Online Appendix

This appendix contained the following cases for more discussion about this paper:

1. More information about the dataset time span, the subsystem, the ARB type of each subsystem, the number of ARBs in each subsystem, and the total ARB files in each dataset.

Table 1 Dataset information description

Project	Time Span	Subsystem	ARB type	# ARBs	ARB file
Linux	Dec 2003 -May 2011, version 2.6	Network Drivers	Memory-related	7	20
			Numerical	1	
			Other logical resource	1	
		SCSI Drivers	Memory-related	4	
		EXT3 FileSystem	Memory-related	3	
			Storage-related	2	
		Networking/IPV4	Memory-related	1	
			Other logical resource	1	
MySQL	Aug 2006 – Feb 2011, version 5.1	InnoDB Storage engine	Memory-related	2	39
			Numerical	1	
			Other logical resource	3	
		Replication	Memory-related	4	
			Other logical resource	1	
		Optimizer	Memory-related	5	
NetBSD	Otc 2015, version 7.x (7.0, 7.0.1, 7.0.2, 7.1, 7.1.1, 7.1.2, 7.2)	kern	Memory-related	3	21
			Other logical resource	1	
		fs	Memory-related	0	
			Other logical resource	1	
		usr.bin	Memory-related	3	
			Other logical resource	2	
		usr.sbin	Memory-related	6	
			Other logical resource	2	
		crypto	Memory-related	3	
			Other logical resource	0	

2. The detailed information about the 52-dimensional feature set used in this paper.

Table 2 Software complex metrics description

Type	Metrics	Description
Program size	AltAvgLineBlank, AltAvgLineCode, AltAvgLineComment, AltCountLineBlank, AltCountLineCode, AltCountLineComment, AvgLine, AvgLineBlank, AvgLineCode, AvgLineComment, CountDeclClass, CountDeclFunction, CountLine, CountLineBlank, CountLineCode, CountLineCodeDecl, CountLineCodeExe, CountLineComment, CountLineInactive, CountLinePreprocessor, CountSemicolon, CountStmt, CountStmtDecl, CountStmtEmpty, CountStmtExe, RatioCommentToCode	Metrics related to the amount of lines of code, declarations, statements, and files
McCabe's cyclomatic complexity	AvgCyclomatic, AvgCyclomaticModified, AvgCyclomaticStrict, AvgEssential, MaxCyclomatic, MaxCyclomaticModified, MaxCyclomaticStrict, SumCyclomatic, SumCyclomaticModified, SumCyclomaticStrict, SumEssential	Metrics related to the control flow graph of functions and methods
Halstead metrics	Program Volume, Program Length, Program Vocabulary, Program Difficulty, Effort, N1, N2, n1, n2	Metrics based on operands and operators in the program
Aging-Related Metrics (ARMs)	AllocOps, DeallocOps, DerefSet, DerefUse, UniqueDerefSet, UniqueDerefUse	Metrics related to memory usage

3. The package versions required for running the experiments, as well as the training parameters for each classifier

Table 3 Parameter values set for different machine learning classifiers

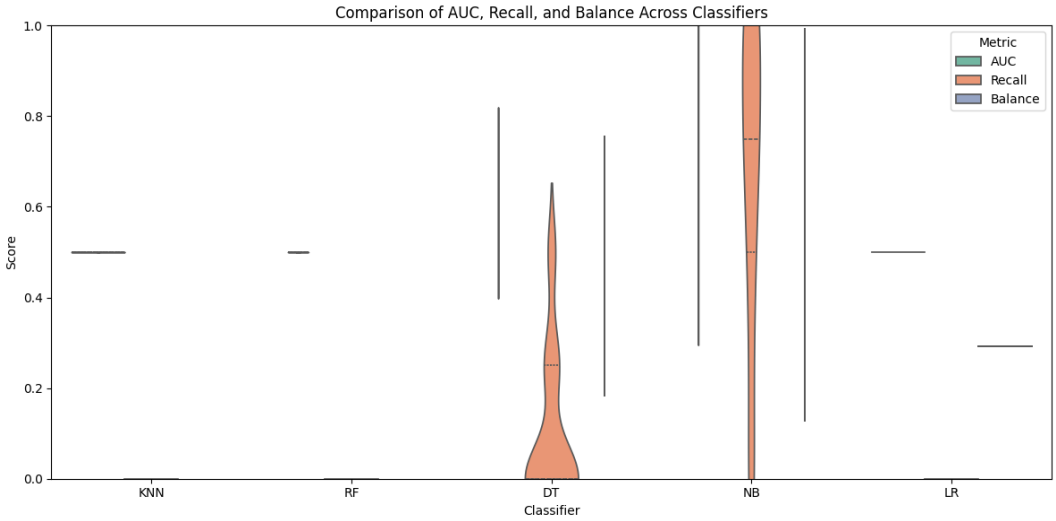
Technique	Parameter values
K-Nearest Neighbour (KNN)	n_neighbors: 5, weights: 'uniform', algorithm: 'auto', leaf_size: 30, metric: 'minkowski'
Naïve Bayes (NB)	With default param eter in scikitlearn
Logistic Regression (LR)	Solvers = liblinear, penalty = l2, c_values = 1.0, class_weight: None
Decision Tree (DT)	mdepth=np.arange(1,40), grid={"criterion":["gini","entropy"],"max_depth":mdepth} max_depth: 5, max_features: None, min_samples_split: 2, min_samples_leaf: 1, random_state: None
Random Forest (RF)	max_depth: 5, n_estimators: 10, max_leaf_nodes: None, random_state: None, min_samples_split: 2, min_samples_leaf: 1

Table 4 The package versions used in the experiment.

