Experiments of Ensemble Adversarial Learning on Benchmark Problems

TABLE I
SUMMARY OF THE BENCHMARK IMBALANCED DATASETS USED.

Dataset	Number of samples	Number of attributes	Imbalance ratio
yeast-0-5-6-7-9_vs_4	498	8	22.71
yeast2vs8	474	8	38.50
yeast6	1474	8	57.96
abalone19	4174	8	129.44
vowel0	936	13	23.63
vehicle0	666	18	34.05
segment0	2008	19	68.24
autos	153	25	50.00
dermatology-6	344	34	56.33
kddcup-buffer_overflow_vs_back	2212	41	244.78
kddcup-rootkit-imap_vs_back	2205	41	1101.5
SRBCT	75	2308	24.00
LUNG2	187	3312	45.75
CAR	165	9182	81.50
BULL	92	17404	45.00

A. Experiments on Benchmark Problems

We first test the proposed GAN ensemble method on 15 datasets, including 11 KEEL data sets from [1] and 4 DNA microarray data sets from [2], [3], which are summarized in Table I. Since the proposed GAN ensemble classifier targets highly imbalanced problems, we remove some minority samples from the original data sets to increase the imbalance ratio.

The following nine popular classification techniques are used for comparison:

- The synthetic minority oversampling technique (SMOTE) [4], one of the most well-known oversampling techniques for imbalanced classification.
- The adaptive synthetic (ADASYN) sampling approach that improves learning from imbalanced data sets by generating more synthetic data for more difficult minority class examples [5].
- The granular support vector machines repetitive undersampling technique (GSVM-RU) [6].
- EasyEnsemble [7], an ensemble-based undersampling that samples several subsets from the majority class and trains a learner on each of them.
- IIvotes [8], a rule-based ensemble with selective data preprocessing.
- EUSBoost [9], an ensemble construction technique that improves RUSBoost using evolutionary undersampling.
- An ensemble of cost-sensitive decision trees (CSTrees) which are trained on random feature subspaces [10].
- The iterative instance adjustment for imbalanced domains (IPADE-ID) [11], an evolutionary technique using iterative instance generation and learning.
- BalEnsemble, an ensemble method that converts an imbalanced data set into multiple balanced ones and builds multiple classifiers on them [12].

For each of the ensemble-based technique, we fine tune the number of ensemble members between [5,30] on each benchmark problem. For our evolutionary algorithm for GAN ensemble fusion, we set $c_r = 0.95$, $m_r = 0.015$, $\eta_{\min} = 0.35$, $\eta_{\max} = 0.75$, $\widehat{g} = 12$, $NP = 30$, and maximum number of generations of is set to 200. The other control parameters of the comparative techniques are set as suggested in the literature, and the test uses a five-fold cross-validation strategy. The data set is stored in an IBM Storwize V7000 storage server (with 24×600G 15K SAS disk, a 300G STEC SSD, and 64GB cache), and the computational environment is a LenovoSystem x3850 X6 server (with 4×Intel Xeon 4830 CPU, 32GB DDR4 memory, and Windows Server NT 6.2 operating system).

The experimental results are evaluated based on the sensitivity measure that denotes what percentage of minority samples are identified as such and the specificity measure that denotes what percentage of majority samples are identified as such:

$$\textit{sensitivity} = \frac{TP}{TP + FN} \tag{1}$$

$$\textit{specificity} = \frac{TN}{FP + TN} \tag{2}$$

where *TP*, *FP*, *TN* and *FN* refer to true positives, false positives, true negatives and false negatives, respectively.

We also use a combined measure, the Area Under the receiver operating characteristic Curve (AUC) [13], which evidences that increasing the number of *TP* without also increasing the number of *FP* and thus is widely used in imbalanced problems:

$$\text{AUC} = \frac{\textit{sensitivity} + \textit{specificity}}{2} \tag{3}$$

Tables II, III, and IV present the *sensitivity*, *specificity*, and AUC results of our GANEnsemble method and the other nine comparative methods on the benchmark problems, respectively. On each benchmark problem, the best result(s) among the ten methods is shown in boldface.

