scientific reports

Check for updates

OPEN

Triplet offline causal discovery based on optimal Markov blanket and its application

Waqar Khan[1]✉, Brekhna Brekhna[2], Jianqiong Huang[1], Yajun Xie[3], Muhammad Suhail Shaikh[4], Muhammad Sadiq Hassan Zada[5] & Yifan Zheng[1]

Offline constraint-based causal feature selection (OC-CFS) algorithms are essential for identifying causal relationships from observational data. However, existing methods often suffer from limitations such as low prediction accuracy or high computational cost, particularly when sample sizes vary. To address these limitations, we propose Triplet, a novel framework that leverages the HITON-MB Parents and Children (PC) strategy to identify strongly relevant PC nodes while eliminating irrelevant and redundant features. It concurrently employs the BAMB strategy to detect relevant spouses and discard irrelevant ones, and applies the STMB non-Markov Blanket (non-MB) strategy to identify and exclude non-MB descendants. Through this integration, the proposed T-OCD$_{MB}$ overcomes these limitations, accurately identifying the true MB with high prediction accuracy and reduced runtime. To validate its effectiveness, we evaluated T-OCD$_{MB}$ on benchmark Bayesian networks (BNs) and real-world datasets. Extensive experimental results demonstrate that T-OCD$_{MB}$ achieves significant improvements in both prediction accuracy and computational efficiency compared to existing methods. On small sample sizes (n=500), T-OCD$_{MB}$ achieved the highest recall in 5 out of 7 datasets, with an average improvement of over 20% compared to rivals. On large sample sizes (n=5000), it excelled in precision, achieving the top score in 4 out of 7 datasets with an average precision of 94%. Computationally, T-OCD$_{MB}$ is highly efficient, operating as the second-fastest method overall. It ran over 55% faster than half of the benchmarks and a remarkable 35% faster than the average competitor on large datasets. The source code for this research is available at the following repository: https://github.com/vickykhan89/T-OCDmb.

Keywords Offline causal discovery, Markov blanket (MB), Bayesian network (BN), High-dimensional

OC-CFS (Offline Constraint-based causal feature selection) has garnered more attentions in diverse fields, including healthcare[1,2], bioinformatics[3–7], epidemiology[8–10], and information technology[11–14]. One of the key objectives in studying such systems is to understand the causal relationships between the system's components[15]. To identify these causal relations, experts can employ causal discovery algorithms on both benchmark and real-world data. These algorithms aim to discover the MB (Markov blanket) from observational data, a crucial concept in a BNs (Bayesian networks)[16–18]. The MB is a fundamental concept in local causal discovery that identifies the minimal set of features directly related to a target feature. It consists of three key components: the target's P/direct causes (parents), C/direct effects (children), and spouses (other direct causes of the target's children). This local causal structure is significant as it serves as a building block for both local causal analysis and global causal discovery tasks[19]. Moreover, the MB of a target feature is theoretically the optimal solution to the feature selection problem[20]. As a result, it has attracted significant interest, leading to the development of constraint-based, score-based, and hybrid algorithms for its discovery. The focus of our paper is on the OC-CFS category, which includes several families, such as simultaneous MB discovery[21], divide-and-conquer MB discovery[22], MB discovery with interleaving PC and spouses learning[23], MB discovery with relaxed assumptions[24], and MB learning for special purpose[25]. For more details, see[20,26]. Each OC-CFS algorithm has its own set of assumptions, which may or may not be suitable for a specific dataset. Therefore, no single algorithm stands out in all situations.

[1]School of Big Data, Fuzhou University of International Studies and Trade, Fuzhou 350202, China. [2]Department of Computer Science, Shaheed Benazir Bhutto Women University, Peshawar 00384, Pakistan. [3]Key Laboratory of Data Science and Intelligent Computing, Fuzhou University of International Studies and Trade, Fuzhou 350202, China. [4]School of Physics and Electronic Engineering, Hanshan Normal University, Guangdong 521000, China. [5]College of Computing and Engineering, University of Derby, Derby DE22 1GB, United Kingdom. ✉email: wangkang@fzfu.edu.cn

One approach to address this limitation is to use of triplet frameworks, which combine several algorithms from different families to enhance the interpretability and robustness of the models (Method, Model, and Algorithm are used interchangeably in this paper).

However, this research focus on a different type of triplet framework, which combines core components from the most significant OC-CFS algorithms. The main advantage of OC-CFS approaches is that they are non-parametric; for example, no assumption is made about the functional form of the underlying causal relationships[27]. OC-CFS approaches learn the MB of a target feature by discovering the dependencies between the target feature and other features through the execution of CI (conditional independence) tests[28]. OC-CFS algorithms fall into three primary categories: simultaneous MB discovery, divide-and-conquer MB discovery, and MB discovery with interleaving PC and spouse learning. Algorithms that operate simultaneously can identify the MB of a target feature by detecting its parents, children, and spouses all at once. These algorithms employ CI tests that utilize the complete candidate MB as the conditioning set, which can lead to improved computational efficiency and fewer required tests. However, this approach faces a significant limitation: as the conditioning MB set grows larger, it demands increasingly large sample sizes to maintain reliable CI testing. In response to these challenges, researchers have developed alternative approaches based on divide-and-conquer principles. These methods first identify the PC set before determining the spouse set separately. While this approach enhances data efficiency through the use of multiple smaller conditional sets during independence testing, it comes at the cost of increased computational time. To overcome these issues, researchers have proposed MB discovery with interleaving PC and spouse algorithms which is the extension of divide-and-conquer approach. Instead of discovering PC and identifying spouses separately, this approach alternates between the PC discovery step and the spouse identification step. Specifically, once a candidate member of PC of target feature is added to the PC at the PC learning step, this approach triggers the spouse discovery step immediately. By interleaving PC and spouse discovery, this approach attempts to keep both PC and spouse sets as small as possible, thereby achieving a trade-off between data efficiency and time efficiency. However, due to false positive PC inclusions, many false positive spouses may enter the spouse set, leading to a large size of the spouse set, which degrades the performance of this approach. Currently, researchers face the dual challenge of improving computational efficiency while maintaining stable prediction accuracy. To address the aforementioned issues, the main challenges, objectives, and contributions of this article are as follows.

Research Challenges, Objectives, and Contributions. Although OC-CFS algorithms are fundamental for MB discovery, they face critical challenges that hinder their performance: (1) achieving high computational efficiency when discovering the strongly relevant PC set; (2) ensuring the complete identification of strongly relevant spouses from the non-PC set; (3) effectively eliminating non-MB features from the final MB set; and (4) maintaining an optimal balance between prediction accuracy and algorithmic efficiency.

To overcome these challenges, this paper introduces a novel Triplet framework, named T-OCD$_{MB}$, as illustrated in Fig. 1. The primary objective is to accurately and efficiently discover the MB from observational data by integrating the most effective components of existing algorithms.

The main contributions of this work are summarized as follows:

1. We propose the T-OCD$_{MB}$ framework, a novel integration of the HITON-MB (for PC discovery), BAMB (for spouse discovery), and STMB (for non-MB descendant removal) algorithms. This unified design is the first to seamlessly combine these three steps, enabling the identification of an accurate MB with superior predictive accuracy and significantly reduced computational runtime.
2. We perform a comprehensive stability analysis of the T-OCD$_{MB}$ algorithm under varying parameter α values and sample sizes (measured by the Rate of Instance, $|R|$). This analysis demonstrates the robustness and consistent performance of our framework across different data conditions.
3. We provide extensive theoretical and empirical validation on benchmark BNs and real-world datasets. Through statistical analysis (using Friedman test), we demonstrate that T-OCD$_{MB}$ achieves statistically significant improvements in performance compared to state-of-the-art methods.

