Supplementary Material for "Causal Feature Selection with Dual Correction"

Xianjie Guo, Kui Yu*, Lin Liu, Fuyuan Cao, and Jiuyong Li

S-1: Tracing the DCMB algorithm

In this section, we give a tracing example as shown in Fig. 1 to show how DCMB works. In Fig. 1, the yellow feature denotes the class variable and the true MB of the class variable in the BN is highlighted in orange. First, Fig. 1(a) gives an example of a simple BN including 9 features, i.e., $\boldsymbol{F}$=$\{T, A, B, O, D, E, G, H, N\}$. Assuming T is the class variable, then $\boldsymbol{MB}(T)$=$\{A, B, O, D\}$. Let $k\_or = 0.5$ and $k\_and = 0.5$, using the BN in Fig. 1(a), DCMB is implemented as follows.

Algorithm 1 DCMB

Input: C: the class variable; $k\_or \in$ [0,1]; $k\_and \in$ [0,1]
Output: MB of C
{*Phase I: Identify candidate parents and children*}
1: [***or_rank***, ***CPC***] = *IdenCPC*(C)
{*Phase II: The "OR" rule for recovering discarded PC*}
2: ***orPC*** = *ORPC*(k_or, ***or_rank***)
{*Phase III: The "AND" rule for removing false PC*}
3: ***andCPC*** = *ANDPC*(k_and, ***CPC***)
4: ***PC*** = ***andCPC*** $\cup$ ***orPC***
{*Phase IV: Find spouses*}
5: ***SP*** = $\emptyset$
6: **for** *each* $X \in$ ***PC*** **do**
7: **for** *each* $Y \in$ ***PC***(X) and $Y \notin$ ***PC*** **do**
8: **if** $\exists$ ***S*** s.t. $C \perp\!\!\!\perp Y|\boldsymbol{S}$ and $C \not\perp\!\!\!\perp Y|\boldsymbol{S} \cup \{X\}$ **then**
9: ***SP*** $\longleftarrow$ ***SP*** $\cup \{Y\}$
10: **end if**
11: **end for**
12: **end for**
13: ***MB*** = ***PC*** $\cup$ ***SP***

Algorithm 2 *IdenCPC*

Input: C; ***F***: union of features and class variable
Output: ***or_rank***: possibly discarded true positives;
CPC: candidate parents and children features
1: Initialize ***or_rank*** = $\emptyset$, ***CPC*** = $\emptyset$, ***F*** = ***F*** $\setminus \{C\}$
{*Step 1: Forward step*}
2: **repeat**
3: **for** each $X \in$ ***F*** **do**
4: [*Dep*[X], *Sep*[X]] = $\arg\min_{\boldsymbol{S} \subseteq \boldsymbol{CPC}} dep(C, X|\boldsymbol{S})$
5: **if** $C \perp\!\!\!\perp X|Sep[X]$ **then**
6: ***F*** = ***F*** $\setminus \{X\}$
7: **if** *Sep*[X] $\neq \emptyset$ **then**
8: ***or_rank*** $\longleftarrow$ ***or_rank*** $\cup \{X\}$
9: **end if**
10: **end if**
11: **end for**
12: $Y = \arg\max_{X \in \boldsymbol{F}} Dep(X)$
13: ***CPC*** = ***CPC*** $\cup \{Y\}$
14: ***F*** = ***F*** $\setminus \{Y\}$
15: **until** ***F*** = $\emptyset$
{*Step 2: Backward step*}
16: **for** each $X \in$ ***CPC*** **do**
17: **if** $\exists \boldsymbol{S} \subseteq \boldsymbol{CPC} \setminus \{X\}$ such that $C \perp\!\!\!\perp X|\boldsymbol{S}$ **then**
18: ***CPC*** = ***CPC*** $\setminus \{X\}$
19: ***or_rank*** $\longleftarrow$ ***or_rank*** $\cup \{X\}$
20: **end if**
21: **end for**

Algorithm 3 *ORPC*

Input: $k\_or \in$ [0,1]; ***or_rank***
Output: ***orPC***: recovered PC by the "OR" rule
1: Initialize ***orPC*** = $\emptyset$
/**Descending order, F_1 has the highest dependency**/
2: $\langle F_1, ..., F_{|\boldsymbol{or\_rank}|} \rangle \longleftarrow$ ***or_rank***
3: **for** i = 1 to $R(|\boldsymbol{or\_rank}| \times k\_or)$ **do**
4: [***or_rank2***, ***CPC2***] = *IdenPC*(F_i)
5: **if** $C \in$ ***CPC2*** **then**
6: ***orPC*** = ***orPC*** $\cup \{F_i\}$
7: **end if**
8: **end for**

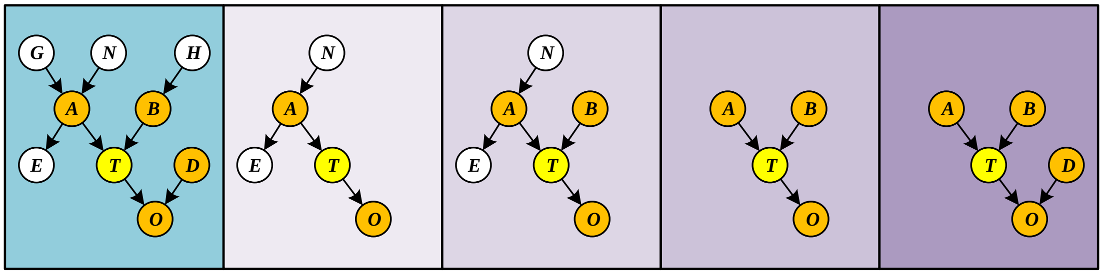
1) **Phase I**: At Line 1 of Algorithm 1, IdenCPC (Algorithm 2) is implemented to discover the candidate parents and children of T (***CPC***) while adding the features currently discarded to ***or_rank*** which contains the possibly discarded MB features. Initially, at Line 1 of Algorithm 2, let ***CPC*** = $\emptyset$, ***or_rank*** = $\emptyset$ and ***F***=$\{A, B, O, D, E, G, H, N\}$. After running Lines 3-14 of Algorithm 2 for the first time,

X. Guo and K. Yu,Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology) and School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, 230601, China; emails: xianjieguo@mail.hfut.edu.cn, yukui@hfut.edu.cn (*Corresponding author: Kui Yu).

L. Liu and J. Li, UniSA STEM, University of South Australia, Adelaide, 5095, Australia; emails: {Lin.Liu, Jiuyong.Li}@unisa.edu.au.

F. Cao, School of Computer and Information Technology, Shanxi University, Taiyuan, 030006, China; emails: cfy@sxu.edu.cn.

Manuscript received * *, *; revised * *, *.