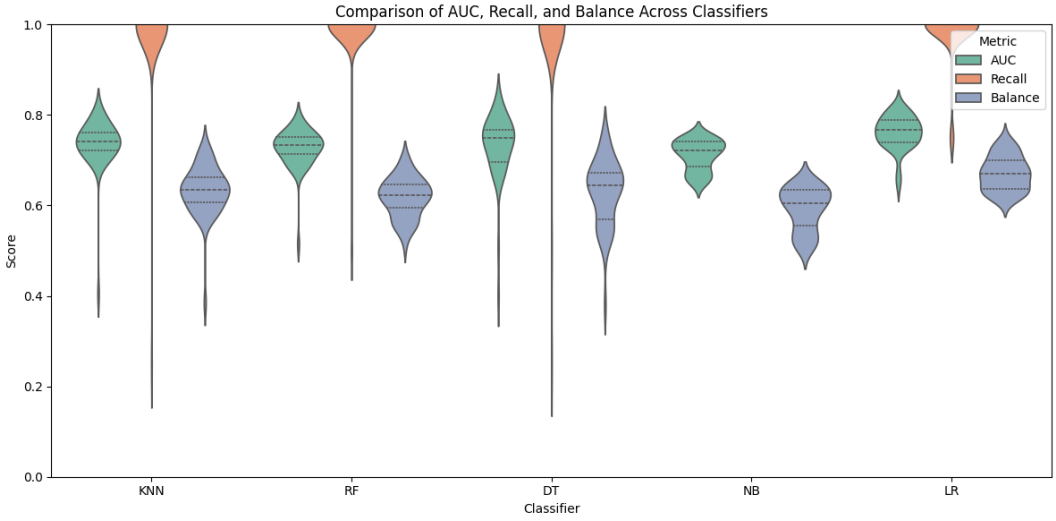
Parameter	Version
ctgan	0.8.0
imbalanced-learn	0.8.1
scikit-learn	0.24.0
setuptools	68.2.2

tensorflow	2.13.1
pandas	1.5.2

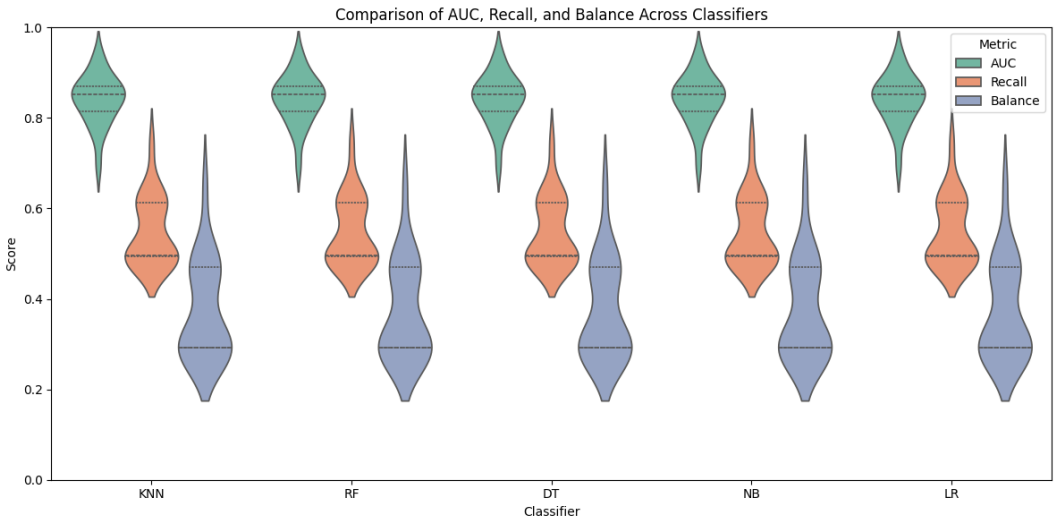
4. The violin plots of Table III, IV, and V in this paper