The remainder of the paper is organized as follows: Section "Background" discusses related work and details the steps that compose different constraint-based algorithms. In "Literature review" provides the background for learning the MB algorithms used in this paper. Section "Framework of T-OCDMB algorithm" proposes the T-OCD$_{MB}$ algorithm. Section "Results and discussion" reports the empirical results. Section "Statistical analysis" presents real-world application scenarios. Finally, Section "Conclusion" concludes the paper, discusses its limitations and suggests directions for future work.

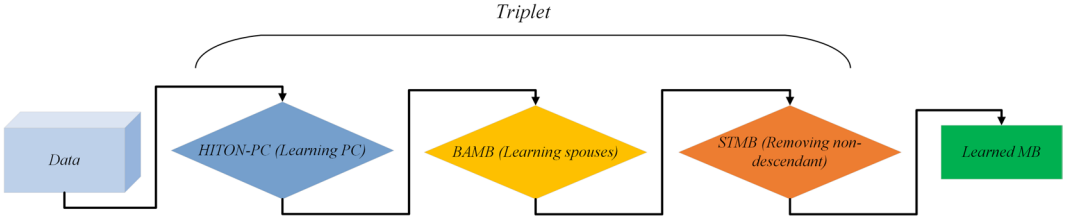


Figure 1. Flowchart of T-OCD$_{MB}$.

Background

This section introduces the fundamental notation and terminology employed throughout our work. Following standard conventions in probabilistic graphical modeling, we denote random variables using uppercase letters $(X; Y)$ and their deterministic counterparts with corresponding lowercase letters $(x; y)$. Sets are represented using uppercase letters $(\mathbb{S}, \mathbb{D})$. For directed graph structures, let $G = (R, E)$ denote a graph with vertex set R and edge set E, where for any node $X \in R$, we denote its parent nodes as $Pa_G(X)$ and its children as $Ch_G(X)$. All probability distributions are denoted by P. This consistent notational framework ensures unambiguous interpretation of subsequent theoretical developments.

Definition 1 (*MB*)[29]. In a BN, the MB of a target *T*, denoted MB_T, is the minimal set of features that renders target *T* (Target *T* and T are used interchangeably in this paper.) conditionally independent of all other features in the network. Formally, $T \perp\!\!\!\perp X | MB_T$, for $\forall X \in R \setminus \{T \cup MB_T\}$. This set consists of:

- The Parents of *T* (its direct causes).
- The Children of *T* (its direct effects).
- The Parents of *T*'s Children (its spouses).

Definition 2 (*Faithfulness*)[30]. A BN satisfies the *faithfulness condition* if and only if:

- Every CI relation present in *P* is entailed by the graph structure *G* via d-separation (see definition 7),
- There are no additional independencies in *P* beyond those implied by *G*'s Markov condition.

Mathematically, for all disjoint subsets $X, Y, Z \subseteq R$:

$$X \perp\!\!\!\perp Y \mid Z \text{ in } P \quad \Leftrightarrow \quad X \text{ is d-separated from } Y \text{ by } Z \text{ in } G.$$

Definition 3 (*d-separation*)[31]. Let $G = (R, E)$ be a DAG, *D* be a path on *G* and *Z* a subset of *R*. The path *D* is blocked by *Z* iff *D* contains:

- a fork in Fig. 2a or a chain in Fig. 2b s.t. that middle vertex *Y* is in *Z*, or
- a collider in Fig. 2c s.t. middle vertex *Y*, or any descendant of it, is not in *Z*.

Literature review

1. **Simultaneous MB:** This category has been dominant in C-CFS, with various algorithms emerging since the pioneering (KS) koller-sahami and GS (Grow-Shrink) MB algorithms. Two factors prevent the KS algorithm from guaranteeing an accurate MB: it requires knowing the MB's size in advance and limits the size of the conditioning set. Meanwhile, the GS algorithm's simple heuristic can lead to testing errors, causing it to include incorrect features in the MB. The IAMB (incremental Association MB) algorithm and its variants, such as inter-IAMB, IAMBnPC, inter-IAMBnPC, MMMBnPC, $FBED^K$, PFBP, and KIAMB, address some of these limitations. Notably, the conditioning set for GS, IAMB, and their derivatives typically constitutes a subset of commonly selected features. However, the reliability of CI tests is contingent upon the length of the MB, necessitating a proportional number of data instances. Despite their utility, these algorithms still face challenges in distinguishing between PC and spouse relationships within the discovered MB. Furthermore, their data efficiency remains limited, particularly when sample sizes are small, impacting the quality of their outputs. Addressing data efficiency concerns, Guo et al.[32] developed EAMB (Error-Aware MB), which operates through two complementary subroutines: ESMB (Efficient Simultaneous MB) and SRMB (Selectively Recover MB). The ESMB subroutine iteratively evaluates and eliminates variables from the candidate MB set during the discovery process, progressively refining the MB set through repeated updates. Complementarily,

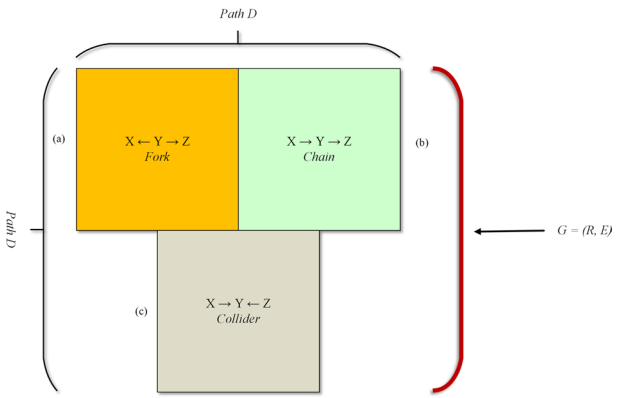


Figure 2. Illustration of three vertices *X*, *Y*, and *Z* in path *D*.

the SRMB subroutine implements an OR-AND rule mechanism to identify and recover any MB variables that may have been overlooked during the ESMB step.

2. **Divide-and-conquer MB:** The MMMB (Min-Max MB) approach reduces sample size by addressing the challenge of finding the MB through two subproblems: identifying PC and spouse relationships. It conducts a segmented examination of recently selected features to learn the target feature PC set. Building upon this, HITON-MB integrates growing and shrinking steps to exclude irrelevant/redundant ("Irrelevant," "redundant," and "false positive" are used interchangeably in this paper.) features from the PC set. However, under MB faithfulness assumptions, both MMMB and HITON-MB were found to be conceptually flawed, necessitating further enhancements for accurate MB discovery. Fu et al. introduced the IPCMB (Iterative Parent-Child-based MB) algorithm, employing a similar PC algorithm as PCMB to identify the PC set. IPCMB enhances efficiency without sacrificing accuracy, its computational speed is hindered by symmetry checks. Recognizing that large conditioning sets can compromise both data efficiency and CI tests reliability, Morais and Aussem[33] introduced the MBOR algorithm. This approach minimizes the size of conditional sets and implements an OR condition during symmetry checking to reduce false negative results. Similarly, STMB (Simultaneous MB) shares its PC exploration methodology with IPCMB but differs by identifying spouses from the entire feature set in conjunction with the current PC set, eliminating the need for symmetry checks at the cost of greater computational intensity.
3. **MB discovery with interleaving PC and spouse:** Ling et al. proposed the BAMB (Balanced MB), while Wang et al. introduced the EEMB (Efficient and Effective MB). These methods employ distinct strategies: BAMB learns the PC and spouse set simultaneously, discarding irrelevant/redundant features in a single step. In contrast, EEMB divides the process into distinct growing and pruning phases. Additionally, Zhaolong et al. devised the MBFS (MB discovery for Feature Selection) algorithm, a specialized subroutine MB algorithm that employs mutual information for PC discovery and distinguishes spouses from the PC set. While efficient, this approach may compromise prediction accuracy. The FSMB (Feature Selection via MB)[23] algorithm represents an enhancement of STMB that employs an alternative strategy for learning the PC set. In summary, offline OC-CFS algorithms have advanced in efficiency or accuracy, but only a few (e.g., BAMB, EEMB, FSMB) attempt to balance both. Their main limitations are an inefficient conditioning strategy that sacrifices data for computational speed, and a critical flaw where errors in the initial PC set propagate to the spouse set. This error propagation inflates the spouse set and ultimately degrades the performance of learning the MB.