As seen in Table II, the proposed GANEnsemble method achieves the best sensitivity values on 11 benchmark problems; for the remaining 4 problems, the sensitivity values of GANEnsemble are the second best on yeast-0-5-6-7-9_vs_4, yeast6 (lower than the BalEnsemble method) and dermatology-6 (lower than ADASYN), and is worse than four other methods on vehicle0. Except dermatology-6, the other three datasets where GANEnsemble does not perform the best typically have relatively low dimensions and/or imbalance ratios. In general, among the ten comparative methods, GANEnsemble

and BalEnsemble show the best performance in identifying minority samples, while GANEnsemble has a more significant performance advantage on datasets with high dimension and/or a high imbalance ratio.

In Table III we see that our GANEnsemble method shows more promising performance in terms of low misclassification rate: Its specificity values are the best on 12 benchmark problems, the second best on 2 problems (SRBCT and LUNG2), and is worse than three other problems on the remaining problem (vowel0). Note that vowel0 is also a dataset with a relatively low dimension and imbalance ratio. For SRBCT and LUNG2, we only use 3 and 4 minority samples which are all correctly identified by GANEnsemble, and the numbers of majority samples misclassified as minority are only 1 and 2, respectively, which can be verified without much effort.

Regarding the comprehensive AUC results shown in Table IV, the performance of GANEnsemble is the best on nine benchmark problems, the second best on 4 problems including yeast-0-5-6-7-9_vs_4, yeast6 (lower than BalEnsemble), vowel0 (lower than SMOTE) and dermatology-6 (lower than ADASYN), and is worse than three other methods on the SRBCT problem.

In summary, the overall performance of the proposed GANEnsemble method is the best among the ten comparative methods, and the performance advantage is more obvious on higher dimensional and more imbalanced problems. Therefore, it is expected that GANEnsemble can be one of the most effective methods for extremely imbalanced problems such as terrorist identification.

REFERENCES

[1] J. Alcala-Fdez, A. Ferndedez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, "KEEL data-mining software tool: data set repository integration of algorithmsand experimental analysis framework," *J. Multi-Valued Logic Soft Comput.*, no. 2-3, pp. 255–287, 2011.

[2] L. Bullinger, K. Döhner, E. Bair, S. Fröhling, R. F. Schlenk, R. Tibshirani, H. Döhner, and J. R. Pollack, "Use of gene-expression profiling to identify prognostic subclasses in adult acute myeloid leukemia," *New England J. Med.*, vol. 350, no. 16, pp. 1605–1616, 2004.

[3] K. Yang, Z. Cai, J. Li, and G. Lin, "A stable gene selection in microarray data analysis," *BMC Bioinform*, p. 228, 2006.

[4] N. Chawla, L. Hall, K. Bowyer, and W. Kegelmeyer, "SMOTE: synthetic minority oversampling technique," *J. Artif. Intell. Res.*, pp. 321–357, 2002.

[5] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IJCNN*, 2008, pp. 1322–1328.

[6] Y. Tang, Y. Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 39, no. 1, pp. 281–288, 2009.

[7] X. Y. Liu, J. Wu, and Z. H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 39, no. 2, pp. 539–550, 2009.

[8] J. Błaszczyński, M. Deckert, J. Stefanowski, and S. Wilk, "Integrating selective pre-processing of imbalanced data with ivotes ensemble," in *Rough Sets and Current Trends in Computing*, M. Szczuka, M. Kryszkiewicz, S. Ramanna, R. Jensen, and Q. Hu, Eds. Berlin, Heidelberg: Springer, 2010, pp. 148–157.

[9] M. Galar, A. Fernndez, E. Barrenechea, and F. Herrera, "EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," *Pattern Recog.*, vol. 46, no. 12, pp. 3460–3471, 2013.