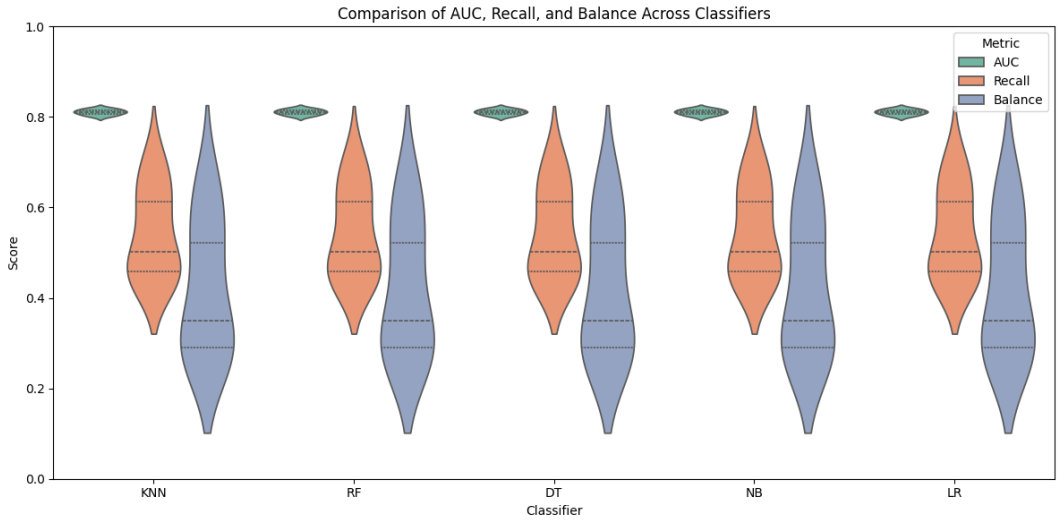
(a) Original result



(b) Intra-class balancing result

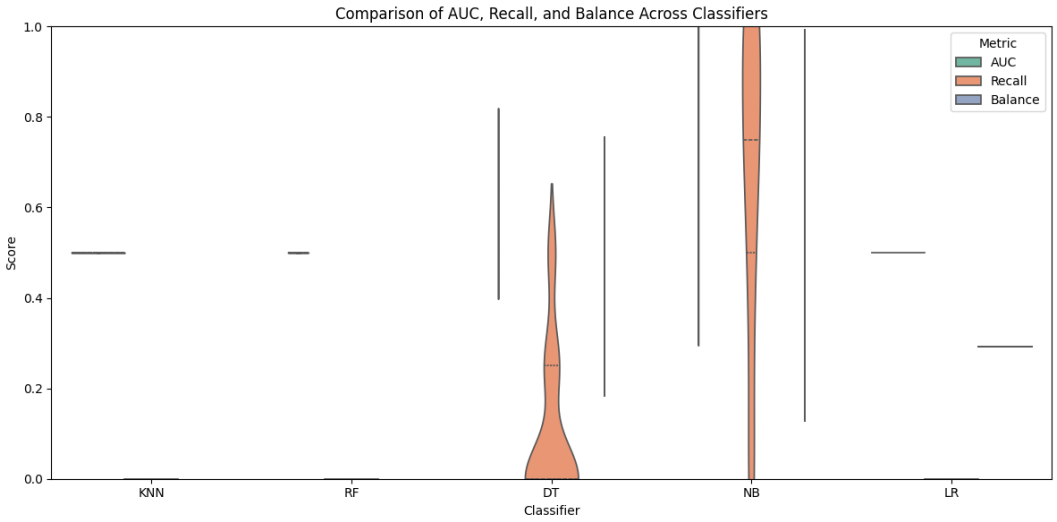


(c) Inter-class balancing result

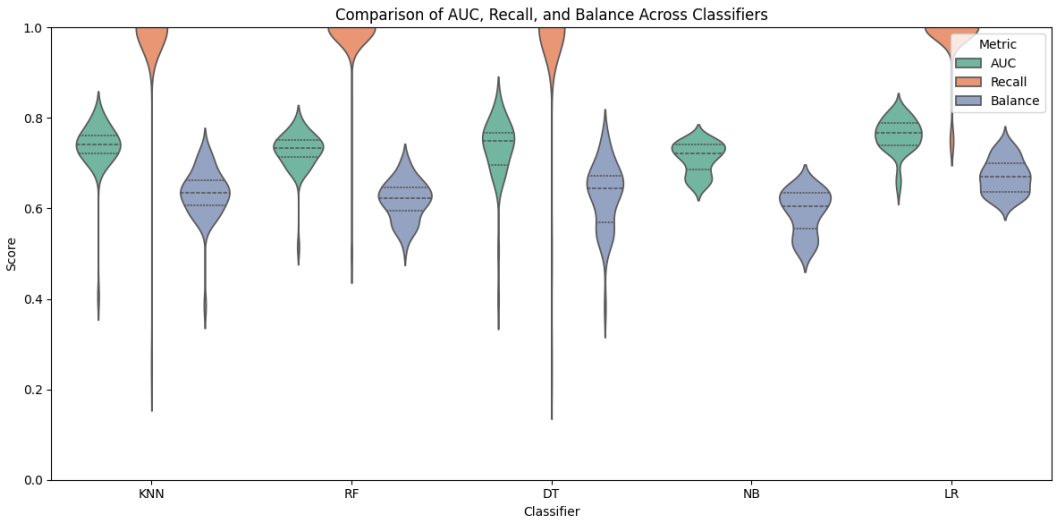


(d) I2SG result

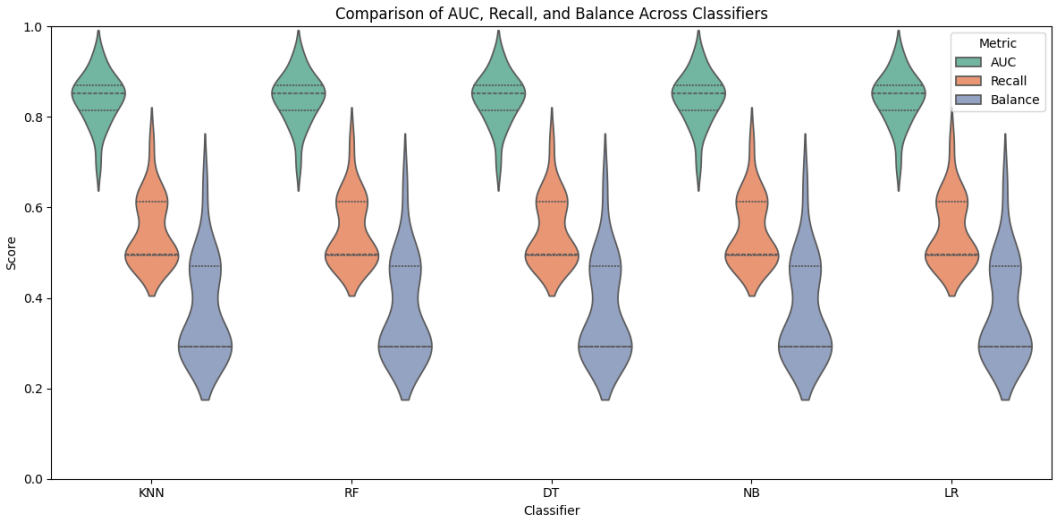
Figure 1 The violin plot of Table III on Linux dataset