Framework of T-OCD$_{MB}$ algorithm

The proposed Triplet framework (T-OCD$_{MB}$) employs CI testing, utilizing the G^2 test for discrete datasets and Fisher's z-test for continuous datasets. The general framework of T-OCD$_{MB}$ is illustrated in Fig. 3, with its detailed pseudocode presented in Algorithm 1.

General idea of the T-OCD$_{MB}$ algorithm

The T-OCD$_{MB}$ algorithm integrates strategies from three established methods–HITON-MB, BAMB, and STMB–to efficiently and accurately identify the MB. The overall workflow and pseudo-code, depicted in Fig. 3 and Algorithm 1, take the entire feature set R and target T as inputs. The procedure unfolds in three principal stages.

Require: *Data D; target, T;*
Ensure: *Return MB of target T*
$PC_T \leftarrow \varnothing$, $spouse_T \leftarrow \varnothing$
repeat
  // /*Step 1: Identifying the PC by utilizing Algorithm 1*/
  $[PC_T, Sep_T] \leftarrow RecogPC_{HITON-PC}(T, PC_T, D)$
  // /*Step 2: Identifying the spouses by utilizing Algorithm 2*/
  $[spouse_T, Sep_T] \leftarrow Recogspouse_{BAMB}(T, PC_T, spouse_T, D)$
  // /*Step 3: Removing the non-MB descendants by utilizing Algorithm 3*/
  $[PC_T] \leftarrow DisPC_{STMB}(T, PC_T, spouse_T, D)$
  $[MB_T] \leftarrow PC_T, spouse_T$
until *no more variables are left to consider for processing*

Algorithm 1. T-OCD$_{MB}$

(1) PC Identification.

In Step 1, the T-OCD$_{MB}$ algorithm uses Theorem 1 to discovers a strongly relevant ("Strongly relevant," and "true positive" are used interchangeably in this paper.) PC for the target T by performing a CI test between each feature in R excluding target T, for example $X_i \in R \setminus T$. Features exhibiting dependence with target T in the PC set, while others are added to a non-PC set.

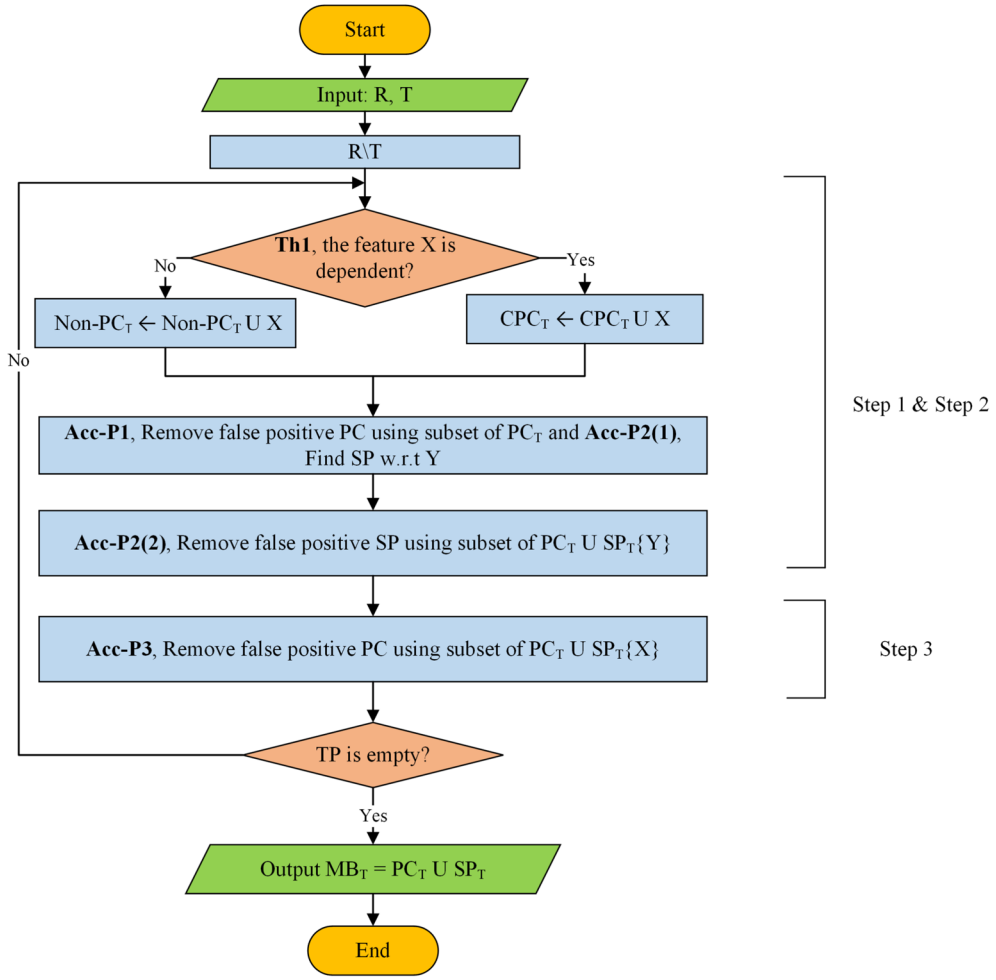


Figure 3. The block diagram of T-OCD$_{MB}$.

Require: *Data D; target, T;*
Ensure: *Return PC of T*
$PC_T \leftarrow \varnothing$
repeat
  Find variable $X \notin PC_T$ that maximizes relationship (X, T) and add X into PC_T
  If there is a variable X and a subset Z of PC_T
  such that. $(X \perp\!\!\!\perp T|Z)$
  Remove X from PC_T
  Do not consider X again for processing
until *no more variables are left to consider for processing*

Algorithm 2. HITON-PC

Theorem 1 Let X and Y be two random variables. The variables X and Y are **null conditionally dependent** if and only if they are marginally dependent. That is:

$$X \not\perp\!\!\!\perp Y \mid \emptyset \quad \Longleftrightarrow \quad P(X,Y) \neq P(X)P(Y)$$

Proof 1 The proof follows directly from the definition of CI and the properties of the empty conditioning set.

1. By definition, $X \perp\!\!\!\perp Y \mid \emptyset$ iff $P(X, Y \mid \emptyset) = P(X \mid \emptyset)P(Y \mid \emptyset)$.
2. Conditioning on the empty set $\emptyset$ is equivalent to marginal probability. Therefore:

$$P(X, Y \mid \emptyset) = P(X, Y)$$
$$P(X \mid \emptyset) = P(X)$$
$$P(Y \mid \emptyset) = P(Y)$$

3. Substituting these into the first statement gives:

$$X \perp\!\!\!\perp Y \mid \emptyset \quad \Longleftrightarrow \quad P(X, Y) = P(X)P(Y)$$

4. Taking the contrapositive of this equivalence proves the theorem:

$$X \not\perp\!\!\!\perp Y \mid \emptyset \quad \Longleftrightarrow \quad P(X, Y) \neq P(X)P(Y) \square$$

(2) Irrelevant/redundant Removal and Spouse Discovery.

In Step 2, the T-OCD$_{MB}$ algorithm removes irrelevant/redundant PC using CI test conditioned on a subset of the current PC set (Proposition 1). Simultaneously, the T-OCD$_{MB}$ algorithm discovers spouse linked through V-structure (Proposition 2(2)). This step also initiates the removal of irrelevant spouses from the spouse set ($spouse_T$) by performing CI tests conditioned on the union of the PC and spouse sets.