(a) a simple BN (b) Phase I (c) Phase II (d) Phase III (e) Phase IV

Fig. 1. An example of tracing DCMB. (a) shows a simple BN; (b), (c), (d) and (e) demonstrate how to trace Phases I, II, III, and IV.

since $D \perp\!\!\!\perp T|\emptyset$ and $H \perp\!\!\!\perp T|\emptyset$ hold, $\{D, H\}$ are neither added to ***CPC*** nor to ***or_rank***. Meanwhile, $O \not\perp\!\!\!\perp T|\emptyset$ and O has the maximum relevancy with T among the features in $\boldsymbol{F}\backslash\boldsymbol{CPC}$, thus O is added to ***CPC***. At present, ***F***=$\{A, B, E, G, N\}$, ***CPC***=$\{O\}$ and ***or_rank*** = $\emptyset$. After implementing Lines 3-14 of Algorithm 2 multiple times, G, A, E and N are added to ***CPC*** successively. Clearly, if and only if $E \not\perp\!\!\!\perp T|A$ and $N \not\perp\!\!\!\perp T|A$ hold (false positives error), E and N can be added to ***CPC***. At this time, since $B \perp\!\!\!\perp T|O$ holds (false negatives error), B is not added to ***CPC*** but it is added to ***or_rank*** (Line 8 of Algorithm 2). When Step 1 of Algorithm 2 is finished, ***F***=$\emptyset$, ***CPC***=$\{O, G, A, E, N\}$ and ***or_rank*** = $\{B\}$. In Step 2 of Algorithm 2, as $G \perp\!\!\!\perp T|A$ holds, G is also added to ***or_rank*** after G is removed from ***CPC*** (Lines 17-20 of Algorithm 2). Finally, as shown in Figure 1(b), we get ***CPC*** = $\{O, A, E, N\}$ and ***or_rank*** = $\{B, G\}$.

2) **Phase II**: At Line 2 of Algorithm 1, ORPC (Algorithm 3) runs to recover the discarded MB features from ***or_rank***. By sorting the features within ***or_rank*** in a descending order at Line 2 of Algorithm 3, we obtain $\langle B, G\rangle \longleftarrow$ ***or_rank*** (i.e., the dependency between B and T is higher than that between G and T). The ORPC algorithm only needs to examine whether ***CPC*** of B (***CPC***(B)) contains T owing to $k\_or = 0.5$ (i.e., *R*(|***or_rank***| $\times$ k_or)=1). Since $T \in$ ***CPC***(B) holds, as shown in Figure 1(c), we retrieve B and get ***orPC*** = $\{B\}$. We can see that ORPC successfully avoids adding G to ***orPC*** even if $T \in$ ***CPC***(G) also holds. In addition, ORPC saves the computational cost of discovering ***CPC***(G).
3) **Phase III**: At Line 3 of Algorithm 1, the ANDPC algorithm (Algorithm 4) aims to remove the false MB features from ***CPC***. At Line 1 of Algorithm 4, ***andCPC***=$\{O, A, E, N\}$. By sorting the features within ***CPC*** in an ascending order at Line 2 of Algorithm 4, we get $\langle E, N, A, O\rangle \longleftarrow$ ***or_rank*** (i.e., E has the lowest dependency with T). Since k_and = 0.5 (i.e., *R*(|***CPC***| $\times$ k_and)=2 at Line 3 of Algorithm 4), instead of checking all features within ***CPC***, ANDPC only needs to check E and N. Since $T \notin$ ***CPC***(E) and $T \notin$ ***CPC***(N) hold, E and N are deleted from ***andCPC*** (Lines 5-7 of Algorithm 4). Finally, as shown in Figure 1(d), we obtain ***andCPC***= $\{O, A\}$, i.e., ***PC*** = ***andCPC*** $\cup$ ***orPC*** = $\{O, A, B\}$ at Line 4 of Algorithm 1.
4) **Phase IV**: Based on the corrected PC of T obtained above, Phase IV finds the spouses of T by discovering the PC of each feature in ***PC***(T), and then identifies the spouses with regard to each feature. Specifically, since $D \perp\!\!\!\perp T|\emptyset$, $D \in$ ***PC***(O) and $T \not\perp\!\!\!\perp D|\{\emptyset \cup O\}$ hold, D is a spouse of T, i.e., ***SP*** = $\{D\}$. Finally, as shown in Figure 1(e), we obtain ***MB*** = ***PC*** $\cup$ ***SP*** = $\{A, B, C, D\}$.

Algorithm 4 *ANDPC*

Input: $k\_and \in$ [0,1]; ***CPC***
Output: ***andCPC***: corrected CPC by the "AND" rule
1: Initialize ***andCPC*** = ***CPC***
/*Ascending order, F_1 has the lowest dependency*/
2: $\langle F_1, ..., F_{|\boldsymbol{CPC}|}\rangle \longleftarrow$ ***CPC***
3: **for** i = 1 to R(|***CPC***| $\times$ k_and) **do**
4: [***or_rank2***, ***CPC2***] = *IdenCPC*(F_i)
5: **if** $C \notin$ ***CPC2*** **then**
6: ***andCPC*** = ***andCPC*** $\setminus \{F_i\}$
7: **end if**
8: **end for**

S-2: Detailed experimental results on Benchmark BN datasets

In this section, we present the detailed experimental results on all benchmark datasets as shown in Tables II and III. In these tables, "-" denotes that a method fails to generate any output with the corresponding dataset after running out of memory, and the best results are highlighted in bold face. Moreover, on a dataset, we only record the best results of F1 metric of DCMB and the corresponding Precision, Recall and Time metrics.

Datasets. We use 14 benchmark BNs with different numbers of variables in our experiments, and details of the 14 benchmark BNs are summarized in Table I [1]. Among them, Child3, Insurance3 and Alarm3 were generated by tiling 3 copies of the Child, Insurance, and Alarm networks, respectively [1]. Similarly, we also generated Child5, Child10, Insurance5, Insurance10, Alarm5 and Alarm10. For each benchmark BN network, we randomly generate three datasets including 500 data instances, 1,000 data instances and 5,000 data instances respectively.

Comparison Methods. We compare DCMB with 12 state-of-the-art causal feature selection algorithms, including I-

[1]Those benchmark BN networks are publicly available at http://www.bnlearn.com/bnrepository/

TABLE I
SUMMARY OF BENCHMARK BNS

Network	Num. Vars	Num. Edges	Max In/out-Degree	Min/Max |PCset|	Variable Domain
Child	20	25	2/7	1/8	2-6
Child3	60	79	3/7	1/8	2-6
Child5	100	126	2/7	1/8	2-6
Child10	200	257	2/7	1/8	2-6
Insurance	27	52	3/7	1/9	2-5
Insurance3	81	163	4/7	1/9	2-5
Insurance5	135	281	5/8	1/10	2-5
Insurance10	270	556	5/8	1/11	2-5
Alarm	37	46	4/5	1/6	2-4
Alarm3	111	149	4/5	1/6	2-4
Alarm5	185	265	4/6	1/8	2-4
Alarm10	370	570	4/7	1/9	2-4
Mildew	35	46	3/3	1/5	3-100
Barley	48	84	4/5	1/8	2-67

AMB [2], FBED^K [3], MMMB [4], PCMB [5], HITON-MB [6], MBOR [7], IPCMB [8], STMB [9], BAMB [10], CCMB [11], EEMB [12][2] and SRMB [13]. Note that PCMB and IPCMB use the "AND" rule while MBOR, CCMB and SRMB employ the "OR" rule.

Evaluation metrics. For benchmark BN networks, the MB of each feature can be read from those networks. Accordingly, in the experiments, we evaluate the algorithms using the following metrics.

