Data preprocessing and dynamic ensemble selection for imbalanced data stream classification Statistical evaluation appendix

Paweł Zyblewski[0000−0002−4224−6709]1, Robert Sabourin[2], and Michał Woźniak[0000−0003−0146−4205]1

[1] Department of Systems and Computer Networks,
Faculty of Electronics, Wrocław University of Science and Technology,
Wrocław, Poland
{pawel.zyblewski,michal.wozniak}@pwr.edu.pl
[2] LIVIA,École de Technologie Supérieure, University of Quebec,
Montreal, Quebec, Canada
robert.sabourin@etsmtl.ca

1 Statistical evaluation

The results of experiments for two measures: BAC (a) and G-mean (b) for different IR values are presented in Tab. 1-4 and for different types of concept drifts in Tab. 5-6. For each experiment we compared different dynamic selection methods and oversampling algorithms for a particular DES approach. Bold indicates the statistically significant best combination method, while brackets are used for the statistically significant best preprocessing algorithm for a given combination strategy. Small numbers under the results indicate the indexes of methods that are statistically significantly outperformed by the considered combination strategy (best in row), while small letters stand for preprocessing methods that are statistically significantly outperformed by the considered one (best in column). The Wilcoxon Signed Rank Test (p-value$\leq .05$) was used.

1.1 Experiment 1 - Impact of imbalance rate

Table 1: Results for 1 : 9 IR

(a)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.650	0.717	0.729	**0.743**	0.725
	–	1	1,2,5	*All*	1,2
	g	–	–	*f*	–
SMOTE ($_b$)	0.664	0.741	**0.768**	0.762	0.754
	–	1	*All*	1,2,5	1,2
	a,d,f,g	*a,e,f,g*	*a,d,f*	*a,f,g*	*a,d,f,g*
SVM-SMOTE ($_c$)	0.677	[0.751]	**0.771**	[0.770]	[0.762]
	–	1	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*a,b,d,f,g*	*All*	*All*
B1-SMOTE ($_d$)	0.657	0.741	**0.763**	0.762	0.750
	–	1	*All*	1,2,5	1,2
	a,f,g	*a,e,f,g*	*a,f*	*a,f,g*	*a,f*
B2-SMOTE ($_e$)	[0.681]	0.738	**[0.772]**	0.763	0.755
	–	1	*All*	1,2,5	1,2
	All	*a,f*	*All*	*a,b,f,g*	*a,b,d,f,g*
SL-SMOTE ($_f$)	0.651	0.718	**0.740**	**0.741**	0.728
	–	1	1,2,5	1,2,5	1,2
	g	–	*a*	–	*a*
ADASYN ($_g$)	0.649	0.738	**0.768**	0.758	0.752
	–	1	*All*	1,2,5	1,2
	–	*a,f*	*a,d,f*	*a,f*	*a,d,f*

(b)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.544	0.683	0.676	**0.705**	0.679
	–	1,3,5	1	*All*	1,3
	g	–	–	–	–
SMOTE ($_b$)	0.569	0.729	0.742	**0.748**	0.733
	–	1	1,2,5	*All*	1,2
	a,d,f,g	*a,d,f*	*a,d,f*	*a,d,f*	*a,d,f*
SVM-SMOTE ($_c$)	0.591	[0.735]	0.742	**[0.752]**	0.738
	–	1	1,2,5	*All*	1,2
	a,b,d,f,g	*All*	*a,d,f*	*a,b,d,f,g*	*a,b,d,f,g*
B1-SMOTE ($_d$)	0.555	0.724	0.734	**0.744**	0.726
	–	1	1,2,5	*All*	1,2
	a,f,g	*a,f*	*a,f*	*a,f*	*a,f*
B2-SMOTE ($_e$)	[0.598]	0.729	**[0.751]**	**[0.752]**	[0.740]
	–	1	1,2,5	1,2,5	1,2
	All	*a,d,f*	*All*	*a,b,d,f,g*	*All*
SL-SMOTE ($_f$)	0.544	0.702	0.705	**0.723**	0.702
	–	1	1,2,5	*All*	1
	g	*a*	*a*	*a*	*a*
ADASYN ($_g$)	0.542	0.729	0.745	**0.748**	0.734
	–	1	1,2,5	*All*	1,2
	–	*a,d,f*	*a,b,c,d,f*	*a,d,f*	*a,d,f*

