0.902**	0.763	0.821	0.832	0.832	0.876	0.846	0.849
Spring	0.614	0.748	0.684	**0.851**	0.643	0.656	0.748	0.747	0.754	0.754	0.795
Broadleaf	0.488	0.748	0.670	0.752	0.521	0.513	0.740	0.748	**0.756**	0.751	0.716
Nova	0.719	0.850	0.762	**0.917**	0.732	0.770	0.850	0.850	0.888	0.857	0.856
Npm	0.434	0.652	0.655	0.681	0.461	0.463	0.653	0.649	0.667	0.670	**0.682**
Mean	0.557	0.728	0.703	**0.785**	0.583	0.596	0.728	0.728	0.755	0.739	0.727

Table 8 Median results in terms of *Recall@20%* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	0.390	0.696	0.678	0.699	0.453	0.435	0.696	0.706	**0.730**	0.719	0.699
JGroups	0.303	0.548	0.562	0.492	0.319	0.327	0.556	0.550	**0.571**	0.549	0.475
Camel	0.390	0.680	0.647	**0.711**	0.452	0.452	0.678	0.678	0.690	0.685	0.677
Tomcat	0.477	0.591	0.592	**0.744**	0.494	0.494	0.597	0.592	0.624	0.613	0.577
Brackets	0.606	0.724	0.749	**0.813**	0.627	0.630	0.725	0.728	0.764	0.733	0.707
Neutron	0.733	0.816	0.790	**0.884**	0.744	0.787	0.814	0.814	0.853	0.827	0.822
Spring	0.556	0.692	0.619	**0.762**	0.592	0.600	0.692	0.692	0.700	0.700	0.721
Broadleaf	0.437	0.704	0.641	0.703	0.470	0.470	0.701	0.705	**0.735**	0.710	0.671
Nova	0.693	0.826	0.747	**0.882**	0.706	0.756	0.827	0.826	0.851	0.832	0.824
Npm	0.294	0.615	0.618	0.622	0.348	0.348	0.612	0.615	0.640	**0.654**	0.642
Mean	0.488	0.689	0.664	**0.731**	0.520	0.530	0.690	0.691	0.716	0.702	0.682

Table 9 Median results in terms of *Precision@20%* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	**0.603**	0.479	0.495	0.488	0.585	0.585	0.484	0.484	0.465	0.476	0.493
JGroups	**0.571**	0.409	0.407	0.462	0.561	0.559	0.417	0.417	0.397	0.409	0.440
Camel	**0.583**	0.489	0.484	0.468	0.576	0.575	0.487	0.487	0.479	0.484	0.497
Tomcat	**0.684**	0.613	0.616	0.540	0.667	0.671	0.608	0.607	0.594	0.606	0.642
Brackets	**0.712**	0.643	0.643	0.588	0.697	0.696	0.653	0.652	0.628	0.646	0.662
Neutron	**0.766**	0.724	0.75	0.681	0.760	0.740	0.724	0.730	0.695	0.724	0.718
Spring	0.712	0.656	0.683	0.628	**0.718**	0.706	0.654	0.659	0.633	0.654	0.656
Broadleaf	0.631	0.525	0.529	0.508	**0.635**	**0.635**	0.525	0.519	0.506	0.511	0.523
Nova	**0.714**	0.694	0.709	0.641	**0.714**	0.707	0.695	0.699	0.676	0.695	0.676
Npm	0.569	0.448	0.484	0.467	**0.584**	0.557	0.470	0.470	0.456	0.442	0.474
Mean	**0.655**	0.568	0.580	0.547	0.650	0.643	0.572	0.572	0.553	0.565	0.578

Table 10 Median results in terms of *F-measure@20%* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	0.453	0.558	0.556	0.553	0.485	0.479	0.558	0.562	0.561	**0.564**	0.558
JGroups	0.356	**0.471**	**0.471**	0.430	0.353	0.353	**0.471**	0.468	0.466	0.470	**0.471**
Camel	0.492	0.562	0.548	**0.563**	0.499	0.501	0.562	0.562	0.555	0.559	0.558
Tomcat	0.548	0.598	0.600	**0.614**	0.556	0.561	0.598	0.606	0.602	0.601	0.592
Brackets	0.600	0.651	0.646	0.671	0.606	0.610	0.667	0.660	0.662	0.664	**0.674**
Neutron	0.744	0.778	0.768	**0.781**	0.748	0.762	0.778	0.778	0.778	0.780	0.778
Spring	0.611	0.661	0.611	**0.676**	0.616	0.616	0.659	0.659	**0.676**	0.663	0.671
Broadleaf	0.500	0.572	0.561	0.565	0.529	0.540	**0.583**	0.577	0.570	0.576	0.576
Nova	0.699	**0.734**	0.723	0.730	0.699	0.700	**0.734**	**0.734**	0.732	**0.734**	0.725
Npm	0.340	0.511	**0.519**	0.481	0.381	0.366	0.511	0.511	0.512	0.512	0.472
Mean	0.534	0.61	0.600	0.606	0.547	0.549	**0.612**	**0.612**	0.611	**0.612**	0.608

Table 11 Median results in terms of *PCI@20%* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	**0.182**	0.394	0.401	0.404	0.211	0.215	0.408	0.405	0.441	0.425	0.397
JGroups	**0.14**	0.364	0.384	0.313	0.158	0.158	0.370	0.370	0.392	0.373	0.315
Camel	**0.209**	0.413	0.403	0.459	0.239	0.239	0.412	0.411	0.429	0.421	0.402
Tomcat	**0.290**	0.395	0.408	0.561	0.302	0.305	0.396	0.396	0.429	0.409	0.370
Brackets	**0.293**	0.385	0.392	0.477	0.307	0.313	0.383	0.384	0.420	0.389	0.368
Neutron	**0.322**	0.396	0.364	0.449	0.339	0.366	0.394	0.394	0.422	0.399	0.399
Spring	**0.315**	0.409	0.386	0.505	0.354	0.373	0.408	0.404	0.429	0.430	0.438
Broadleaf	**0.170**	0.358	0.329	0.372	0.204	0.204	0.359	0.362	0.394	0.378	0.349
Nova	**0.366**	0.406	0.380	0.463	0.367	0.387	0.406	0.406	0.431	0.407	0.408
Npm	**0.163**	0.392	0.404	0.459	0.189	0.201	0.392	0.408	0.404	0.432	0.392
Mean	**0.245**	0.391	0.385	0.446	0.267	0.276	0.393	0.394	0.419	0.406	0.384

Table 12 Median results in terms of *IFA* for each project

Project	NONE	RUM	NearMiss	ENN	TomekLink	OSS	ROM	SMOTE	BSMOTE	SMOTE+Tomek	SMOTE+ENN
Fabric8	**2**	4	3.5	4	**2**	**2**	4	4	4	4.5	5
JGroups	**2**	4	5	3	**2**	**2**	4	5	5	5	4
Camel	**2**	3	3	3	**2**	**2**	3	3	3	3	3
Tomcat	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Brackets	**2**	3	3	4	**2**	**2**	3	3	3	3	**2**
Neutron	**1**	2	2	2	**1**	2	2	2	2	2	2
Spring	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Broadleaf	**2**	4	3	3	**2**	**2**	4	4	4	4	3
Nova	**2**	**2**	**2**	3	**2**	**2**	**2**	**2**	3	**2**	3
Npm	**2**	3	3	3	**2**	**2**	3	4	4	3	3
Mean	**1.8**	2.8	2.75	2.8	**1.8**	1.9	2.8	3	3.1	2.95	2.8