- $Precision$. The precision metric denotes the number of true positives in the output (i.e., the features in the output of an algorithm belonging to the true MB of a given target in a test DAG) divided by the number of features in the output of the algorithm.
- $Recall$. The recall metric represents the number of true positives in the output divided by the number of true positives (the number of the true MB of a given target) in a test DAG.
- $F1$. $F1 = 2 * Precision * Recall/(Precision + Recall)$. The F1 score is the harmonic average of the precision and recall, where F1 = 1 is the best case (perfect precision and recall) while F1 = 0 is the worst case.
- $Time$. We report running time (in seconds) as the efficiency measure of different algorithms.

Implementation Details.

- All algorithms are implemented in C/C++. For the FBED^K algorithm, the value of K is set to 1, which is enough to make FBED^K converge.
- The conditional independence tests are G^2 tests with a statistical significance level of 0.01.
- For an algorithm, we identify the MBs of all features in each BN and report the average results of F1, Precision, Recall and Time.

From Tables II and III, we have the following conclusions:

- **F1 metric.** On most datasets, DCMB achieves the highest accuracy. Specially, on the Insurance benchmark BN dataset with 5000 samples, the F1 metric of DCMB is at least 3.5% higher than that of the other algorithms. For algorithms (e.g., PCMB and IPCMB) only adopting the "AND" rule, on the benchmark BN dataset with small-sized data samples (such as 500 and 1000 samples), their F1 metric is generally lower than other algorithms. This is because many CI tests will be unreliable when implementing MB learning methods on small sample datasets, leading to many true MB features being discarded. Continuing to use the "AND" rule to correct ***CPC*** will cause more true MB features being abandoned. For algorithms (e.g., MBOR, CCMB and SRMB) only using the "OR" rule, on the benchmark BN dataset with large-sized data samples (such as 5000 samples), their F1 metric values have not improved much compared with the other algorithms. The explanation for this is that, on datasets with large number of samples, reliable CI tests guarantee that almost all MB features are successfully discovered. In other words, the "OR" rule almost loses its effect on MB learning and even bring adverse effects due to non MB features being selected. The reason why the overall performance of STMB is poor is that it will add a lot of non MB features to MB of C as the spouses of C. On the Mildew and Barley benchmark BN datasets, since the value range of each variable is large, the CI tests will become unreliable even on a datasets with 5000 samples [14], which seriously deteriorates the performance of all algorithms. In addition, on all datasets, we observe that SRMB achieves a comparable performance against CCMB.
- **Precision and Recall metrics.** On the whole, the precision metric of algorithms employing the "AND" rule is higher than that of other algorithms, especially on datasets with large number of samples, while the recall metric of algorithms adopting the "OR" rule is higher than that of other algorithms, especially on datasets with small number of samples. Since DCMB uses both the "AND" and "OR" rules, meanwhile, its selective strategy prevents true MB features from being deleted and non MB features from being selected, the precision and recall metrics of DCMB are always high on all datasets. SRMB and CCMB are all designed to recover false negatives. However, the precision value of SRMB is always higher than that of CCMB, and the recall value of SRMB is always lower than that of CCMB, since CCMB tends to obtain more features than SRMB, even if these features are not the true MB features.
- **Time metric.** FBED^K is the fastest algorithm among all MB learning algorithms under comparison. Algorithms using the "AND" rule are slightly slower than algorithms without using any of the two rules. In contrast, algorithms without using any of the two rules are significantly faster than algorithms using the "OR" rule, since $(|\boldsymbol{F}| - |\boldsymbol{PC}|) \gg |\boldsymbol{PC}|$ holds on most datasets. Particularly, on the Mildew and Barley benchmark BN datasets, $(|\boldsymbol{F}| - |\boldsymbol{PC}|) < |\boldsymbol{PC}|$ holds, and thus algorithms adopting the "OR" rule are faster than algorithms employing the "AND" rule on these two datasets. Especially, although MBOR uses the "OR" rule, it is not slow, since it utilizes a fast but data inefficient algorithm to correct the MB features. As a

[2]The source codes are available at https://github.com/kuiy/CausalFS

divide-and-conquer MB learning method, EEMB shows high efficiency on all datasets. On most datasets, the time cost of DCMB is much lower than that of algorithms adopting the "OR" rule and slightly higher than that of algorithms employing the "AND" rule, since DCMB adopts the dual selective correction strategy.

S-3: Experimental results of the single correction strategy

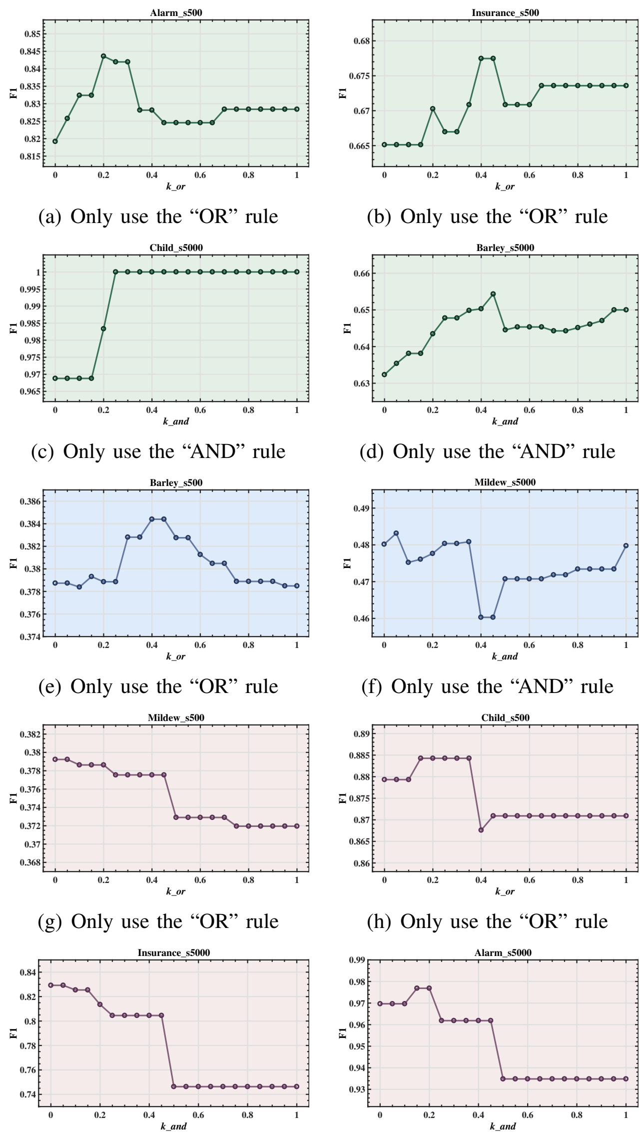
In this section, we validate the effectiveness of the single correction strategy using either the "AND" rule or the "OR" rule, respectively.

Using DCMB (Algorithm 1), Fig. 2 shows the experimental results on the benchmark datasets (i.e., Child with 500 and 5000 samples, Insurance with 500 and 5000 samples, Alarm with 500 and 5000 samples, Mildew with 500 and 5000 samples, Barley with 500 and 5000 samples). Specifically, we set k_and of DCMB to 0 (i.e., the "AND" rule does not work) while traversing k_or of DCMB from 0 to 1, and we record the change process of F1 metric as shown in Figs. 2(a), (b), (e), (g) and (h). In the same way, we set k_or of DCMB to 0 (i.e., the "OR" rule does not work) while traversing k_and of DCMB from 0 to 1, and the experimental results are shown in Figs. 2(c), (d), (f), (i) and (j).