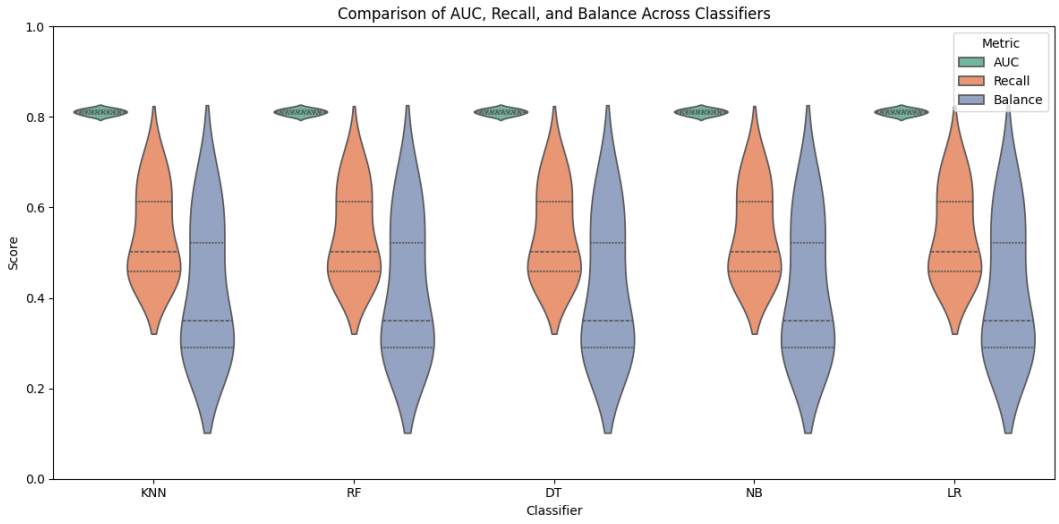
(a) Original result



(b) Intra-class balancing result

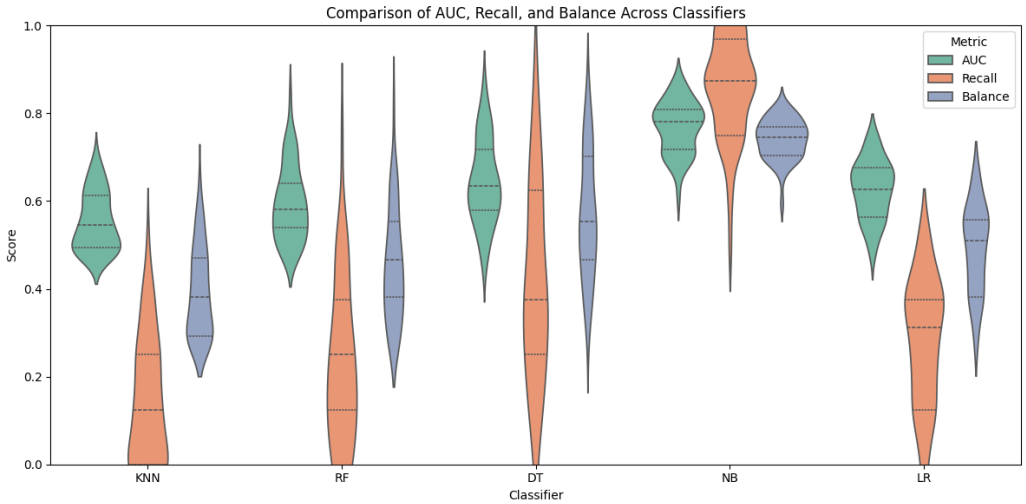


(c) Inter-class balancing result

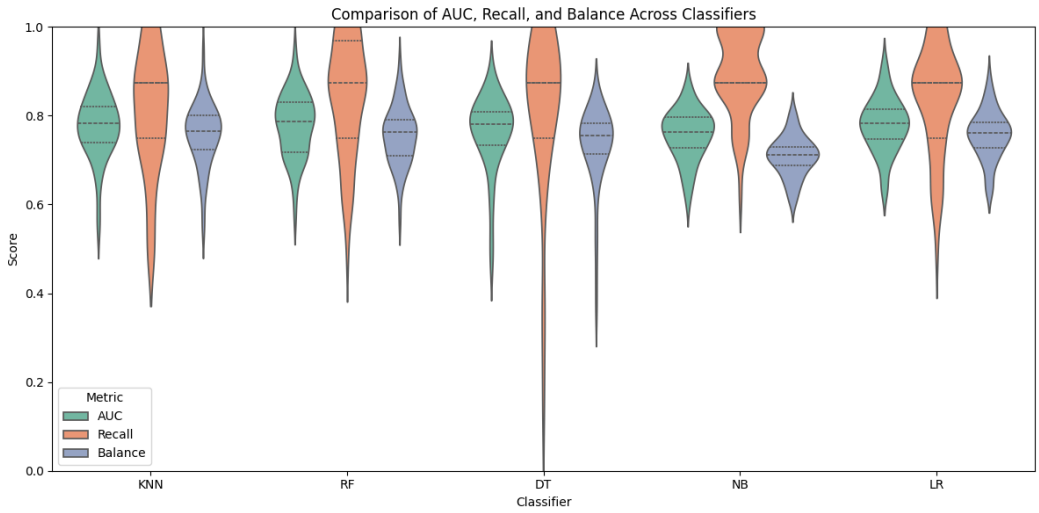


(d) I2SG result

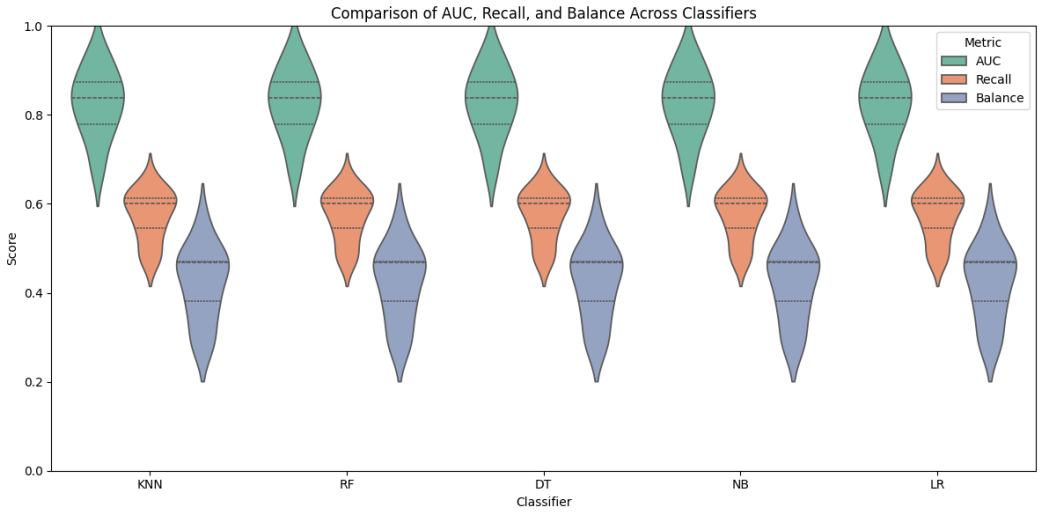
Figure 2 The violin plot of Table III on Linux dataset