Table 2: Results for 2 : 8 IR

(a)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.744	0.779	0.809	**0.814**	0.800
	–	1	1,2,5	*all*	1,2
	–	*f*	–	*f*	–
SMOTE ($_b$)	0.757	0.793	**0.829**	0.820	0.815
	–	1	*All*	1,2,5	1,2
	a,d,f,g	*a,e,f,g*	*a,f*	*a,e,f,g*	*a,d,e,f,g*
SVM-SMOTE ($_c$)	0.771	[0.801]	**[0.833]**	[0.826]	[0.820]
	–	1	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*All*	*All*	*All*
B1-SMOTE ($_d$)	0.754	0.793	**0.829**	0.820	0.813
	–	1	*All*	1,2,5	1,2
	a,f,g	*a,e,f,g*	*a,f*	*a,b,e,f,g*	*a,e,f*
B2-SMOTE ($_e$)	[0.773]	0.782	**0.830**	0.814	0.811
	–	1	*All*	1,2,5	1,2
	All	*a,f*	*a,b,d,f,g*	*f*	*a,f*
SL-SMOTE ($_f$)	0.747	0.776	**0.819**	0.805	0.800
	–	1	*All*	1,2,5	1,2
	a,g	–	*a*	–	*a*
ADASYN ($_g$)	0.744	0.788	**0.830**	0.814	0.813
	–	1	*All*	1,2	1,2
	–	*a,e,f*	*a,b,d,f*	*f*	*a,e,f*

(b)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.704	0.768	0.792	**0.803**	0.783
	–	1	1,2,5	*All*	1,2
	–	–	–	*f*	–
SMOTE ($_b$)	0.724	0.789	**0.820**	0.816	0.807
	–	1	*All*	1,2,5	1,2
	a,d,f,g	*a,e,f,g*	*a,f*	*a,e,f,g*	*a,d,f*
SVM-SMOTE ($_c$)	0.744	[0.797]	**[0.825]**	[0.822]	[0.813]
	–	1	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*a,b,d,f,g*	*All*	*All*
B1-SMOTE ($_d$)	0.719	0.789	**0.821**	0.817	0.805
	–	1	*All*	1,2,5	1,2
	a,f,g	*a,e,f,g*	*a,b,f*	*a,b,e,f,g*	*a,f*
B2-SMOTE ($_e$)	[0.746]	0.780	**[0.825]**	0.812	0.806
	–	1	*All*	1,2,5	1,2
	All	*a,f*	*a,b,d,f,g*	*a,f*	*a,d,f*
SL-SMOTE ($_f$)	0.708	0.772	**0.809**	0.802	0.792
	–	1	*All*	1,2,5	1,2
	a,g	*a*	*a*	–	*a*
ADASYN ($_g$)	0.704	0.786	**0.822**	0.811	0.807
	–	1	*All*	1,2,5	1,2
	–	*a,e,f*	*a,b,d,f*	*a,f*	*a,d,f*