Through the observation of the experimental results in Fig. 2, we have the following interesting findings.

1) In Figs. 2(a), (b), (c) and (d), $\mathrm{F1}_{k\_or=1}$ (the value of F1 metric when $k\_or = 1$) is greater than $\mathrm{F1}_{k\_or=0}$ and $\mathrm{F1}_{k\_and=1}$ is higher than $\mathrm{F1}_{k\_and=0}$. However, if the selective correction strategy is adopted, we can get higher MB discovery accuracy as shown in Figs. 2(a), (b) and (d). For Fig. 2(c), when $k \geqslant 0.25$, F1 metric no longer changes, and we do not need to utilize the "AND" rule to correct all variables within ***CPC*** (candidate PC).
2) As shown in Figs. 2(e) and (f), the values of $\mathrm{F1}_{k\_or=1}$ and $\mathrm{F1}_{k\_or=0}$ are approximate, and $\mathrm{F1}_{k\_and=1}$ is almost equal $\mathrm{F1}_{k\_and=0}$. But in Fig. 2(e), when $k\_or = 0.4$, F1 metric reaches a peak. Similarly, in Fig. 2(f), we find that MB learning is more accurate when $k\_and = 0.05$.
3) From Figs. 2(g), (h), (i) and (j), we can see that $\mathrm{F1}_{k\_or=1}$<$\mathrm{F1}_{k\_or=0}$ and $\mathrm{F1}_{k\_and=1}$<$\mathrm{F1}_{k\_and=0}$. This suggests that the "OR" rule or the "AND" rule not only fails to correct the false positive and false negative errors but also hurts the performance of algorithms employing two rules. However, in Figs. 2(h) and (j), F1 metric starts with an increase and then goes down. In other words, if the selective correction strategy is not employed, the optimal solution of MB learning will be masked. In Figs. 2(g) and (i), F1 metric has been declining, which means the "OR" rule or the "AND" rule only brings adverse effects to the algorithms adopting two rules. Thus, on the Mildew with 500 samples and Insurance with 5000 samples, k_or and k_and of DCMB should be set to 0, respectively.

Based on the findings discussed above, we conclude that our proposed correction strategy not only can significantly improve the accuracy of MB discovery but also is less computational expensive than the algorithms employing the "OR" rule or the "AND" rule.



(a) Only use the "OR" rule (b) Only use the "OR" rule
(c) Only use the "AND" rule (d) Only use the "AND" rule
(e) Only use the "OR" rule (f) Only use the "AND" rule
(g) Only use the "OR" rule (h) Only use the "OR" rule
(i) Only use the "AND" rule (j) Only use the "AND" rule

Fig. 2. Experimental results on benchmark datasets for validating the effectiveness using the single selective-correction strategy.

S-4: Statistical tests for verifying whether SA-DCMB is significantly better than other methods

In this section, we adopt the Friedman test and Nemenyi test [15] to further compare the performance of SA-DCMB with that of its rivals.

We first perform the Friedman test at the 0.05 significance level under the null-hypothesis which states that the performance of all algorithms is the same on all datasets (i.e., the average ranks of all algorithms are equivalent). The average ranks of SA-DCMB and its rivals when using different classifiers are summarized in Table IV. Since the IPCMB, STMB, BAMB, CCMB, EEMB, SRMB and QPFS algorithms cannot produce any output on some datasets, we do not record their average ranks in this table. From Table IV, we can see that the null hypothesis is rejected on these two classifiers. We also note that SA-DCMB performs better than its rivals (the lower rank value is better).

TABLE II
COMPARISON OF DCMB WITH OTHER MB METHODS ON BENCHMARK BN DATASETS (1)