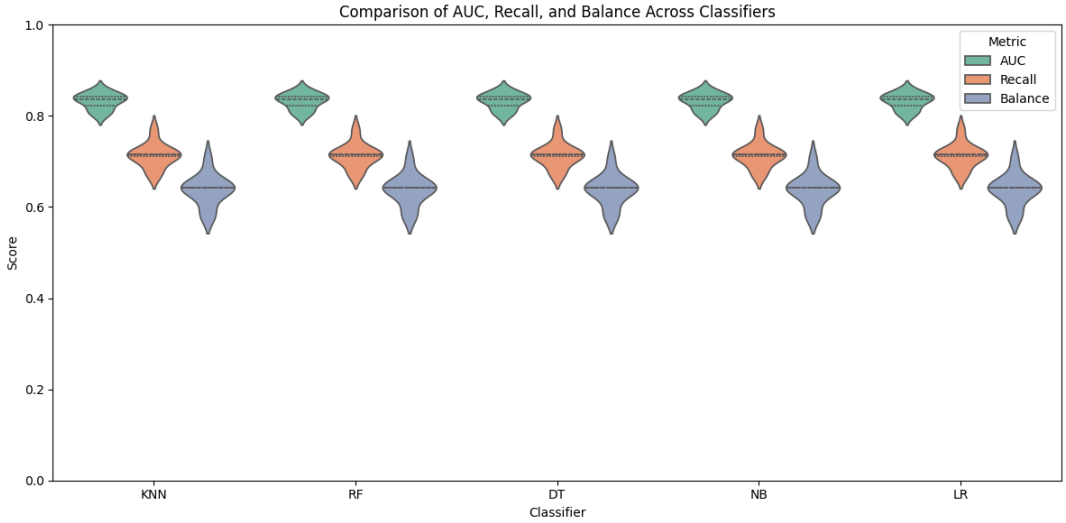
(a) Original result



(b) Intra-class balancing result

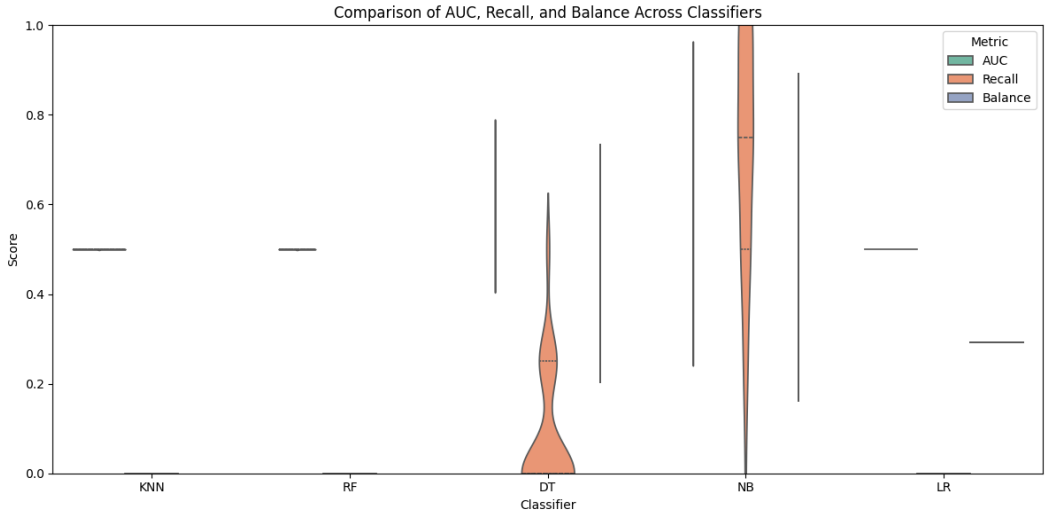


(c) Inter-class balancing result

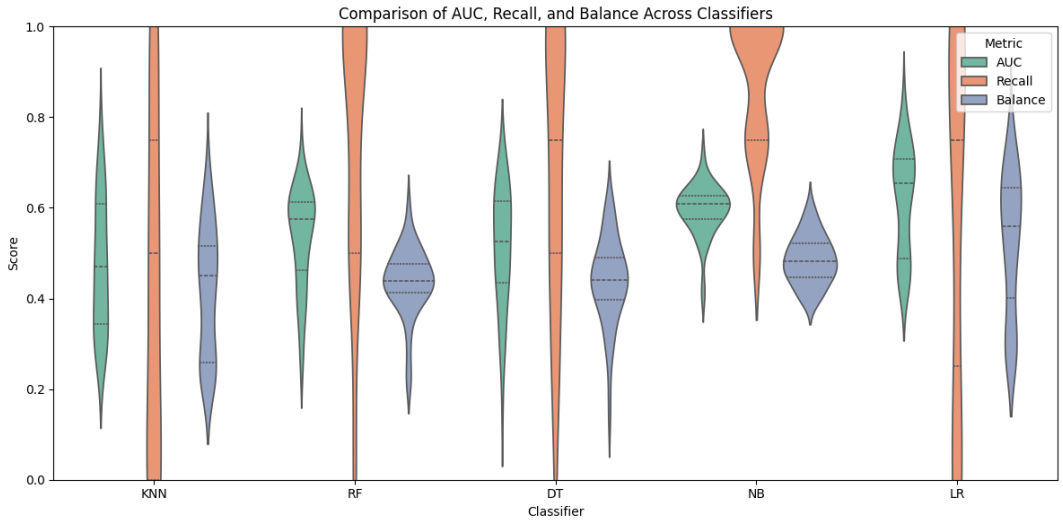


(d) I2SG result

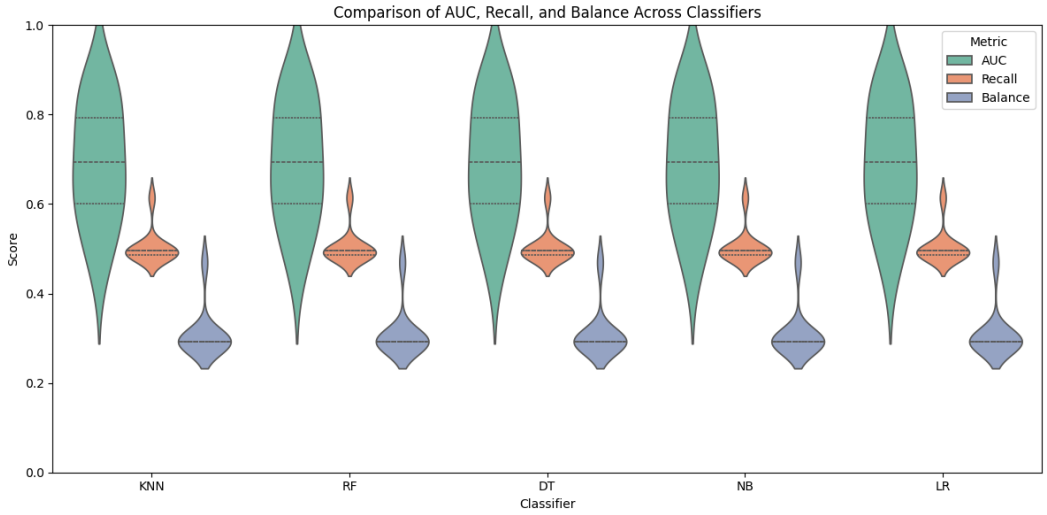
Figure 3 The violin plot of Table IV on MySQL dataset