Require: *Data D; target, T;*
Ensure: *Return spouse of T*
$spouses_T \leftarrow \varnothing$
repeat
  for *each* $X \in \{U \setminus \{T\} \setminus spouse_T\}$ **do**
    if $X \not\perp\!\!\!\perp T | Sep_T \{X\} \cup \{Y\}$ **then**
      $spouse_T \{Y\} \leftarrow spouse_T \{Y\} \cup \{X\}$
  $spouses_Y \leftarrow \varnothing$
  $B \leftarrow X \not\perp\!\!\!\perp T | Sep_T \{X\} \cup \{Y\}$
  $spouse_T \{A\} \leftarrow spouse_T \{A\} \setminus \{B\}$
  $spouse_T \{A\} \leftarrow spouse_T \{A\} \cup \{B\}$
  for *each* $X \in spouse_T \{A\}$ **do**
    if $X \perp\!\!\!\perp T | Z \cup Y$ **then**
      $spouse_T \{Y\} \leftarrow spouse_T \{Y\} \setminus \{X\}$
until *no more variables are left in* $spouse_T \{Y\}$ *for processing*

Algorithm 3. BAMB-spouse

Proposition 1 Feature *X* have no relationship with the *Y*, given *Z*, such as $X \perp\!\!\!\perp Y | Z$, then $X \notin PC_T$.

Proof 2 By Definition 1, the following holds:

$$P(X, Y|Z) = \frac{P(X,Y)}{P(Z)} = P(X|Z)\, P(Y|Z) \Rightarrow X \perp\!\!\!\perp Y|Z$$

Feature *X* and *Y* both belongs to *R*, i.e., $X, Y \in R$ are conditionally independent given *Z*. Therefore, feature *X* should be removed from PC of target feature *T*, i.e., PC_T. □

Therefore, proposition 1 is proven.

Proposition 2 [34] Under the faithfulness assumption, a Bayesian network over a set of variables *U* satisfies the following graphical criteria:

1. **Adjacency Criterion:** For any distinct pair of nodes $X \in U$ and $Y \in U$, *X* and *Y* are adjacent if and only if they are conditionally dependent given any subset $Z \subseteq U \setminus \{X, Y\}$.

$$X \not\perp\!\!\!\perp Y \mid Z \quad \text{for all} \quad Z \subseteq U \setminus \{X, Y\}$$

2. **V-Structure Criterion:** For any distinct triplet of nodes $X, Y, Z \in U$ that form a triple $(X - Z - Y)$, they form a v-structure $(X \rightarrow Z \leftarrow Y)$ if and only if there exists a set $S \subseteq U \setminus \{X, Y, Z\}$ such that *X* and *Y* are independent given *S* but become dependent when *Z* is added to the conditioning set.

$$X \perp\!\!\!\perp Y \mid S \quad \text{and} \quad X \not\perp\!\!\!\perp Y \mid S \cup \{Z\}$$

(3) Non-MB Descendant Removal.

In step 3, the T-OCD$_{MB}$ algorithm purges non-MB descendants that may have infiltrated the PC and spouse sets. This is achieved by performing CI tests for each feature X conditional on the entire current MB ($PC_T \cup spouse_T \{Y\} \setminus \{X\}$), as per Proposition 3. Features found to be independent are removed.

Proposition 3 Descendants of the target T are the source of false positives F in PC_T.

Proof 3 By the Markov blanket condition, any non-descendant F_i is independent of T given its parents ($X_i \perp\!\!\!\perp T \mid P_T$) and is thus excluded from PC_T. Therefore, any false positive F included in PC_T must be a descendant of T. □

The algorithm terminates when no more features can be processed, outputting the optimal Markov blanket as the union of the final purified PC and spouse sets ($MB_T = PC_T \cup spouse_T$).

1 **Require:** *Data D; target, T;*
2 **Ensure:** *Return PC and spouse of target T*
3 $spouses_T \leftarrow \varnothing$
4 **repeat**
5 $D \leftarrow PC_T$
6 **for** *each* $X \in D$ **do**
7 **if** $X \perp\!\!\!\perp T | PC_T \cup spouse_T \{Y\} \setminus \{X\}$ **then**
8 $PC_T \leftarrow PC_T \setminus \{X\}$
9 $MB \leftarrow spouses_T \cup PC_T$
10 **until** *no more variables are left in* PC_T *and* $spouse_T$ *for processing*

Algorithm 4. STMB-non-descendants

The proposed T-OCD$_{MB}$ algorithm and analysis

This section details the implementation of T-OCD$_{MB}$, as formalized in Algorithm 1. The algorithm is architected around three core modules, each leveraging a distinct strategic strength from existing methods.

Step 1: PC Recognition via HITON-PC. The algorithm employs the HITON-PC strategy (Algorithm 2) to identify the PC set. HITON-PC uses an Interleaved Forward-Backward Search (IFBS) that alternates between adding the feature with the strongest association with T (forward phase, line 6 in Algorithm 2) and immediately removing any features that become conditionally independent given any subset of the current PC set (backward phase, lines 7-9 in Algorithm 2). This iterative process ensures the PC set is robust against irrelevant/redundant features.

Step 2: Spouse Recognition via BAMB. The algorithm adopts the BAMB strategy (Algorithm 3) for spouse discovery. Instead of searching for spouses only within the PC sets of other features, BAMB efficiently discovers spouses directly from the set $R \setminus \{T\} \setminus PC_T$ (lines 5-7 in Algorithm 3). A feature Y is added to the spouse set $spouse_T(X)$ (line 7) if it is dependent on T conditioned on a subset of the separating set of X and Y. Crucially, BAMB interleaves candidate addition with an irrelevant/redundant removal step (lines 8-14), ensuring the spouse set remains minimal and accurate throughout the search, enhancing both efficiency and reliability.

Step 3: Non-MB Descendant Removal via STMB. The final purification step utilizes the STMB strategy (Algorithm 4) to eliminate non-MB descendants that may remain in the PC or spouse sets. This is vital as such descendants, while possibly correlated, are not part of the true MB. For each feature X in the current PC_T (line 5) and $spouse_T$ (line 8), the algorithm tests if X is independent of T given the rest of the MB ($MB_T \setminus \{X\}$).

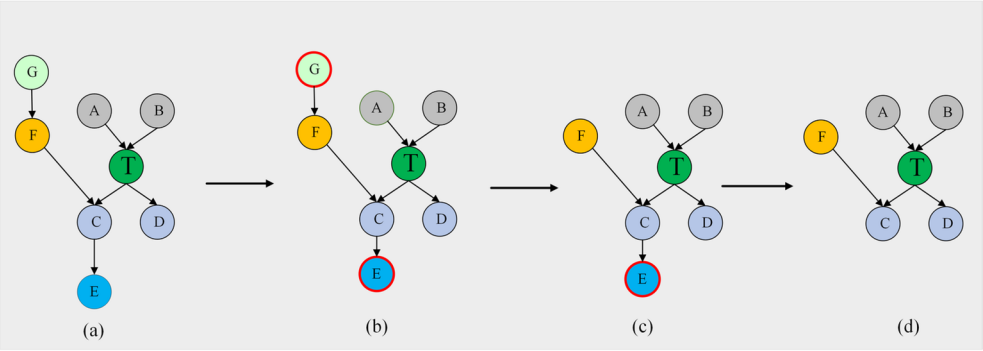


Figure 4. Illustration of the execution of T-OCD$_{MB}$.

If independent, X is removed (lines 6 and 9). This step guarantees that the final output satisfies the Markov condition.

Here, we provide an example to demonstrate the execution process of T-OCD$_{MB}$ using BN shown in Fig. 4. For our analysis, we designate T as the target feature with its MB_T initialized as $\{A, B, C, D, F\}$. This configuration serves as our baseline for tracing the algorithm's step-by-step operation and evaluating its performance.