[10] B. Krawczyk, M. Wozniak, and G. Schaefer, "Cost-sensitive decision tree ensembles for effective imbalanced classification," *Appl. Soft Comput.*, vol. 14, pp. 554–562, 2014.

[11] V. López, I. Triguero, C. J. Carmona, S. García, and F. Herrera, "Addressing imbalanced classification with instance generation techniques: IPADE-ID," *Neurocomputing*, vol. 126, pp. 15–28, 2014.

[12] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recog.*, vol. 48, no. 5, pp. 1623–1637, 2015.

[13] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, 2005.

TABLE II
THE SENSITIVITY RESULTS OF THE COMPARATIVE CLASSIFIERS ON THE BENCHMARK IMBALANCED DATASETS.

Dataset	SMOTE	ADASYN	GSVM-RU	EasyEnsemble	IIvotes	EUSBoost	CSTrees	IPADE-ID	BalEnsemble	GANEnsemble
yeast-0-5-6-7-9_vs_4	0.4286	0.5714	0.7143	0.5714	0.6190	0.6667	0.6190	0.7143	**0.7619**	0.7143
yeast2vs8	0.6667	0.5000	0.6667	0.5833	0.5833	0.6667	**0.7500**	0.5000	**0.7500**	**0.7500**
yeast6	0.6400	0.6000	0.6800	0.6400	0.6800	0.8000	0.7200	0.6800	**0.8400**	0.8000
abalone19	0.3438	0.4375	0.3750	0.6250	0.5625	0.5938	0.5938	0.4063	0.6250	**0.6875**
vowel0	**0.9737**	0.9474	0.9737	0.8947	0.9211	0.9474	0.9211	0.9474	**0.9737**	**0.9737**
vehicle0	0.7895	0.7895	0.7895	0.8421	**0.8947**	0.8421	0.7895	0.7895	0.8421	0.7895
segment0	0.6897	0.6897	0.6207	0.6552	0.6552	0.6897	0.7241	0.6552	0.7931	**0.8621**
autos	**1.0000**	**1.0000**	0.6667	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
dermatology-6	0.8333	**1.0000**	0.6667	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333	0.8333
kddcup-buffer_overflow_vs_back	0.6667	0.6667	0.5556	**0.7778**	0.6667	**0.7778**	**0.7778**	0.6667	**0.7778**	**0.7778**
kddcup-rootkit-imap_vs_back	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
SRBCT	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
LUNG2	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CAR	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
BULL	0.5000	0.5000	0.5000	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

TABLE III
THE SPECIFICITY RESULTS OF THE COMPARATIVE CLASSIFIERS ON THE BENCHMARK IMBALANCED DATASETS.