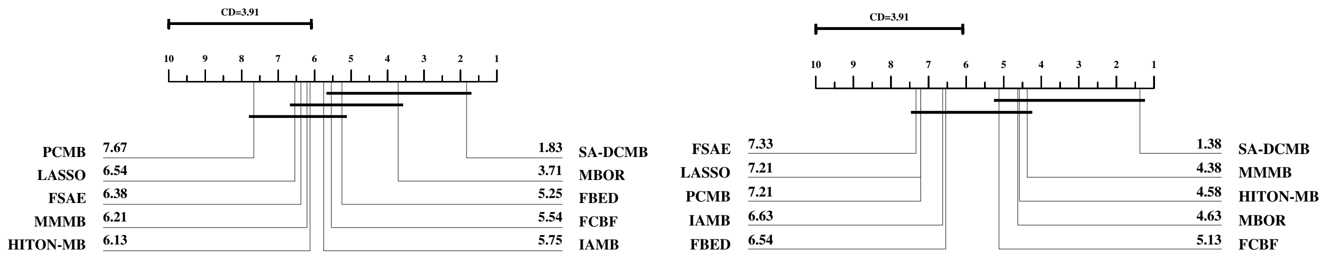
	#Sample	500				1,000				5,000			
Dataset	Algorithm	F1(↑)	Precision(↑)	Recall(↑)	Time(↓)	F1(↑)	Precision(↑)	Recall(↑)	Time(↓)	F1(↑)	Precision(↑)	Recall(↑)	Time(↓)
	IAMB	0.746	**0.887**	0.673	0.001	0.817	0.892	0.781	0.002	0.922	0.941	0.927	0.008
	FBED	0.740	**0.887**	0.666	**0.000**	0.820	0.892	0.784	**0.001**	0.927	0.953	0.924	**0.005**
	MMMB	0.772	0.865	0.735	0.001	0.895	0.965	0.858	0.003	0.975	0.978	0.977	0.016
	PCMB	0.686	0.818	0.642	0.004	0.837	0.935	0.787	0.009	0.974	**1.000**	0.956	0.062
	HITON-MB	0.771	0.856	0.735	0.001	0.897	0.970	0.857	0.003	0.973	0.972	0.980	0.019
	MBOR	0.797	0.886	0.751	0.001	0.891	0.943	0.870	0.003	0.975	0.983	0.973	0.022
Alarm	IPCMB	0.675	0.820	0.622	0.002	0.836	0.934	0.785	0.005	0.979	**1.000**	0.964	0.041
	STMB	0.605	0.616	0.718	0.001	0.707	0.693	0.841	0.002	0.795	0.764	0.961	0.016
	BAMB	0.757	0.865	0.709	0.001	0.864	0.942	0.820	0.002	0.955	0.974	0.948	0.016
	CCMB	0.804	0.853	**0.796**	0.012	0.911	0.951	**0.898**	0.027	0.967	0.961	**0.984**	0.149
	EEMB	0.760	0.856	0.716	0.001	0.869	0.947	0.826	0.002	0.960	0.991	0.943	0.012
	SRMB	0.801	0.861	0.786	0.013	0.908	0.958	0.887	0.029	0.963	0.966	0.977	0.157
	DCMB	**0.812**	0.870	**0.796**	0.007	**0.916**	**0.974**	0.888	0.018	**0.986**	0.991	**0.984**	0.081
	IAMB	0.654	0.787	0.620	0.002	0.701	0.774	0.697	0.005	0.798	0.798	0.857	0.029
	FBED	0.665	0.803	0.622	**0.001**	0.709	0.797	0.691	**0.002**	0.841	0.869	0.852	**0.012**
	MMMB	0.739	0.877	0.682	0.002	0.785	0.895	0.742	0.004	0.889	0.918	0.882	0.027
	PCMB	0.689	0.863	0.620	0.007	0.766	0.923	0.695	0.015	0.893	0.953	0.856	0.104
	HITON-MB	0.742	0.880	0.685	0.002	0.788	0.895	0.748	0.005	0.883	0.909	0.882	0.031
	MBOR	0.752	0.868	0.705	0.003	0.797	0.879	0.764	0.006	0.889	0.919	0.877	0.048
Alarm3	IPCMB	0.673	0.856	0.601	0.003	0.786	**0.932**	0.719	0.006	0.889	0.943	0.859	0.053
	STMB	0.551	0.576	0.654	0.002	0.583	0.554	0.755	0.005	0.641	0.569	0.883	0.030
	BAMB	0.726	**0.884**	0.663	0.002	0.774	0.905	0.719	0.005	0.872	0.918	0.850	0.028
	CCMB	0.768	0.853	**0.745**	0.067	0.794	0.852	**0.789**	0.143	0.866	0.865	**0.897**	0.784
	EEMB	0.722	0.879	0.659	0.002	0.775	0.903	0.722	0.004	0.882	0.936	0.850	0.024
	SRMB	0.766	0.859	0.739	0.069	0.797	0.861	0.786	0.148	0.868	0.871	0.894	0.813
	DCMB	**0.773**	0.863	0.742	0.025	**0.809**	0.930	0.748	0.041	**0.910**	**0.968**	0.876	0.199
	IAMB	0.634	0.742	0.623	0.003	0.685	0.757	0.701	0.008	0.697	0.675	0.823	0.059
	FBED	0.657	0.789	0.627	**0.002**	0.697	0.793	0.687	**0.004**	0.777	0.791	0.820	**0.021**
	MMMB	0.712	0.846	0.674	0.003	0.773	0.886	0.731	0.007	0.875	0.933	0.858	0.040
	PCMB	0.673	0.857	0.608	0.011	0.734	0.903	0.663	0.021	0.872	0.964	0.825	0.152
	HITON-MB	0.711	0.844	0.674	0.003	0.772	0.885	0.730	0.007	0.874	0.931	0.858	0.044
	MBOR	**0.731**	0.852	0.690	0.004	0.787	0.879	0.751	0.009	0.879	0.941	0.852	0.069
Alarm5	IPCMB	0.664	0.851	0.599	0.004	0.732	0.899	0.663	0.008	0.847	0.946	0.802	0.063
	STMB	0.495	0.509	0.654	0.003	0.550	0.531	0.723	0.007	0.578	0.507	0.856	0.044
	BAMB	0.700	0.867	0.644	0.004	0.760	0.898	0.703	0.007	0.872	0.947	0.840	0.044
	CCMB	0.729	0.808	**0.725**	0.178	0.794	0.852	**0.786**	0.356	0.860	0.887	**0.879**	1.975
	EEMB	0.705	**0.869**	0.652	0.003	0.764	0.900	0.709	0.007	0.870	0.953	0.829	0.039
	SRMB	0.726	0.814	0.716	0.181	0.793	0.856	0.782	0.364	0.863	0.895	0.876	2.034
	DCMB	0.730	0.820	0.720	0.054	**0.806**	**0.910**	0.768	0.096	**0.892**	**0.981**	0.849	0.386
	IAMB	0.545	0.632	0.559	0.008	0.600	0.637	0.657	0.019	0.630	0.588	0.790	0.249
	FBED	0.580	0.697	0.568	**0.004**	0.645	0.713	0.659	**0.008**	0.731	0.735	0.793	**0.083**
	MMMB	0.667	0.817	0.623	0.006	0.756	0.878	0.710	0.013	0.842	0.905	0.823	0.154
	PCMB	0.634	0.843	0.563	0.022	0.728	0.903	0.652	0.048	0.847	**0.967**	0.786	0.646
	HITON-MB	0.664	0.809	0.624	0.007	0.757	0.878	0.712	0.014	0.839	0.901	0.822	0.174
	MBOR	**0.690**	**0.855**	0.629	0.008	0.766	0.885	0.717	0.019	0.852	0.917	0.828	0.268
Alarm10	IPCMB	0.616	0.832	0.543	0.007	0.729	0.905	0.651	0.016	0.830	0.954	0.768	0.235
	STMB	0.385	0.361	0.597	0.006	0.443	0.380	0.708	0.013	0.487	0.399	0.836	0.180
	BAMB	0.647	0.826	0.591	0.007	0.737	0.890	0.675	0.015	0.838	0.916	0.808	0.189
	CCMB	0.684	0.773	**0.673**	0.708	0.760	0.835	**0.745**	1.406	0.825	0.845	**0.850**	16.420
	EEMB	0.652	0.831	0.598	0.007	0.742	0.892	0.682	0.014	0.834	0.917	0.799	0.187
	SRMB	0.683	0.779	0.667	0.716	0.758	0.841	0.737	1.425	0.829	0.857	0.846	16.671