Table 3: Results for 3 : 7 IR

(a)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.800	0.806	**0.846**	0.844	0.834
	–	1	*All*	1,2,5	1,2
	–	*e,f*	–	*e,f,g*	*f*
SMOTE ($_b$)	0.806	0.815	**0.856**	0.846	0.841
	–	1	*All*	1,2,5	1,2
	a,f,g	*a,d,e,f,g*	*a,f*	*a,d,e,f,g*	*a,d,e,f,g*
SVM-SMOTE ($_c$)	0.816	[0.819]	**[0.858]**	[0.847]	[0.843]
	–	–	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*All*	*All*	*All*
B1-SMOTE ($_d$)	0.808	0.813	**0.856**	0.844	0.839
	–	1	*All*	1,2,5	1,2
	a,b,f,g	*a,e,f,g*	*a,b,e,f*	*e,f,g*	*a,e,f*
B2-SMOTE ($_e$)	[0.819]	0.800	**0.855**	0.836	0.835
	2	–	*All*	1,2	1,2
	All	–	*a,f*	*f*	*a,f*
SL-SMOTE ($_f$)	0.802	0.801	**0.850**	0.833	0.831
	2	–	*All*	1,2,5	1,2
	a,g	–	*a*	–	–
ADASYN ($_g$)	0.800	0.809	**0.856**	0.838	0.839
	–	1	*All*	1,2	1,2,4
	–	*a,e,f*	*a,b,e,f*	*e,f*	*a,e,f*

(b)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.786	0.803	0.840	**0.841**	0.828
	–	1	1,2,5	*All*	1,2
	–	*e,f*	–	*e,f,g*	–
SMOTE ($_b$)	0.794	0.814	**0.852**	0.844	0.838
	–	1	*All*	1,2,5	1,2
	a,f,g	*a,d,e,f,g*	*a,f*	*a,d,e,f,g*	*a,d,e,f,g*
SVM-SMOTE ($_c$)	0.807	[0.817]	**[0.855]**	[0.846]	[0.841]
	–	1	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*All*	*All*	*All*
B1-SMOTE ($_d$)	0.797	0.811	**0.853**	0.843	0.836
	–	1	*All*	1,2,5	1,2
	a,b,f,g	*a,e,f,g*	*a,b,f*	*a,e,f,g*	*a,e,f*
B2-SMOTE ($_e$)	[0.810]	0.799	**0.853**	0.835	0.833
	2	–	*All*	1,2,5	1,2
	All	–	*a,b,f*	*f*	*a,f*
SL-SMOTE ($_f$)	0.790	0.799	**0.847**	0.831	0.828
	–	1	*All*	1,2,5	1,2
	a,g	–	*a*	–	–
ADASYN ($_g$)	0.786	0.808	**0.853**	0.837	0.836
	–	1	*All*	1,2	1,2
	–	*a,e,f*	*a,b,f*	*e,f*	*a,e,f*

Table 4: Results for 4 : 6 IR

(a)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.827	0.819	**0.864**	[0.857]	0.851
	2	–	*All*	1,2,5	1,2
	–	*e,f,g*	–	*All*	*e,f*
SMOTE ($_b$)	0.828	0.823	**0.867**	0.856	0.853
	2	–	*All*	1,2,5	1,2
	a,f,g	*a,d,e,f,g*	*a,f*	*c,d,e,f,g*	*a,d,e,f,g*
SVM-SMOTE ($_c$)	0.834	[0.823]	**[0.868]**	0.856	[0.853]
	2	–	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*All*	*d,e,f,g*	*All*
B1-SMOTE ($_d$)	0.832	0.821	**0.867**	0.854	0.852
	2	–	*All*	1,2,5	1,2
	a,b,f,g	*a,e,f,g*	*a,b,e,f*	*e,f,g*	*a,e,f*
B2-SMOTE ($_e$)	[0.836]	0.811	**0.866**	0.848	0.848
	2	–	*All*	1,2	1,2
	All	–	*a,f*	–	*f*
SL-SMOTE ($_f$)	0.827	0.815	**0.864**	0.849	0.847
	2	–	*All*	1,2,5	1,2
	–	*e*	–	*e*	–
ADASYN ($_g$)	0.827	0.818	**0.868**	0.852	0.852
	2	–	*All*	1,2	1,2
	–	*e,f*	*a,b,e,f*	*e,f*	*a,e,f*