Step 1 of T-OCD $_{MB}$. Referring to the simple network shown in Fig. 4a. We observe that $F \perp\!\!\!\perp T|\emptyset$ and $G \perp\!\!\!\perp T|\emptyset$, therefore, features F and G are not added to CPC_T. Notice that, the true PC set includes, $PC_T = \{A, B, C, D\}$. However, no subsets within PC_T make E conditionally independent of T, as evidence by $E \not\perp\!\!\!\perp T|\emptyset$ and $E \not\perp\!\!\!\perp T|C$.

Step 2 of T-OCD $_{MB}$. Since $F \not\perp\!\!\!\perp T|C$, feature F is added to $CSP_T\{C\}$. Note that, the true positive spouse set only includes, $CSP_T\{C\} = \{F\}$. Although $G \notin CSP_T\{C\}$ in the true graph, it would be incorrectly added by the algorithm due to $G \not\perp\!\!\!\perp T|C$.

Thus, as shown in Fig. 4c, after Step 1, there are some non-MB features in the *CPC* and *CSP* of target feature T: $CPC_T = \{A, B, C, D, \mathrm{E}\}$ and some spouses parents features of target feature T: $CSP_T\{C\} = \{F, \mathrm{G}\}$. Therefore, step 3 is necessary to remove these irrelevant/redundant.

Step 3 of T-OCD $_{MB}$. As shown in Fig. 4b, there exists $Z \subseteq CPC_T \cup CSP_T(C) \setminus G$ such that the conditioning set $Z \cup C$ makes feature G conditionally independent of T: $G \perp\!\!\!\perp T \mid C, F$. Therefore, G is removed from $CSP_T(C)$. Similarly, as shown in Fig. 4c, there exists $Z \subseteq CPC_T \setminus E$ such that the conditioning set $Z \cup CSP_T(C)$ makes E conditionally independent of T: $E \perp\!\!\!\perp T \mid C, F$. Consequently, E is removed from CPC_T. After completing Steps 1-3, T-OCD$_{MB}$ correctly identifies all and only the true MB features of target T.

Time complexity

The time complexity of MB discovery algorithms, a critical determinant of their scalability, is governed by the number of required CI tests. This computational cost is primarily driven by two factors: the process of identifying the PC set and the subsequent discovery of spouses, both of which are influenced by the sizes of the entire feature set ($|R|$) and the target's PC set ($|PC|$).

Our proposed T-OCD$_{MB}$ algorithm achieves an efficient complexity of $O\left(|R|.2^{|PC|}\right)$. This performance stem from the novel integration of strategies from HITON-MB, BAMB, and STMB. For instance, while HITON-PC interleaves forward and backward phases for a complexity of $O(|R||PC|)$, and BAMB employs a different approach, our synthesis optimizes the overall search process. Consequently, T-OCD$_{MB}$ confines its exponential operations to the size of the ($|PC|$), rather than the entire feature set. In practice, the computation time is also influenced by the network structure, and the algorithm typically performs faster than the worst-case complexity, achieving greater speed improvements in larger, more connected networks.

This result places T-OCD$_{MB}$ within the most efficient modern complexity class. As summarized in Table 1, the algorithmic landscape can be divided into distinct tiers. The simplest algorithm, IAMB, achieves $O(|R|^2)$ complexity through a greedy strategy, though this can sacrifice accuracy. In contrast, algorithms like IPCMB and STMB face a prohibitive $O(|R|2^{|R|})$ complexity, rendering them unsuitable for high-dimensional data. An intermediate tier, including MMMB, HITON-MB, and PCMB, exhibits a complexity of $O(|R|.|PC|.2^{|PC|})$, which is hampered by an additional multiplicative $|PC|$ factor. T-OCD$_{MB}$ belongs to the superior tier alongside BAMB, EEMB, EMB, and FSMB, all sharing the $O(|R|2^{|PC|})$ complexity. Given that $|PC| \ll |R|$ for most features in real-world networks, this confers a significant scalability advantage, making T-OCD$_{MB}$ both theoretically efficient and practical for large-scale applications.

Theoretical correctness of the T-OCD$_{MB}$

Theorem 2 Under the faithfulness assumption, the T-OCD$_{MB}$ algorithm is guaranteed to outputs all and only the optimal MB of the given target T.

Proof 4 In Step 1, Algorithm 1 (T-OCD$_{MB}$) identifies the true PC features within the MB of the target T. Features conditionally dependent on target T given the empty set are added to the PC set, denoted as PC$_T$. Con-

Algorithms	Time complexity	Algorithms	Time complexity
IAMB	$O\left(\lvert R^2\rvert\right)$	MMMB	$O\left(2^{\lvert PC\rvert}\lvert R\rvert\lvert PC\rvert\right)$
HITON-MB	$O\left(2^{\lvert PC\rvert}\lvert R\rvert\lvert PC\rvert\right)$	PCMB	$O\left(2^{\lvert PC\rvert}\lvert R\rvert\lvert PC^2\rvert\right)$
IPCMB	$O\left(2^{\lvert R\rvert}\lvert R\rvert\lvert PC\rvert\right)$	STMB	$O\left(2^{\lvert R\rvert}\lvert R\rvert\right)$
BAMB	$O\left(2^{\lvert PC\rvert}\lvert R\rvert\right)$	EEMB	$O\left(2^{\lvert PC\rvert}\lvert R\rvert\right)$
EMB	$O\left(2^{\lvert PC\rvert}\lvert R\rvert\right)$	FSMB	$O\left(2^{\lvert PC\rvert}\lvert R\rvert\right)$
T-OCD$_{MB}$	$O\left(2^{\lvert PC\rvert}\lvert R\rvert\right)$	–	–

Table 1. Time complexity of the constraint-based algorithms.

Datasets	Features	Edges	Max in/out degree	Min/max $\|PC-set\|$
Insurance	27	52	3/7	1/9
Mildew	35	46	3/3	1/15
Child3	60	79	3/7	1/8
Hepar2	70	123	6/17	1/19
Child10	200	126	2/7	1/9
Alarm10	370	570	4/7	1/9
Pig	441	592	2/39	1/41
Gene	801	972	4/10	0/11

Table 2. Summary of the benchmark synthetic BNs.

Dataset	Features	Instances	Field of data	Dataset	Features	Instances	Field of data
crx	15	653	Business	sonar	60	208	Target identification
optdigits	64	5620	Handwriting digit recognition	coil2000	85	9822	Business
colon	2000	62	Micro-array	lung	3312	203	Micro-array
sido0	4932	12,678	Pharmacology	prostate-GE	5967	102	Micro-array
arcene	10,001	100	Mass Spectrometry	leukemia2	11,225	72	Micro-array
11Tumors	12553	174	Micro-array	SMK-CAN-187	19,993	187	Micro-array
GLI-85	22283	85	Micro-array	Dorothea	1,000,000	1950	Drug Discovery

Table 3. Summary of the real-world datasets.

versely, features conditionally independent of target T are placed into a separate Non-PC$_T$ set and are excluded from all subsequent operations in Step 1.

In Step 2, the algorithm refines the PC$_T$ set by removing false positives identified in Step 1. Simultaneously, it discovers the true spouse features of T. A feature Y is identified as a spouse if it forms a V-structure collider with T (i.e., $Y \rightarrow X \leftarrow T$). Due to its exhaustive search strategy, T-OCD$_{MB}$ guarantees that no true spouse is missed, as it evaluates all features not in the final PCT set. Additionally, T-OCD$_{MB}$ removes false positive spouses by using the union of the PC set (PC$_T$) and the spouse set (Spouse$_Y$) as the conditioning set.

In Step 3, T-OCD$_{MB}$ removes non-MB features. As shown in Fig. 4b,c, although the path $E-C-T$ is blocked by feature C, an alternative path $E-F-C-T$ connects feature E to target T. Furthermore, since T-OCD$_{MB}$ finds spouses from the Non-PC set, feature G could form the V-structure ($G \rightarrow C \leftarrow T$). Consequently, two types of false positive may exist:

- Non-child descendants of T in the PC set.
- Parents of T's spouses in the spouse set.