	DCMB	0.688	0.850	0.634	0.142	**0.773**	**0.907**	0.721	0.208	**0.855**	0.955	0.805	2.480
	IAMB	0.821	0.892	0.804	**0.000**	0.836	0.863	0.872	**0.001**	0.867	0.837	0.940	0.004
	FBED	0.830	0.913	0.804	**0.000**	0.836	0.863	0.872	**0.001**	0.894	0.877	0.940	**0.003**
	MMMB	0.879	0.971	0.830	0.001	0.860	0.898	0.881	0.003	**1.000**	**1.000**	**1.000**	0.022
	PCMB	0.776	0.931	0.706	0.004	0.827	0.933	0.783	0.010	**1.000**	**1.000**	**1.000**	0.097
	HITON-MB	0.866	**0.981**	0.810	0.001	0.852	0.888	0.875	0.003	**1.000**	**1.000**	**1.000**	0.026
	MBOR	0.879	0.971	0.829	0.001	0.839	0.852	0.863	0.002	0.963	0.961	0.975	0.018
Child	IPCMB	0.793	0.931	0.731	0.002	0.827	0.921	0.793	0.005	**1.000**	**1.000**	**1.000**	0.042
	STMB	0.867	0.874	**0.885**	0.001	0.828	0.851	0.853	0.002	0.876	0.823	0.988	0.011
	BAMB	0.881	**0.981**	0.833	0.001	0.875	0.925	0.885	0.002	0.988	0.992	0.988	0.017
	CCMB	0.881	0.949	0.860	0.005	0.852	0.838	**0.922**	0.011	**1.000**	**1.000**	**1.000**	0.081
	EEMB	0.857	0.955	0.810	0.001	0.859	0.903	0.875	**0.001**	0.976	0.971	0.988	0.010
	SRMB	0.882	0.952	0.859	0.005	0.854	0.845	0.919	0.012	**1.000**	**1.000**	**1.000**	0.085
	DCMB	**0.887**	**0.981**	0.837	0.004	**0.893**	**0.958**	0.868	0.009	**1.000**	**1.000**	**1.000**	0.059
	IAMB	0.698	0.749	0.737	**0.001**	0.696	0.695	0.803	0.003	0.756	0.704	0.920	0.017
	FBED	0.707	0.763	0.729	**0.001**	0.732	0.748	0.798	**0.001**	0.849	0.822	0.927	**0.007**
	MMMB	0.805	0.885	0.780	0.002	0.863	0.949	0.837	0.004	0.944	0.948	0.963	0.027
	PCMB	0.732	0.872	0.676	0.006	0.816	**0.951**	0.758	0.015	0.956	**0.981**	0.950	0.114
	HITON-MB	0.797	0.882	0.770	0.002	0.867	0.947	0.844	0.005	0.944	0.948	0.963	0.033
	MBOR	0.820	**0.916**	0.772	0.002	0.855	0.919	0.850	0.004	0.957	0.972	0.956	0.033
Child3	IPCMB	0.728	0.825	0.686	0.003	0.793	0.906	0.745	0.008	0.956	**0.981**	0.950	0.053
	STMB	0.652	0.616	0.774	**0.001**	0.699	0.680	0.831	0.003	0.788	0.710	**0.973**	0.022
	BAMB	0.833	0.913	0.801	0.002	0.863	0.931	0.852	0.003	0.945	0.953	0.957	0.023
	CCMB	0.825	0.850	**0.843**	0.025	0.870	0.913	**0.880**	0.055	0.934	0.930	0.967	0.340
	EEMB	0.815	0.904	0.780	**0.001**	0.860	0.926	0.853	0.003	0.953	0.963	0.963	0.017
	SRMB	0.828	0.861	0.839	0.026	0.872	0.919	0.878	0.058	0.935	0.934	0.965	0.357
	DCMB	**0.845**	0.899	0.829	0.009	**0.880**	0.934	0.877	0.024	**0.960**	0.979	0.959	0.133
	IAMB	0.617	0.648	0.711	0.002	0.657	0.627	0.857	0.005	0.674	0.590	0.948	0.033
	FBED	0.660	0.706	0.710	**0.001**	0.728	0.708	0.848	**0.002**	0.784	0.711	0.951	**0.011**
	MMMB	0.826	0.902	0.803	0.002	0.880	0.910	0.887	0.005	0.949	0.928	0.992	0.036
	PCMB	0.757	0.851	0.722	0.009	0.823	0.915	0.777	0.019	0.980	**0.985**	0.982	0.151
	HITON-MB	0.811	0.899	0.783	0.003	0.883	0.912	0.892	0.006	0.949	0.932	0.990	0.043
	MBOR	0.826	**0.927**	0.786	0.003	0.881	0.902	0.893	0.006	0.959	0.951	0.983	0.042
Child5	IPCMB	0.755	0.851	0.714	0.003	0.821	0.901	0.778	0.009	0.977	0.980	0.982	0.062
	STMB	0.619	0.557	0.808	0.002	0.671	0.583	0.894	0.005	0.747	0.644	0.987	0.030
	BAMB	0.823	0.894	0.806	0.002	0.879	0.919	0.882	0.005	0.945	0.924	0.988	0.032
	CCMB	0.834	0.880	**0.844**	0.060	0.882	0.871	**0.930**	0.130	0.928	0.892	**1.000**	0.764
	EEMB	0.812	0.889	0.790	0.002	0.891	0.920	0.903	0.004	0.947	0.922	0.994	0.025
	SRMB	0.837	0.889	0.841	0.062	0.883	0.875	0.928	0.135	0.928	0.893	**1.000**	0.797
	DCMB	**0.845**	0.915	0.834	0.015	**0.903**	**0.971**	0.871	0.056	**0.981**	0.980	0.990	0.295
	IAMB	0.544	0.528	0.724	0.005	0.580	0.527	0.841	0.012	0.575	0.472	0.941	0.081
	FBED	0.603	0.604	0.713	**0.002**	0.691	0.648	0.846	**0.004**	0.709	0.607	0.943	**0.023**
	MMMB	0.796	0.851	0.791	0.004	0.878	0.927	0.872	0.008	0.944	0.916	0.994	0.055
	PCMB	0.748	0.861	0.703	0.016	0.811	0.922	0.762	0.031	0.970	0.962	0.986	0.237
	HITON-MB	0.787	0.854	0.778	0.005	0.879	0.925	0.874	0.009	0.943	0.915	0.992	0.060
	MBOR	0.797	0.898	0.758	0.005	0.882	0.936	0.864	0.011	0.942	0.920	0.985	0.067
Child10	IPCMB	0.742	0.846	0.699	0.005	0.805	0.916	0.752	0.012	**0.979**	**0.976**	0.986	0.081
	STMB	0.487	0.395	0.798	0.004	0.510	0.391	0.882	0.008	0.601	0.469	0.991	0.052
	BAMB	0.805	0.873	0.796	0.005	0.891	**0.941**	0.882	0.009	0.945	0.924	0.986	0.054
	CCMB	0.792	0.798	**0.839**	0.220	0.888	0.892	**0.926**	0.455	0.924	0.885	**0.995**	2.499
	EEMB	0.795	0.874	0.778	0.004	0.885	0.925	0.885	0.008	0.951	0.929	0.991	0.046
	SRMB	0.794	0.805	0.836	0.225	0.889	0.897	0.923	0.467	0.925	0.890	0.992	2.580
	DCMB	**0.810**	**0.923**	0.768	0.046	**0.901**	0.934	0.901	0.126	0.975	0.966	0.991	0.807