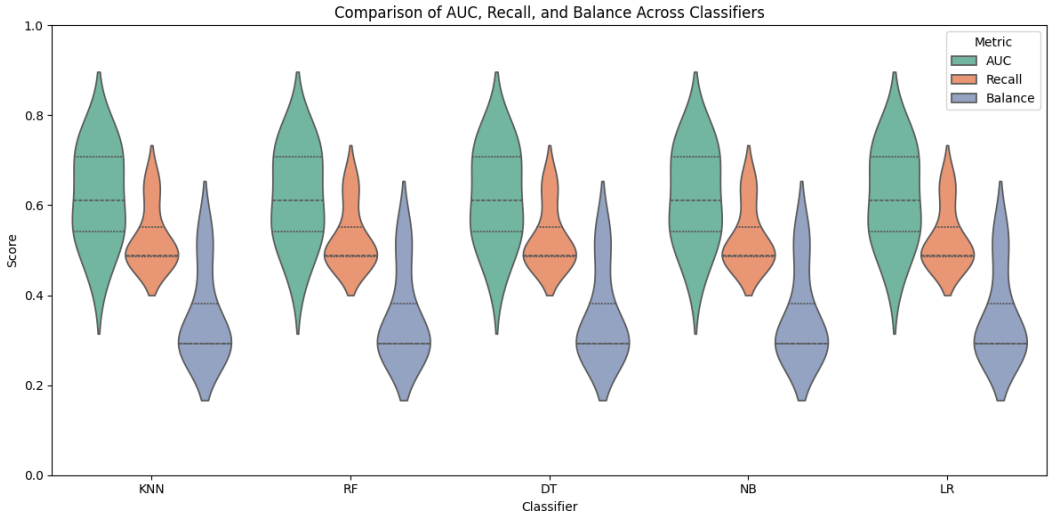
(a) Original result



(b) Intra-class balancing result



(c) Inter-class balancing result



(d) I2SG result

Figure 4 The violin plot of Table V on NetBSD dataset

5. The time required for training the model and the maximum memory consumption during the training process on Linux, MySQL and NetBSD datasets. Please note that the memory consumption is measured in MB, while the training time is in millisecond (ms), and the training time marked with an asterisk (*) is in seconds (s).

TABLE 4: Performance comparison of I2SG with different sampling methods

Classifier	Performance	ROS	RUS	SMOTE	AdaBoost	Adasyn	BSMOTE	VGAN	WGANGP	I2SG
Linux	Train time	32.86	11.51	36.71	2.53*	65.63	58.00	91.50*	74.31	4.27*
	Memory	95.75	126.51	126.63	164.93	95.79	95.79	678.82	146.92	320.45
MySQL	Train time	7.60	6.33	13.38	0.98*	29.19	33.08	99.77*	10.55	4.22*
	Memory	95.04	95.02	108.375	148.36	95.293	91.125	671.48	143.62	309.97
NetBSD	Train time	14.19	17.63	22.31	133.30	38.27	49.33	88.44*	41.48	4.27
	Memory	117.70	117.63	117.55	1.70*	120.25	119.02	787.06	295.43	309.28

From the above table, we can observe that in terms of training time, the I2SG model takes the longest to train, with a time of 4.27 seconds on Linux, which is the longest among all methods. The models with the shortest training times are ROS and RUS, which are the simplest sampling models. In terms of memory usage, the I2SG model shows a relatively high memory usage of 320.45 MB on Linux, which is significantly higher than other methods. This is because the model needs to process both intra-class and inter-class imbalances in two steps, leading to a higher complexity. On MySQL, the memory usage is 309.97 MB, still higher compared to other methods. On NetBSD, the memory usage is 309.28 MB, which is relatively consistent, but still higher than AdaBoost and WGAN, indicating some difference in efficiency. By comparing different resampling methods, it can be observed that ROS, RUS, and SMOTE generally have shorter training times. This is because they do not involve additional computational steps or parameters, so they require less time to process imbalanced data. On the other hand, methods like AdaBoost, VGAN, and WGANGP show longer training times and higher memory usage, especially on NetBSD, which is attributed to the parameter optimization of deep models. The more complex the model, the longer the optimization process.

6. The F-score to evaluate the overall performance of the methods. It is important to note that, in order to evaluate the model's performance on each class and reflect its overall performance across all classes, we use the macro average to calculate the F1 score. This is because we found that if we evaluate only the minority classes, the model's performance on all three datasets is almost zero, which is not useful for our model analysis.

For RQ1:

TABLE 5: The performance of I2SG model on Linux dataset in terms of F1

Classifier	Original	Intra-class	Inter-class	I2SG
KNN	0.499	0.317	0.498	0.438
RF	0.498	0.307	0.499	0.401
DT	0.513	0.281	0.498	0.414

NB	0.520	0.291	0.505	0.393
LR	0.499	0.327	0.494	0.481
AVG	0.506	0.305	0.499	0.425

TABLE 6: The performance of I2SG model on MySQL dataset in terms of F1

Classifier	Original	Intra-class	Inter-class	I2SG
KNN	0.568	0.520	0.478	0.673
RF	0.551	0.508	0.660	0.617
DT	0.591	0.487	0.602	0.584
NB	0.602	0.503	0.781	0.692
LR	0.541	0.536	0.627	0.697
AVG	0.571	0.511	0.630	0.653

TABLE 7: The performance of I2SG model on NetBSD dataset in terms of F1

Classifier	Original	Intra-class	Inter-class	I2SG
KNN	0.497	0.170	0.497	0.378
RF	0.497	0.136	0.506	0.362
DT	0.501	0.143	0.499	0.375
NB	0.503	0.113	0.485	0.503
LR	0.497	0.165	0.495	0.516
AVG	0.499	0.145	0.496	0.427

For RQ2:

TABLE 8: The performance of traditional sampling methods on Linux dataset in terms of F1