(b)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.822	0.818	**0.862**	[0.856]	0.849
	2	–	*All*	1,2,5	1,2
	–	*e,f*	–	*All*	*e,f*
SMOTE ($_b$)	0.824	0.822	**0.865**	0.855	0.851
	2	–	*All*	1,2,5	1,2
	a,f,g	*a,d,e,f,g*	*a,f*	*c,d,e,f,g*	*a,d,e,f*
SVM-SMOTE ($_c$)	0.831	[0.823]	**[0.867]**	0.854	[0.852]
	2	–	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*All*	*d,e,f,g*	*All*
B1-SMOTE ($_d$)	0.828	0.820	**0.866**	0.853	0.850
	2	–	*All*	1,2,5	1,2
	a,b,f,g	*a,e,f,g*	*a,b,e,f*	*e,f,g*	*a,e,f*
B2-SMOTE ($_e$)	[0.833]	0.810	**0.865**	0.847	0.847
	2	–	*All*	1,2	1,2,4
	All	–	*a,f*	–	*f*
SL-SMOTE ($_f$)	0.822	0.814	**0.863**	0.848	0.845
	2	–	*All*	1,2,5	1,2
	–	*e*	*a*	*e*	–
ADASYN ($_g$)	0.822	0.817	**0.866**	0.851	0.850
	2	–	*All*	1,2	1,2
	–	*e,f*	*a,b,e,f*	*e,f*	*a,e,f*

1.2 Experiment 2 - Drift type

As the purpose of the following experiment is to evaluate the proposed framework in the case of different types (sudden or incremental) concept drift appearance. We focus on the streams with high imbalance ratios (i.e., 1 : 9 and 2 : 8), typical for the real-life decision tasks.

Table 5: Results for sudden drift.

(a)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.717	0.756	0.780	**0.784**	0.774
	–	1	1,2,5	*All*	1,2
	g	*f*	–	*f*	–
SMOTE ($_b$)	0.732	0.771	**0.803**	0.793	0.790
	–	1	*All*	1,2,5	1,2
	a,d,f,g	*a,e,f,g*	*a,d,f,g*	*a,e,f,g*	*a,d,e,f,g*
SVM-SMOTE ($_c$)	0.746	[0.780]	**[0.807]**	[0.800]	[0.797]
	–	1	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*All*	*All*	*All*
B1-SMOTE ($_d$)	0.727	0.771	**0.801**	0.794	0.788
	–	1	*All*	1,2,5	1,2
	a,f,g	*a,e,f,g*	*a,f*	*a,e,f,g*	*a,e,f,g*
B2-SMOTE ($_e$)	[0.749]	0.763	**0.805**	0.789	0.786
	–	1	*All*	1,2,5	1,2
	All	*a,f*	*a,b,d,f,g*	*a,f,g*	*a,f*
SL-SMOTE ($_f$)	0.721	0.753	**0.792**	0.776	0.773
	–	1	*All*	1,2,5	1,2
	a,g	–	*a*	–	–
ADASYN ($_g$)	0.716	0.767	**0.802**	0.788	0.787
	–	1	*All*	1,2	1,2
	–	*a,e,f*	*a,d,f*	*a,f*	*a,f*

(b)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.657	0.735	0.750	**0.762**	0.748
	–	1	1,2,5	*All*	1,2
	g	–	–	–	–
SMOTE ($_b$)	0.679	0.764	**0.787**	0.784	0.777
	–	1	*All*	1,2,5	1,2
	a,d,f,g	*a,d,e,f,g*	*a,d,f*	*a,d,e,f,g*	*a,d,f*
SVM-SMOTE ($_c$)	0.700	[0.771]	**0.790**	[0.789]	[0.783]
	–	1	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*a,b,d,f,g*	*All*	*All*
B1-SMOTE ($_d$)	0.672	0.761	**0.783**	0.783	0.773
	–	1	*All*	1,2,5	1,2
	a,f,g	*a,e,f*	*a,f*	*a,f,g*	*a,f*
B2-SMOTE ($_e$)	[0.706]	0.758	**[0.793]**	0.783	0.778
	–	1	*All*	1,2,5	1,2
	All	*a,f*	*All*	*a,d,f,g*	*a,d,f,g*
SL-SMOTE ($_f$)	0.662	0.745	**0.775**	0.767	0.761
	–	1	*All*	1,2,5	1,2
	a,g	*a*	*a*	*a*	*a*
ADASYN ($_g$)	0.656	0.761	**0.788**	0.781	0.776
	–	1	*All*	1,2,5	1,2
	–	*a,e,f*	*a,b,d,f*	*a,f*	*a,d,f*