TABLE III
COMPARISON OF DCMB WITH OTHER MB METHODS ON BENCHMARK BN DATASETS (2)

	#Sample	500				1,000				5,000			
Dataset	Algorithm	F1(↑)	Precision(↑)	Recall(↑)	Time(↓)	F1(↑)	Precision(↑)	Recall(↑)	Time(↓)	F1(↑)	Precision(↑)	Recall(↑)	Time(↓)
Insurance	IAMB	0.582	0.818	0.499	0.001	0.646	0.877	0.559	**0.001**	0.784	0.909	0.728	0.006
	FBED	0.580	0.809	0.499	**0.000**	0.651	0.883	0.559	**0.001**	0.773	0.909	0.704	**0.004**
	MMMB	0.638	0.851	0.551	0.001	0.708	0.881	0.628	0.003	0.817	0.915	0.764	0.033
	PCMB	0.530	0.837	0.412	0.004	0.661	0.869	0.563	0.012	0.762	0.901	0.685	0.148
	HITON-MB	0.643	0.865	0.557	0.001	0.704	0.881	0.623	0.003	0.801	0.897	0.751	0.035
	MBOR	0.652	0.837	0.585	0.001	0.709	0.885	0.626	0.004	0.803	0.891	0.767	0.046
	IPCMB	0.524	0.837	0.407	0.002	0.661	0.869	0.563	0.005	0.738	0.875	0.666	0.062
	STMB	0.551	0.724	0.493	0.001	0.595	0.651	0.595	0.002	0.726	0.753	**0.801**	0.021
	BAMB	0.637	0.840	0.556	0.001	0.690	0.880	0.599	0.003	0.799	0.915	0.740	0.027
	CCMB	0.643	0.790	**0.602**	0.008	0.707	0.827	**0.649**	0.019	0.800	0.839	**0.801**	0.162
	EEMB	0.647	0.856	0.563	0.001	0.699	**0.894**	0.614	0.002	0.796	0.931	0.723	0.015
	SRMB	0.642	0.793	0.597	0.009	0.707	0.829	0.646	0.021	0.801	0.844	0.798	0.169
	DCMB	**0.653**	**0.875**	0.557	0.006	**0.718**	0.893	0.635	0.014	**0.852**	**0.943**	**0.801**	0.133
Insurance3	IAMB	0.631	0.842	0.556	**0.001**	0.679	0.834	0.621	0.003	0.729	0.794	0.739	0.022
	FBED	0.636	0.851	0.554	**0.001**	0.697	0.852	0.624	**0.002**	0.737	0.818	0.713	**0.010**
	MMMB	0.685	0.818	0.640	0.003	0.764	0.876	0.719	0.007	0.820	0.896	0.810	0.065
	PCMB	0.679	**0.877**	0.597	0.011	0.762	**0.937**	0.676	0.031	0.817	0.928	0.762	0.348
	HITON-MB	0.679	0.804	0.640	0.003	0.767	0.884	0.716	0.008	0.820	0.895	0.810	0.075
	MBOR	0.704	0.871	0.638	0.004	0.776	0.907	0.709	0.009	0.825	0.904	0.803	0.106
	IPCMB	0.642	0.858	0.565	0.004	0.749	0.911	0.672	0.011	0.802	0.889	0.768	0.148
	STMB	0.560	0.535	**0.715**	0.003	0.586	0.530	**0.778**	0.006	0.554	0.476	0.821	0.054
	BAMB	0.705	0.865	0.653	0.003	0.776	0.914	0.708	0.007	0.819	0.892	0.808	0.067
	CCMB	0.690	0.762	0.689	0.048	0.754	0.809	0.750	0.108	0.806	0.818	**0.853**	0.848
	EEMB	0.698	0.855	0.647	0.003	0.769	0.897	0.706	0.006	0.816	0.905	0.789	0.043
	SRMB	0.695	0.780	0.681	0.050	0.757	0.819	0.746	0.111	0.805	0.821	0.848	0.871
	DCMB	**0.710**	0.856	0.648	0.017	**0.782**	0.884	0.736	0.037	**0.849**	**0.932**	0.823	0.578
Insurance5	IAMB	0.574	0.760	0.522	0.003	0.616	0.741	0.600	0.006	0.688	0.755	0.727	0.040
	FBED	0.598	0.799	0.529	**0.001**	0.632	0.764	0.598	**0.003**	0.717	0.808	0.705	**0.015**
	MMMB	0.671	0.845	0.610	0.004	0.738	0.874	0.687	0.009	0.811	0.899	0.790	0.081
	PCMB	0.646	**0.898**	0.546	0.015	0.726	**0.927**	0.637	0.040	0.816	0.934	0.757	0.419
	HITON-MB	0.672	0.846	0.613	0.004	0.736	0.871	0.689	0.010	0.816	0.909	0.789	0.093
	MBOR	0.659	0.881	0.572	0.005	0.737	0.889	0.676	0.013	0.814	0.892	0.788	0.138
	IPCMB	0.630	0.876	0.536	0.005	0.706	0.883	0.630	0.014	0.803	0.908	0.757	0.160
	STMB	0.527	0.483	**0.690**	0.004	0.514	0.431	**0.755**	0.009	0.557	0.497	0.820	0.075
	BAMB	0.686	0.878	0.614	0.004	0.738	0.898	0.677	0.010	0.818	0.903	0.791	0.085
	CCMB	0.674	0.784	0.647	0.111	0.734	0.801	0.729	0.253	0.803	0.838	**0.826**	1.835
	EEMB	0.682	0.872	0.611	0.004	0.728	0.891	0.664	0.008	0.805	0.898	0.775	0.058
	SRMB	0.677	0.793	0.643	0.114	0.735	0.806	0.726	0.259	0.803	0.840	0.825	1.875
	DCMB	**0.689**	0.868	0.618	0.026	**0.764**	0.888	0.716	0.063	**0.838**	**0.943**	0.795	0.652
Insurance10	IAMB	0.546	0.687	0.538	0.005	0.578	0.684	0.595	0.012	0.637	0.672	0.728	0.090
	FBED	0.575	0.746	0.536	**0.002**	0.617	0.738	0.599	**0.005**	0.681	0.730	0.715	**0.031**
	MMMB	0.672	0.806	0.635	0.006	0.730	0.851	0.689	0.014	0.805	0.886	0.793	0.105
	PCMB	0.670	**0.896**	0.579	0.025	0.726	**0.922**	0.643	0.064	0.803	0.940	0.740	0.550
	HITON-MB	0.674	0.809	0.637	0.007	0.728	0.848	0.688	0.015	0.806	0.893	0.793	0.122
	MBOR	0.676	0.879	0.601	0.008	0.738	0.874	0.684	0.021	0.803	0.876	0.788	0.188
	IPCMB	0.645	0.882	0.556	0.007	0.706	0.887	0.632	0.020	0.791	0.896	0.746	0.175
	STMB	0.418	0.340	**0.673**	0.007	0.419	0.327	**0.737**	0.016	0.438	0.344	0.820	0.110
	BAMB	0.680	0.836	0.630	0.008	0.737	0.889	0.681	0.018	0.809	0.895	0.788	0.123
	CCMB	0.666	0.744	0.665	0.391	0.721	0.786	0.719	0.854	0.793	0.812	**0.836**	5.376
	EEMB	0.679	0.839	0.628	0.007	0.725	0.879	0.667	0.015	0.805	0.898	0.776	0.099
	SRMB	0.670	0.761	0.658	0.397	0.723	0.792	0.716	0.867	0.792	0.816	0.831	5.567
	DCMB	**0.686**	0.842	0.631	0.051	**0.745**	0.896	0.684	0.108	**0.830**	**0.947**	0.781	0.814
Mildew	IAMB	0.289	0.600	0.199	**0.000**	0.338	0.624	0.251	0.001	**0.529**	**0.688**	0.463	0.004
	FBED	0.289	0.600	0.199	**0.000**	0.340	0.633	0.251	**0.000**	0.474	0.657	0.387	**0.002**
	MMMB	0.344	0.496	0.324	0.001	0.384	0.408	0.446	0.002	0.455	0.392	0.711	0.025
	PCMB	0.342	0.500	0.310	0.004	0.385	0.446	0.423	0.019	0.466	0.448	0.651	0.260
	HITON-MB	0.156	0.171	0.171	0.001	0.299	0.292	0.376	0.002	0.457	0.380	**0.775**	0.032
	MBOR	0.331	**0.622**	0.247	0.001	**0.420**	**0.639**	0.338	0.004	-	-	-	-
	IPCMB	0.268	0.202	0.532	0.007	0.357	0.322	0.617	42.319	-	-	-	-
	STMB	0.266	0.196	**0.548**	**0.000**	0.337	0.280	**0.641**	1.540	-	-	-	-
	BAMB	0.162	0.173	0.183	**0.000**	0.319	0.299	0.431	0.001	-	-	-	-
	CCMB	0.346	0.492	0.333	0.004	0.386	0.391	0.466	0.011	0.445	0.364	0.754	0.096
	EEMB	0.160	0.174	0.180	**0.000**	0.317	0.298	0.421	0.001	-	-	-	-
	SRMB	0.342	0.497	0.320	0.004	0.388	0.401	0.459	0.012	0.445	0.380	0.740	0.102
	DCMB	**0.352**	0.500	0.333	0.004	0.398	0.449	0.443	0.018	0.482	0.445	0.702	0.233
Barley	IAMB	0.284	**0.667**	0.192	**0.000**	0.326	0.615	0.237	0.001	0.489	**0.736**	0.402	0.005
	FBED	0.284	**0.667**	0.192	**0.000**	0.334	**0.625**	0.244	**0.000**	0.492	**0.736**	0.406	**0.004**
	MMMB	0.379	0.335	0.558	0.003	0.414	0.340	0.645	0.006	0.630	0.581	0.787	0.050
	PCMB	0.379	0.339	0.544	0.038	0.409	0.362	0.611	0.086	0.629	0.620	0.748	0.423
	HITON-MB	0.344	0.303	0.522	0.003	0.384	0.311	0.620	0.008	0.632	0.583	0.785	0.071
	MBOR	0.343	0.618	0.271	0.002	0.421	0.565	0.377	0.004	-	-	-	-
	IPCMB	0.363	0.306	0.565	0.008	0.398	0.336	0.602	0.015	0.634	0.624	0.744	0.117
	STMB	0.353	0.285	**0.581**	0.001	0.368	0.285	0.638	0.002	-	-	-	-
	BAMB	0.342	0.300	0.522	0.001	0.389	0.315	0.623	0.005	-	-	-	-
	CCMB	0.378	0.328	0.569	0.011	0.411	0.334	**0.672**	0.027	0.624	0.548	**0.814**	0.288
	EEMB	0.343	0.301	0.520	0.001	0.386	0.314	0.612	0.004	-	-	-	-
	SRMB	0.380	0.334	0.567	0.011	0.411	0.340	0.667	0.029	0.623	0.551	0.810	0.294
	DCMB	**0.389**	0.343	0.567	0.029	**0.423**	0.347	0.661	0.075	**0.645**	0.592	0.794	0.209