Classifier	ROS	RUS	SMOTE	AdaBoost	Adasyn	BSMOTE	I2SG
KNN	0.538	0.438	0.454	-	0.456	0.499	0.438
RF	0.504	0.457	0.501	0.499	0.502	0.499	0.401
DT	0.505	0.454	0.496	0.498	0.508	0.498	0.414
NB	0.496	0.477	0.499	0.408	0.498	0.502	0.393
LR	0.388	0.402	0.406	0.497	0.396	0.498	0.481
AVG	0.486	0.446	0.471	0.475	0.472	0.499	0.425

TABLE 9: The performance of traditional sampling methods on MySQL dataset in terms of F1

Classifier	ROS	RUS	SMOTE	AdaBoost	Adasyn	BSMOTE	I2SG
KNN	0.603	0.577	0.548	-	0.512	0.547	0.673
RF	0.727	0.596	0.698	0.638	0.690	0.716	0.617
DT	0.673	0.570	0.710	0.640	0.683	0.643	0.584
NB	0.750	0.730	0.746	0.678	0.737	0.743	0.692
LR	0.540	0.548	0.605	0.5549	0.55	0.567	0.697
AVG	0.659	0.604	0.661	0.628	0.634	0.643	0.653

TABLE 10: The performance of traditional sampling methods on NetBSD dataset in terms of F1

Classifier	ROS	RUS	SMOTE	AdaBoost	Adasyn	BSMOTE	I2SG
KNN	0.563	0.469	0.447	-	0.434	0.504	0.378
RF	0.524	0.452	0.522	0.497	0.52	0.544	0.362
DT	0.523	0.421	0.525	0.497	0.522	0.540	0.375
NB	0.429	0.459	0.399	0.523	0.390	0.480	0.503
LR	0.065	0.407	0.418	0.496	0.057	0.443	0.516
AVG	0.421	0.442	0.462	0.503	0.384	0.502	0.427

For RQ3:

TABLE 11: The performance of I2SG and GAN-based methods on Linux dataset in terms of F1

Classifier	CTGAN	VGAN	WGANGP	I2SG
KNN	0.498	0.555	0.499	0.438
RF	0.499	0.513	0.498	0.401
DT	0.498	0.510	0.498	0.414
NB	0.505	0.498	0.496	0.393
LR	0.494	0.480	0.498	0.481
AVG	0.499	0.511	0.498	0.425

TABLE 12: The performance of I2SG and GAN-based methods on MySQL dataset in terms of F1

Classifier	CTGAN	VGAN	WGANGP	I2SG
KNN	0.478	0.569	0.478	0.673
RF	0.660	0.605	0.589	0.617
DT	0.602	0.652	0.592	0.584
NB	0.781	0.621	0.620	0.692
LR	0.627	0.560	0.545	0.697
AVG	0.630	0.601	0.565	0.653

TABLE 13: The performance of I2SG and GAN-based methods on NetBSD dataset in terms of F1

Classifier	CTGAN	VGAN	WGANGP	I2SG
KNN	0.497	0.487	0.497	0.378
RF	0.506	0.496	0.497	0.362
DT	0.499	0.495	0.497	0.375
NB	0.485	0.413	0.497	0.503
LR	0.495	0.471	0.496	0.516
AVG	0.496	0.473	0.497	0.427

For RQ4:

TABLE 14: Comparison results on SDP datasets in terms of F1

Classifier	CM1	PC1	PDE	camle1.0	skarbonka	tomcat
Original	0.092	0.090	0.133	0.021	0.078	0.083
ROS	0.481	0.551	0.629	0.557	0.483	0.583
RUS	0.457	0.537	0.607	0.478	0.513	0.553
SMOTE	0.478	0.581	0.637	0.549	0.481	0.597
BSMOTE	0.487	0.581	0.634	0.559	0.488	0.587
AdaBoost	0.422	0.617	0.606	0.485	0.392	0.557
Adasyn	0.486	0.567	0.635	0.548	0.430	0.575
VGAN	0.544	0.558	0.562	0.515	0.596	0.608
CTGAN	0.464	0.642	0.629	0.541	0.532	0.582
WGANGP	0.495	0.578	0.601	0.498	0.489	0.574
I2SG	0.503	0.522	0.429	0.452	0.640	0.518

7. The comparative experiment between the proposed method and different sampling methods. Specifically, in the intra-class balancing stage, we use the clustering-based noise cleaning method proposed in the paper, while in the inter-class balancing stage, we apply eight methods we discussed in this paper (ROS, RUS, SMOTE, AdaBoost, Adasyn, BSMOTE, VGAN, WGANGP).

TABLE 15: Performance comparison of Intra-class balancing combined with different sampling methods on Linux dataset

Classifier	Performance	ROS	RUS	SMOTE	AdaBoost	Adasyn	BSMOTE	VGAN	WGANGP	I2SG
KNN	AUC	0.712	0.718	0.715	-	0.763	0.721	0.799	0.5	0.834
	Recall	1	1	1	-	0.997	0.933	0.749	0.5	0.781
	Balance	0.593	0.602	0.597	-	0.666	0.610	0.679	0.293	0.775
	F1	0.308	0.314	0.311	-	0.357	0.320	0.541	0.499	0.438
RF	AUC	0.702	0.706	0.706	0.859	0.810	0.714	0.833	0.476	0.843
	Recall	1	1	0.997	0.500	0.833	0.997	0.509	0.5	0.753
	Balance	0.579	0.584	0.587	0.293	0.788	0.597	0.679	0.293	0.724
	F1	0.297	0.301	0.303	0.498	0.462	0.311	0.502	0.499	0.401
DT	AUC	0.685	0.683	0.696	0.768	0.756	0.658	0.738	0.5	0.768
	Recall	0.970	0.997	0.983	0.500	0.720	0.943	0.533	0.5	0.772
	Balance	0.573	0.553	0.581	0.293	0.722	0.545	0.346	0.293	0.732
	F1	0.293	0.277	0.298	0.498	0.461	0.275	0.518	0.499	0.414
NB	AUC	0.763	0.774	0.777	0.389	0.777	0.764	0.772	0.5	0.844
	Recall	0.849	0.859	0.873	0.455	0.873	0.833	0.678	0.5	0.869
	Balance	0.724	0.737	0.741	0.308	0.743	0.721	0.587	0.293	0.815
	F1	0.417	0.423	0.420	0.456	0.420	0.424	0.513	0.006	0.393
LR	AUC	0.718	0.721	0.722	0.357	0.716	0.730	0.882	0.499	0.815
	Recall	1	1	1	0.500	0.700	0.993	0.743	0.458	0.684