T-OCD$_{MB}$ uses Definition 1 to remove these false positives. The candidate PC and spouse sets together form the MB of target T. The algorithm directly removes parents of spouses from the candidate spouse set. True spouses are not removed because the conditioning set always includes the common child of T and its spouses. Thus, after Step 2, T-OCD$_{MB}$ retains only the true spouses. The candidate PC set contains all true PC members of T. The union of the PC set and the spouse set together constitute the MB of T. The algorithm then removes non-child descendant nodes from the candidate PC set. Since the true PC features are always dependent on target T given any subset in R, only the true PC of T remain after Step 3. Therefore, T-OCD$_{MB}$ uses Theorem 1 and Propositions 1, 2, and 3 to accurately identify all and only the members of the target T's MB. □

Results and discussion

This section presents a comparison between the T-OCD$_{MB}$ algorithm and cutting-edge MB discovery algorithms, assessing both efficiency and effectiveness. The comparison is conducted using six benchmark BNs (see Table 2) and ten real-world datasets (see Table 3). The algorithms under scrutiny encompass several constraint-based methods: HITON-MB, STMB, BAMB, EEMB, EMB, and FSMB. All algorithms are implemented using MATLAB. The experiments are carried out on a Windows 11 operating system, utilizing an Intel Core i7-6200U processor with 16 GB of RAM. For evaluating independence, the G^2-test and Fisher's z-test are employed at a significance level of 0.01, with the superior outcomes being highlighted in * in superscript within the tables.

1. Effectiveness: The effectiveness of the algorithm is evaluated using two metrics, as defined in Eqs. 1[35] and 2[35]:

$$Recall = TP \div (TP + FN), \tag{1}$$

Datasets	IAMB	MMMB	HITON-MB	MBOR	STMB	BAMB	EEMB	EAMB	FSMB	T-OCD$_{MB}$
Insurance	**0.95 ± 0.03**	0.78 ± 0.05	0.79 ± 0.04	0.92 ± 0.03	0.81 ± 0.05	0.73 ± 0.01	0.84 ± 0.04	0.87 ± 0.03	0.90 ± 0.01	0.52 ± 0.02
Mildew	0.75 ± 0.02	0.35 ± 0.05	0.35 ± 0.05	0.74 ± 0.01	0.18 ± 0.01	0.53 ± 0.03	0.53 ± 0.03	0.74 ± 0.02	0.82 ± 0.02	**0.90 ± 0.02**
Child3	0.70 ± 0.01	0.71 ± 0.01	0.71±0.01	0.90 ± 0.01	0.70 ± 0.01	0.86 ± 0.01	0.67 ± 0.01	0.81 ± 0.01	0.74 ± 0.01	**0.93 ± 0.01**
Child10	0.60 ± 0.02	0.69 ± 0.02	0.69 ± 0.02	0.56 ± 0.01	0.26 ± 0.01	0.70 ± 0.02	0.65 ± 0.02	0.69 ± 0.01	0.75 ± 0.02	**0.91 ± 0.01**
Alarm10	0.70 ± 0.01	0.75 ± 0.01	0.76 ± 0.01	0.84 ± 0.01	0.31 ± 0.01	0.76 ± 0.01	0.80 ± 0.01	0.71 ± 0.01	0.88 ± 0.01	**0.89 ± 0.01**
Pig	0.82 ± 0.01	0.88 ± 0.01	0.89 ± 0.01	**0.95 ± 0.01**	0.17 ± 0.01	0.73 ± 0.01	0.72 ± 0.01	0.82 ± 0.01	0.90 ± 0.02	0.92 ± 0.01
Gene	0.69 ± 0.01	0.79 ± 0.01	0.79 ± 0.01	0.96 ± 0.01	0.16 ± 0.01	0.79 ± 0.01	0.78 ± 0.01	0.68 ± 0.01	0.89 ± 0.01	**0.90 ± 0.01**

Table 4. Precision results of T-OCD$_{MB}$ and its rivals on benchmark BNs with 500 sample size.

Datasets	IAMB	MMMB	HITON-MB	MBOR	STMB	BAMB	EEMB	EAMB	FSMB	T-OCD$_{MB}$
Insurance	0.45 ± 0.03	0.61 ± 0.03	0.61 ± 0.03	0.73 ± 0.03	0.61 ± 0.05	0.60 ± 0.02	0.58 ± 0.02	0.46 ± 0.02	0.52 ± 0.01	**0.83 ± 0.02**
Mildew	0.20 ± 0.01	0.50 ± 0.04	0.50 ± 0.04	0.28 ± 0.01	0.55 ± 0.02	0.31 ± 0.02	0.31 ± 0.02	0.20 ± 0.01	0.27 ± 0.01	**0.70 ± 0.01**
Child3	0.60 ± 0.02	0.65 ± 0.03	0.69 ± 0.01	0.60 ± 0.02	0.73 ± 0.01	0.70 ± 0.02	0.50 ± 0.02	0.69 ± 0.01	0.69 ± 0.02	**0.99 ± 0.01**
Child10	0.64 ± 0.01	0.64 ± 0.01	0.71 ± 0.02	0.55 ± 0.02	0.86 ± 0.01	0.71 ± 0.01	0.71 ± 0.01	0.68 ± 0.01	0.70 ± 0.01	**0.99 ± 0.01**
Alarm10	0.50 ± 0.01	0.62 ± 0.01	0.63 ± 0.01	0.59 ± 0.01	0.67 ± 0.01	0.61 ± 0.01	0.61 ± 0.01	0.54 ± 0.01	0.58 ± 0.01	**0.92 ± 0.01**
Pig	0.84 ± 0.01	**1.00 ± 0.01**	**1.00 ± 0.01**	0.99 ± 0.01	**1.00 ± 0.01**	0.86 ± 0.01	0.82 ± 0.01	0.81 ± 0.01	0.91 ± 0.01	**1.00 ± 0.00**
Gene	0.78 ± 0.01	0.91 ± 0.01	0.91 ± 0.01	0.95 ± 0.01	0.96 ± 0.01	0.90 ± 0.01	0.91 ± 0.01	0.77 ± 0.01	0.95 ± 0.01	**1.00 ± 0.00**

Table 5. Recall results of T-OCD$_{MB}$ and its rivals on benchmark BNs with 500 sample size.

Datasets	IAMB	MMMB	HITON-MB	MBOR	STMB	BAMB	EEMB	EAMB	FSMB	T-OCD$_{MB}$
Insurance	**0.94 ± 0.01**	0.88 ± 0.03	0.89 ± 0.03	0.92 ± 0.02	0.64 ± 0.04	0.89 ± 0.03	0.89 ± 0.02	0.90 ± 0.02	0.94 ± 0.01	0.88 ± 0.01
Mildew	0.60 ± 0.01	0.23 ± 0.01	0.23 ± 0.01	0.79 ± 0.01	0.27 ± 0.01	0.41 ± 0.01	0.67 ± 0.01	0.66 ± 0.01	0.77 ± 0.01	**0.80 ± 0.01**
Child3	0.73 ± 0.03	0.94 ± 0.02	0.95 ± 0.02	**0.97 ± 0.02**	0.66 ± 0.02	0.93 ± 0.02	0.95 ± 0.02	0.80 ± 0.02	0.91 ± 0.01	**0.97 ± 0.02**
Child10	0.87 ± 0.01	0.91 ± 0.01	0.92 ± 0.01	**0.96 ± 0.01**	0.45 ± 0.02	0.76 ± 0.01	0.65 ± 0.01	0.66 ± 0.01	0.76 ± 0.01	**0.96 ± 0.01**
Alarm10	0.80 ± 0.02	0.90 ± 0.02	0.92 ± 0.02	0.94 ± 0.01	0.40 ± 0.02	0.85 ± 0.02	0.92 ± 0.01	0.70 ± 0.01	0.90 ± 0.01	**0.97 ± 0.01**
Pig	0.62 ± 0.01	0.61±0.01	0.61 ± 0.01	0.97±0.01	0.18±0.01	0.82 ± 0.01	**0.93 ± 0.01**	0.74 ± 0.01	0.93 ± 0.01	**0.93 ± 0.01**
Gene	0.76 ± 0.01	0.77 ± 0.01	0.77 ± 0.01	0.96 ± 0.01	0.13 ± 0.01	0.64 ± 0.01	0.61 ± 0.01	0.55 ± 0.01	0.90 ± 0.01	**0.94 ± 0.01**

Table 6. Precision results of T-OCD$_{MB}$ and its rivals on benchmark BNs with 5000 sample size.