Dataset	SMOTE	ADASYN	GSVM-RU	EasyEnsemble	IIvotes	EUSBoost	CSTrees	IPADE-ID	BalEnsemble	GANEnsemble
yeast-0-5-6-7-9_vs_4	0.8910	0.8973	0.8700	0.9015	0.9078	0.8952	0.9057	0.8994	0.8952	**0.9245**
yeast2vs8	0.9004	0.8961	0.8810	0.9069	0.9091	0.9004	0.9134	0.8918	0.9091	**0.9372**
yeast6	0.8951	0.9041	0.9013	0.9041	0.9082	0.9048	0.9103	0.9137	0.9096	**0.9406**
abalone19	0.7154	0.6540	0.6922	0.7653	0.7593	0.7429	0.7675	0.7429	0.7525	**0.9099**
vowel0	0.9955	0.9944	0.9922	0.9911	0.9889	0.9900	0.9933	**0.9967**	0.9911	0.9933
vehicle0	0.9567	0.8794	0.8733	0.8903	0.8964	0.8887	0.8995	0.8949	0.9073	**0.9753**
segment0	0.8378	0.8585	0.8353	0.8999	0.9096	0.8964	0.9075	0.8898	0.8449	**0.9520**
autos	0.5533	0.6600	0.6133	0.6933	0.6733	0.6200	0.7400	0.6000	0.7600	**0.9400**
dermatology-6	0.8550	0.8166	0.8491	0.8787	0.8580	0.8314	0.8757	0.8935	0.9142	**0.9645**
kddcup-buffer_overflow_vs_back	0.7635	0.8021	0.7581	0.7844	0.8112	0.7726	0.8012	0.7935	0.7190	**0.9142**
kddcup-rootkit-imap_vs_back	0.7326	0.7612	0.7422	0.7649	0.7826	0.7594	0.7966	0.8216	0.7726	**0.9383**
SRBCT	**1.0000**	0.9583	0.9444	0.9722	**1.0000**	0.9583	**1.0000**	0.9444	0.9722	0.9861
LUNG2	0.9727	0.9672	0.9290	0.9781	0.9836	0.9891	0.9891	**0.9945**	0.9727	0.9891
CAR	0.9080	0.9202	0.7791	0.9264	0.9387	**0.9509**	0.9387	0.9448	0.9080	**0.9509**
BULL	0.9000	0.9111	0.7889	0.7778	0.8222	0.8333	0.8000	0.8556	0.8222	**0.9444**

TABLE IV
THE AUC RESULTS OF THE COMPARATIVE CLASSIFIERS ON THE BENCHMARK IMBALANCED DATASETS.

Dataset	SMOTE	ADASYN	GSVM-RU	EasyEnsemble	IIvotes	EUSBoost	CSTrees	IPADE-ID	BalEnsemble	GANEnsemble
yeast-0-5-6-7-9_vs_4	0.6598	0.7344	0.7922	0.7364	0.7634	0.7809	0.7624	0.8068	**0.8285**	0.8194
yeast2vs8	0.7835	0.6981	0.7738	0.7451	0.7462	0.7835	0.8317	0.6959	0.8295	**0.8436**
yeast6	0.7676	0.7520	0.7907	0.7720	0.7941	0.8524	0.8151	0.7969	**0.8748**	0.8703
abalone19	0.5296	0.5458	0.5336	0.6952	0.6609	0.6683	0.6806	0.5746	0.6888	**0.7987**
vowel0	**0.9846**	0.9709	0.9829	0.9429	0.9550	0.9687	0.9572	0.9720	0.9824	0.9835
vehicle0	0.8731	0.8345	0.8314	0.8662	**0.8956**	0.8654	0.8445	0.8422	0.8747	0.8824
segment0	0.7637	0.7741	0.7280	0.7776	0.7824	0.7930	0.8158	0.7725	0.8190	**0.9070**
autos	0.7767	0.8300	0.6400	0.8467	0.8367	0.8100	0.8700	0.8000	0.8800	**0.9700**
dermatology-6	0.8442	**0.9083**	0.7579	0.8560	0.8457	0.8323	0.8545	0.8634	0.8738	0.8989
kddcup-buffer_overflow_vs_back	0.7151	0.7344	0.6568	0.7811	0.7389	0.7752	0.7895	0.7301	0.7484	**0.8460**
kddcup-rootkit-imap_vs_back	0.8663	0.8806	0.8711	0.8824	0.8913	0.8797	0.8983	0.9108	0.8863	**0.9691**
SRBCT	**1.0000**	0.9792	0.9722	0.9861	**1.0000**	0.9792	**1.0000**	0.9722	0.9861	0.9931
LUNG2	0.9863	0.9836	0.9645	0.9891	0.9918	**0.9945**	**0.9945**	0.9973	0.9863	**0.9945**
CAR	0.9540	0.9601	0.8896	0.9632	0.9693	0.9755	0.9693	0.9724	0.9540	**0.9755**
BULL	0.7000	0.7056	0.6444	0.8889	0.9111	0.9167	0.9000	0.9278	0.9111	**0.9722**