	Balance	0.601	0.605	0.607	0.374	0.716	0.622	0.686	0.293	0.632
	F1	0.314	0.316	0.318	0.472	0.437	0.329	0.510	0.229	0.481
AVG	AUC	0.716	0.720	0.723	0.593	0.764	0.717	0.805	0.495	0.821
	Recall	0.964	0.971	0.971	0.489	0.825	0.939	0.642	0.492	0.772
	Balance	0.614	0.616	0.623	0.317	0.727	0.619	0.595	0.293	0.736
	F1	0.326	0.326	0.330	0.481	0.427	0.332	0.517	0.346	0.425

TABLE 16: Performance comparison of Intra-class balancing combined with different sampling methods on MySQL dataset

Classifier	Performance	ROS	RUS	SMOTE	AdaBoost	Adasyn	BSMOTE	VGAN	WGANGP	I2SG
KNN	AUC	0.749	0.761	0.756	-	0.750	0.742	0.746	0.5	0.861
	Recall	0.884	0.882	0.881	-	0.893	0.877	0.623	0.5	0.794
	Balance	0.709	0.728	0.714	-	0.705	0.704	0.54	0.293	0.789
	F1	0.520	0.535	0.531	-	0.515	0.514	0.590	0.078	0.673
RF	AUC	0.746	0.757	0.748	0.844	0.757	0.738	0.786	0.752	0.899
	Recall	0.905	0.886	0.884	0.556	0.818	0.877	0.543	0.566	0.852
	Balance	0.694	0.716	0.701	0.398	0.737	0.696	0.54	0.412	0.823
	F1	0.505	0.528	0.517	0.560	0.562	0.508	0.541	0.521	0.617
DT	AUC	0.729	0.701	0.714	0.845	0.709	0.707	0.662	0.584	0.820
	Recall	0.886	0.781	0.863	0.561	0.700	0.861	0.554	0.584	0.819
	Balance	0.677	0.656	0.659	0.407	0.686	0.654	0.398	0.452	0.800
	F1	0.492	0.502	0.483	0.573	0.550	0.475	0.555	0.529	0.584
NB	AUC	0.772	0.773	0.776	0.579	0.773	0.771	0.790	0.681	0.834
	Recall	0.870	0.865	0.881	0.534	0.865	0.882	0.676	0.5	0.807
	Balance	0.747	0.749	0.747	0.410	0.751	0.739	0.605	0.293	0.805
	F1	0.556	0.560	0.556	0.497	0.560	0.548	0.642	0.478	0.692
LR	AUC	0.771	0.766	0.775	0.513	0.771	0.756	0.776	0.359	0.845
	Recall	0.902	0.882	0.882	0.616	0.865	0.881	0.642	0.537	0.750
	Balance	0.732	0.735	0.742	0.494	0.746	0.722	0.575	0.293	0.732
	F1	0.539	0.542	0.555	0.618	0.558	0.530	0.601	0.506	0.697
AVG	AUC	0.753	0.752	0.754	0.695	0.752	0.743	0.752	0.5752	0.845
	Recall	0.889	0.859	0.878	0.567	0.828	0.876	0.608	0.537	0.750
	Balance	0.712	0.717	0.713	0.427	0.725	0.703	0.532	0.349	0.732
	F1	0.522	0.533	0.528	0.562	0.549	0.515	0.586	0.422	0.653

TABLE 17: Performance comparison of Intra-class balancing combined with different sampling methods on NetBSD dataset

Classifier	Performance	ROS	RUS	SMOTE	AdaBoost	Adasyn	BSMOTE	VGAN	WGANGP	I2SG
KNN	AUC	0.564	0.571	0.564	-	0.654	0.588	0.565	0.482	0.681

	Recall	0.958	0.915	0.964	-	0.661	1	0.556	0.500	0.661
	Balance	0.410	0.447	0.407	-	0.645	0.417	0.438	0.293	0.624
	F1	0.155	0.194	0.151	-	0.414	0.157	0.497	0.011	0.378
RF	AUC	0.537	0.552	0.541	0.636	0.624	0.551	0.630	0.582	0.701
	Recall	0.964	0.933	0.945	0.500	0.355	0.964	0.498	0.499	0.675
	Balance	0.368	0.407	0.387	0.293	0.535	0.390	0.431	0.293	0.679
	F1	0.109	0.154	0.132	0.497	0.505	0.132	0.496	0.351	0.362
DT	AUC	0.539	0.548	0.542	0.570	0.595	0.561	0.584	0.529	0.670
	Recall	0.976	0.915	0.945	0.500	0.361	0.927	0.498	0.504	0.670
	Balance	0.365	0.402	0.390	0.293	0.525	0.410	0.293	0.293	0.655
	F1	0.103	0.154	0.134	0.497	0.474	0.154	0.496	0.440	0.375
NB	AUC	0.646	0.632	0.600	0.501	0.752	0.711	0.484	0.352	0.645
	Recall	0.636	0.636	0.545	0.496	0.818	0.715	0.509	0.500	0.647
	Balance	0.645	0.632	0.596	0.383	0.744	0.689	0.328	0.293	0.647
	F1	0.417	0.406	0.413	0.450	0.437	0.442	0.499	0.157	0.503
LR	AUC	0.550	0.559	0.548	0.272	0.676	0.565	0.538	0.257	0.703
	Recall	0.948	0.915	0.945	0.488	0.624	0.964	0.555	0.452	0.677
	Balance	0.396	0.429	0.398	0.308	0.650	0.409	0.472	0.293	0.632
	F1	0.141	0.178	0.144	0.486	0.446	0.151	0.471	0.434	0.516
AVG	AUC	0.567	0.572	0.559	0.495	0.660	0.595	0.560	0.440	0.680
	Recall	0.896	0.863	0.869	0.496	0.564	0.914	0.523	0.491	0.666
	Balance	0.437	0.463	0.436	0.319	0.620	0.463	0.392	0.293	0.647
	F1	0.185	0.217	0.195	0.483	0.455	0.207	0.492	0.2786	0.427