Datasets	IAMB	MMMB	HITON-MB	MBOR	STMB	BAMB	EEMB	EAMB	FSMB	T-OCD$_{MB}$
Insurance	0.88 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.73 ± 0.02	**0.98 ± 0.02**	**0.98 ± 0.02**	0.75 ± 0.02	0.65 ± 0.02	0.77 ± 0.01	0.65 ± 0.01
Mildew	0.41 ± 0.02	0.90 ± 0.03	0.90 ± 0.03	0.47 ± 0.01	0.86 ± 0.02	0.11 ± 0.02	0.40 ± 0.01	0.71 ± 0.01	0.44 ± 0.01	**0.92 ± 0.01**
Child3	0.90 ± 0.03	0.97 ± 0.02	0.97 ± 0.02	**0.97 ± 0.02**	**0.97 ± 0.02**	0.91 ± 0.02	**0.97 ± 0.02**	0.92 ± 0.02	0.88 ± 0.01	**0.97 ± 0.02**
Child10	0.88 ± 0.01	**0.99 ± 0.01**	**0.99 ± 0.01**	**0.99 ± 0.01**	**0.99 ± 0.01**	**0.99 ± 0.01**	0.60 ± 0.01	0.92 ± 0.01	0.76 ± 0.01	**0.99 ± 0.01**
Alarm10	0.64 ± 0.03	0.72 ± 0.03	0.73 ± 0.02	0.82 ± 0.01	0.76 ± 0.02	0.74 ± 0.02	0.80 ± 0.01	0.76 ± 0.01	0.87 ± 0.01	**0.84 ± 0.01**
Pig	0.96 ± 0.01	0.42 ± 0.01	0.42 ± 0.01	**1.00 ± 0.01**	**1.00 ± 0.01**	**1.00 ± 0.01**	**1.00 ± 0.01**	0.95±0.01	0.94 ± 0.01	**1.00 ± 0.00**
Gene	0.89 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.98 ± 0.01	0.99 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.88 ± 0.01	0.98 ± 0.01	**1.00 ± 0.01**

Table 7. Recall results of T-OCD$_{MB}$ and its rivals on benchmark BNs with 5000 sample size.

$$Precision = TP \div (TP + FP), \tag{2}$$

where TP: True positive, TN: True negative, FP: False positive, and FN: False negative. Recall measures the proportion of actual MB features that are correctly identified. Precision measures the proportion of identified MB features that are actually correct. The F1-score represents the harmonic mean of precision and recall.

2. Efficiency: Algorithm efficiency is evaluated using two metrics:

   - CI test: The exact CI tests may range from a few tests in smaller-scale studies to a larger number in more extensive analyses involving multiple features.

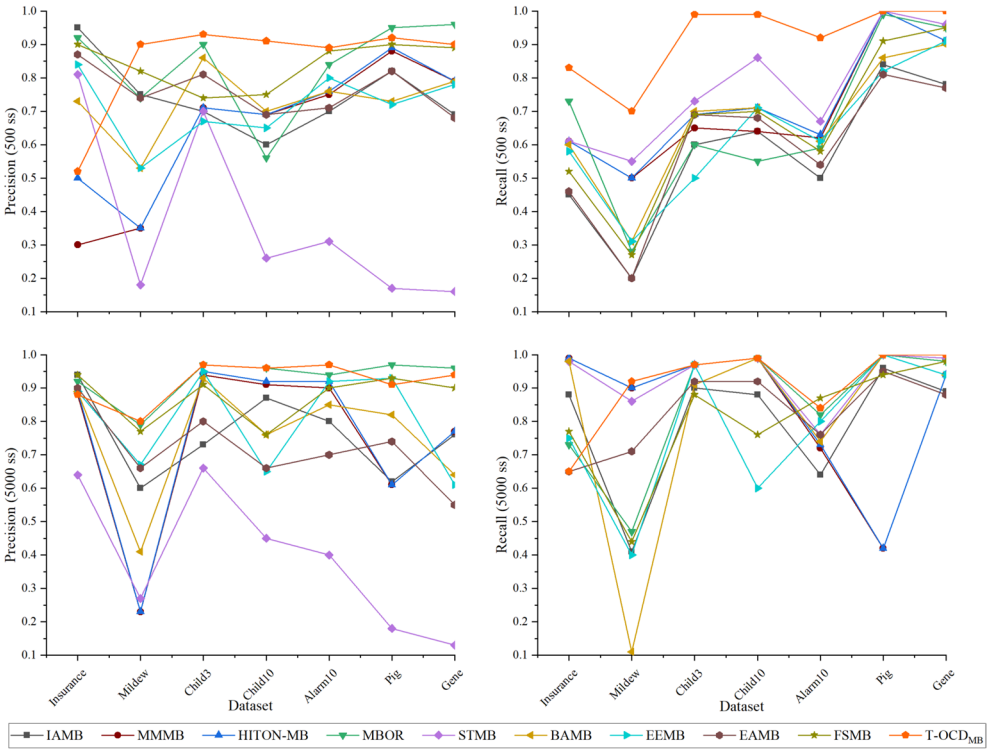


Figure 5. Prediction accuracy of the T-OCD$_{MB}$ and its rivals using 12 datasets.

Datasets	IAMB	MMMB	HITON-MB	MBOR	STMB	BAMB	EEMB	EAMB	FSMB	T-OCD$_{MB}$
				Sample size = 500						
Average time (s)	**4.9**	7.7	6.4	8.1	6.5	5.5	5.3	6.2	5.5	5.1
				Sample size = 5000						
Average time (s)	**8.3**	28.3	17.3	29.6	25.7	14.8	13.3	15.1	13.3	12.9

Table 8. Average running time of T-OCD$_{MB}$ and its rivals on benchmark BNs with 500 sample size.

- Time: The efficiency of these algorithms is evaluated by measuring their runtime in seconds.

Evaluating algorithms on benchmark BNs

We conduct a systematic evaluation of the algorithms' prediction accuracy using seven standard benchmark BNs. Detailed results are presented in Tables 4, 5, 6 and 7 and visualized in Fig. 5, which shows precision and recall values across the benchmark networks. Average running times are summarized in Table 8. The evaluation uses datasets of 500 and 5000 instances for each BN and employs the previously described metrics to assess the accuracy of MB feature identification.

Sample size 500:

Precision: T-OCD$_{MB}$ outperforms its competitors in terms of precision across all benchmark BNs, with the exception of IAMB on the Insurance. This discrepancy may be due to the reliability of our independence test for this particular benchmark. However, for all other benchmarks with varying feature numbers, T-OCD$_{MB}$ achieves the highest precision compared to IAMB, MMMB, HITON-MB, MBOR, STMB, BAMB, EEMB, EAMB, and FSMB. As a result, the feature set produced by T-OCD$_{MB}$ includes more relevant features in the MB of the target feature T and generates fewer false positives.