Table 6: Results for incremental drift

(a)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.677	0.741	0.757	**0.773**	0.751
	–	1	1,2,5	*All*	1,2
	–	–	–	*f*	–
SMOTE ($_b$)	0.689	0.762	**0.793**	0.788	0.778
	–	1	*All*	1,2,5	1,2
	a,d,f,g	*a,e,f,g*	*a,d,f*	*a,f,g*	*a,d,f*
SVM-SMOTE ($_c$)	0.703	[0.771]	**0.796**	[0.796]	[0.785]
	–	1	*All*	1,2,5	1,2
	a,b,d,f,g	*All*	*a,b,d,f,g*	*All*	*All*
B1-SMOTE ($_d$)	0.684	0.762	**0.791**	0.789	0.775
	–	1	*All*	1,2,5	1,2
	a,f,g	*a,e,f,g*	*a,f*	*a,e,f,g*	*a,f*
B2-SMOTE ($_e$)	[0.704]	0.757	**[0.797]**	0.789	0.780
	–	1	*All*	1,2,5	1,2
	All	*a,f*	*All*	*a,f,g*	*a,b,d,f,g*
SL-SMOTE ($_f$)	0.677	0.741	0.767	**0.770**	0.756
	–	1	1,2,5	*All*	1,2
	g	–	*a*	–	*a*
ADASYN ($_g$)	0.676	0.759	**0.795**	0.784	0.778
	–	1	*All*	1,2,5	1,2
	–	*a,e,f*	*a,b,d,f*	*a,f*	*a,d,f*

(b)

Preprocessing method	Naive	KNORA-E	KNORA-U	DES-KNN	DES-CLustering
	(1)	(2)	(3)	(4)	(5)
None ($_a$)	0.592	0.717	0.718	**0.746**	0.714
	–	1	1,2,5	*All*	1
	–	–	–	–	–
SMOTE ($_b$)	0.613	0.754	0.775	**0.780**	0.763
	–	1	1,2,5	*All*	1,2
	a,d,f,g	*a,d,e,f,g*	*a,d,f*	*a,d,f,g*	*a,d,f*
SVM-SMOTE ($_c$)	0.635	[0.761]	0.777	**[0.784]**	[0.768]
	–	1	1,2,5	*All*	1,2
	a,b,d,f,g	*All*	*a,b,d,f*	*All*	*a,b,d,f,g*
B1-SMOTE ($_d$)	0.602	0.752	0.772	**0.778**	0.758
	–	1	1,2,5	*All*	1,2
	a,f,g	*a,e,f*	*a,f*	*a,f*	*a,f*
B2-SMOTE ($_e$)	[0.638]	0.751	**[0.783]**	0.782	[0.769]
	–	1	*All*	1,2,5	1,2
	All	*a,f*	*All*	*a,b,d,f,g*	*a,b,d,f,g*
SL-SMOTE ($_f$)	0.591	0.729	0.739	**0.757**	0.733
	–	1	1,2,5	*All*	1,2
	–	*a*	*a*	*a*	*a*
ADASYN ($_g$)	0.590	0.753	**0.779**	0.779	0.766
	–	1	*All*	1,2,5	1,2
	–	*a,e,f*	*a,b,c,d,f*	*a,f*	*a,b,d,f*

Acknowledgement

This work was supported by the Polish National Science Centre under the grant No. 2017/27/B/ST6/01325 as well as by the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Science and Technology.