TABLE IV
THE AVERAGE RANKS OF SA-DCMB AND ITS RIVALS USING NB AND KNN CLASSIFIERS.

Algorithm		IAMB	FBED	MMMB	PCMB	HITON-MB	MBOR	FCBF	LASSO	FSAE	SA-DCMB
Avg rank	NB	5.75	5.25	6.21	7.67	6.13	3.71	5.54	6.54	6.38	**1.83**
	KNN	6.63	6.54	4.38	7.21	4.58	4.63	5.13	7.21	7.33	**1.38**



(a) NB Classifier (b) KNN Classifier

Fig. 3. Crucial difference diagram of the Nemenyi test for NB and KNN classifier on 12 real-world datasets (Since IPCMB, STMB, BAMB, CCMB, EEMB, SRMB and QPFS fail to generate any output on some datasets, their results are not shown in the crucial difference diagram.)

To further analyze the significant difference between SA-DCMB and its rivals, we perform the Nemenyi test, which states that the performance levels of two algorithms are significantly different if the corresponding average ranks differ by at least one critical difference (CD). The CD for the Nemenyi test is calculated as follows (i.e., Eq. (1)).

$$\mathrm{CD} = q_{\alpha,m}\sqrt{\frac{m(m+1)}{6|\mathcal{D}|}} \tag{1}$$

where α is the significance level, $|m|$ is the number of comparison algorithms, and $|\mathcal{D}|$ denotes the number of real-world datasets. In our experiments, $m = 10$, $q_{\alpha=0.05,m=10} = 3.164$ at significance level $\alpha = 0.05$. Whether using NB or KNN classifiers, $|\mathcal{D}| = 12$, and thus CD=3.91.

Figs. 3(a) and (b) provide the CD diagrams, where the average rank of each algorithm is marked along the axis (lower ranks to the right). Using NB classifier, we observe that SA-DCMB achieves a comparable performance against MBOR, FBED^K and FCBF, and SA-DCMB significantly outperforms the other algorithms. Using KNN classifier, we note that SA-DCMB significantly outperforms IAMB, FBED^K, PCMB, LASSO and FSAE, and SA-DCMB achieves a comparable performance against the other algorithms. SA-DCMB is the only algorithm that achieves the lowest rank value whether using NB or KNN classifiers.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (under grant 2020AAA0106100), the National Natural Science Foundation of China (under Grant 61876206), and the Australian Research Council Discovery Projects (under grant DP200101210).

References

[1] A. Statnikov, I. Tsamardinos, and C. Aliferis, "An algorithm for generation of large bayesian networks," *Department of Biomedical Informatics, Discovery Systems Laboratory, Vanderbilt University, Tech. Rep. Technical Report DSL-03-01*, 2003.

[2] I. Tsamardinos and C. F. Aliferis, "Towards principled feature selection: Relevancy, filters and wrappers," in *Proceedings of International Workshop on Artificial Intelligence and Statistics*. Morgan Kaufmann Publishers: Key West, FL, USA, 2003.

[3] G. Borboudakis and I. Tsamardinos, "Forward-backward selection with early dropping," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 276–314, 2019.

[4] I. Tsamardinos, C. F. Aliferis, and A. Statnikov, "Time and sample efficient discovery of Markov blankets and direct causal relations," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2003, pp. 673–678.

[5] J. M. Pena, R. Nilsson, J. Björkegren, and J. Tegnér, "Towards scalable and data efficient learning of Markov boundaries," *International Journal of Approximate Reasoning*, vol. 45, no. 2, pp. 211–232, 2007.

[6] C. F. Aliferis, I. Tsamardinos, and A. Statnikov, "HITON: a novel Markov Blanket algorithm for optimal variable selection," in *American Medical Informatics Association Annual Symposium*, vol. 2003. American Medical Informatics Association, 2003, p. 21.

[7] S. R. De Morais and A. Aussem, "A novel scalable and data efficient feature subset selection algorithm," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 298–312.

[8] S. Fu and M. C. Desmarais, "Fast Markov blanket discovery algorithm via local learning within single pass," in *Proceedings of Conference of the Canadian Society for Computational Studies of Intelligence*, vol. 5032. Springer, 2008, pp. 96–107.

[9] T. Gao and Q. Ji, "Efficient Markov blanket discovery and its application," *IEEE Transactions on Cybernetics*, vol. 47, no. 5, pp. 1169–1179, 2017.

[10] Z. Ling, K. Yu, H. Wang, L. Liu, W. Ding, and X. Wu, "BAMB: A balanced Markov blanket discovery approach to feature selection," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 5, pp. 1–25, 2019.

[11] X. Wu, B. Jiang, K. Yu, H. Chen *et al.*, "Accurate Markov boundary discovery for causal feature selection," *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 4983–4996, 2019.

[12] H. Wang, Z. Ling, K. Yu, and X. Wu, "Towards efficient and effective discovery of Markov blankets for feature selection," *Information Sciences*, vol. 509, pp. 227–242, 2020.

[13] X. Wu, B. Jiang, K. Yu, and H. Chen, "Separation and recovery Markov boundary discovery and its application in EEG-based emotion recognition," *Information Sciences*, vol. 571, pp. 262–278, 2021.

[14] S. Yaramakala and D. Margaritis, "Speculative Markov blanket discovery for optimal feature selection," in *IEEE International Conference on Data Mining*. IEEE, 2005, pp. 4–pp.

[15] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.