Recall: The proposed T-OCD$_{MB}$ algorithm demonstrate superior recall across all benchmarks, particularly on BNs with relatively few features such as Insurance, Mildew, Child3, Child10, Alarm10, Pig, and Gene. This can be attributed to the fact that, in the context of fixed features, when the BN contains a small number of features, T-OCD$_{MB}$ can identify the optimal candidate feature set for each target feature T from the available features, similar to other MB discovery algorithms, but without the need for global information about all features. However, on the Pig dataset, MMMB, HITON-MB, and STMB achieve recall performance comparable to T-OCD$_{MB}$. This is due to the fact that these three algorithms' CI tests are particularly reliable on the Pig benchmark, which contains a larger number of features.

Sample size 5000:

Precision: T-OCD$_{MB}$ outperforms its competitors in terms of precision across all benchmark BNs, except for IAMB on the Insurance BN. This difference may be attributed to the reliability of our independence test for this specific benchmark. Additionally, MBOR and EEMB show comparable accuracy to the T-OCD$_{MB}$ algorithm. However, for all other benchmarks, which vary in feature count, T-OCD$_{MB}$ maintain superior precision over other algorithms. Therefore, the feature set produced by T-OCD$_{MB}$ contains more strongly true positive features in the MB of the target feature T and fewer false positives.

Recall: The proposed T-OCD$_{MB}$ is worse than HITON-MB and MMMB but better than IAMB on all benchmark BNs, particularly on child, insurance, and alarm with a small number of features. The explanation is that, in the context of fixed features, when the number of features in the benchmark is large, T-OCD$_{MB}$, MMMB, HITON-MB, and STMB can determine the best candidate feature set for each target feature T of interest from all the features at each time. However, with increasing numbers of features, MMMB, HTION-MB, and STMB, demonstrate comparable accuracy to the proposed T-OCD$_{MB}$ algorithm.

Running time across sample sizes of 500 and 5000: The average execution times, detailed in the Table 8, benchmark the computational efficiency of ten algorithms. As expected, the IAMB algorithm, known for its simplicity, records the shortest running time (4.9s for n=500; 8.3s for n=5000).

Notably, our proposed T-OCD$_{MB}$ algorithm emerges as the second-fastest method across both sample sizes, outperforming all other contemporaries including MMMB, HITON-MB, MBOR, STMB, BAMB, EAMB, and FSMB. It achieved execution times of 5.1s and 12.9s for sample sizes of n=500 and n=5000, respectively. This improvement results from the T-OCD$_{MB}$ method, which integrates the individual strengths of HITON-MB, BAMB, and STMB into a novel approach that optimizes the search process. In addition to that, the advantage of the T-OCD$_{MB}$ becomes even more pronounced in terms of prediction accuracy. T-OCD$_{MB}$ is sample size agnostic and achieved best accuracy results on both small and large sample sizes as evident in Tables 4, 5, 6 and 7. In contrast, IAMB uses the entire currently selected features for CI tests at each computation, which reduces the number of independence tests, but requires more data samples for each test since the number of data samples required is exponential to the size of the conditioning set. Thus the IAMB algorithms are computationally efficient but not data efficient. When the sample size of a data set is not sufficiently large, IAMB algorithm cannot find the MB accurately, which degrades IAMB performance in terms of prediction accuracy, as mentioned in Tables 4, 5, 6 and 7.

Experiments on real-world datasets

We evaluate the proposed T-OCD$_{MB}$ algorithm on fourteen real-world from public datasets (see Table 9) using 10-fold cross-validation. The evaluation is based on several metrics, including classifier performance, number of selected features, and running time. For Classification, we employ three classifiers from the MATLAB R2021a built-in toolbox: Fine$_{Tree}$, SVM, and Cosine$_{KNN}$. The datasets, described in Table 3, include from the UCI repository[36], text classification, face database, and bio-informatics datasets from the Gene Expression Model Selector (GEMS) project[37] and from Arizona State University (ASU)[38].

Prediction accuracy: The experimental results demonstrate compelling patterns in classification performance across different classifiers. T-OCD$_{MB}$ consistently achieves superior predictive accuracy, as shown in Table 9 and Fig. 6. For example, on the arcene dataset, T-OCD$_{MB}$ achieves 99% accuracy across all three classifiers (Fine Tree, SVM, and Cosine KNN), demonstrating its robust feature selection capability. The algorithm also excels on the sonar dataset, achieving 99% accuracy with Cosine KNN while maintaining high performance with SVM (96%) and Fine Tree (88%). On the coil2000 dataset, T-OCD$_{MB}$ achieves balanced high performance (96%, 98%, and 98% across the three classifiers, respectively), outperforming other algorithms. In contrast, HITON-MB shows varying performance, with strong results on some datasets like crx (93% across all classifiers) but inconsistent performance on others such as arcene (70%, 73%, and 69%). STMB and BAMB show competitive performance on datasets like coil2000 but experience significant accuracy degradation on complex datasets like arcene, where STMB achieves only 61%, 67%, and 71%. The performance advantage of T-OCD$_{MB}$ is most pronounced on high-dimensional datasets, where it maintains high accuracy while selecting fewer features.

Selected Features: Analysis of OC-CFS performance across multiple datasets reveals distinct patterns in efficiency and dimensionality reduction. T-OCD demonstrates exceptional efficiency, consistently identifying more compact feature sets while maintaining high accuracy. For example, on the high-dimensional GLI-85 dataset, T-OCD selected only 6 features (0.6% of the original space), compared to BAMB's 181 features (18.1%), representing a 96.7% reduction. Similarly, on SMK-CAN-187, T-OCD$_{MB}$ identified 6 features versus STMB's 46, achieving an 87% more compact set. This selective capability is evident on prostate-GE, where T-OCD$_{MB}$ used only 5 features to achieve 86% accuracy, while STMB required 12 features (140% more) for 69% accuracy. Across all datasets, T-OCD selected 35–75% fewer features than other methods while maintaining comparable or superior performance. On larger datasets like 11Tumors, T-OCD selected 12 features versus STMB's 152 (92% reduction) while achieving higher accuracy (88% vs 77%). This consistent efficiency demonstrates T-OCD$_{MB}$'s superior ability to identify causally relevant features with significantly reduced computational complexity.

Running Time: The computational performance of T-OCD$_{MB}$ was evaluated against HITON-MB, STMB, and BAMB across multiple datasets. Execution times are reported in seconds, with the fastest time for each dataset highlighted in bold.

T-OCD$_{MB}$ consistently achieved the shortest execution times, demonstrating superior computational efficiency. For instance, on the crx dataset, T-OCD completed in 0.79 seconds–6% faster than HITON-MB (0.84 s), 24% faster than STMB (1.04 s), and 4% faster than BAMB (0.82 s). Similarly, on the larger Dorothea dataset, T-OCD$_{MB}$ required 41.54 seconds, outperforming HITON-MB (69.00 s) by 39.8% and BAMB (49.85 s) by 16.7%. This efficiency advantage is consistent across datasets of varying sizes and complexities, as seen in the 11Tumors dataset where T-OCD$_{MB}$ (12.35 s) was 31.8% faster than HITON-MB (18.12 s) and 42.4% faster than

Dataset	Algorithm	Fine$_{Tree}$ Clf	SVM Clf	Cosine$_{KNN}$ Clf	Avg. Acc.	Time	Selected Features
crx	T-OCD$_{MB}$	**97**	**97**	**97**	**97.00**	**0.79**	5
	HITON-MB	88	93	93	91.33	0.84	5
	STMB	88	93	93	91.33	1.04	6
	BAMB	89	93	92	91.33	0.82	**3**
sonar	T-OCD$_{MB}$	**99**	**99**	**99**	**99.00**	**0.81**	**14**
	HITON-MB	83	86	70	79.67	0.89	59
	STMB	83	82	83	82.67	1.07	20
	BAMB	83	83	84	83.33	0.85	20
optdigits	T-OCD$_{MB}$	**89**	**94**	**90**	**91.00**	**0.89**	12
	HITON-MB	88	91	88	89.00	0.94	**5**
	STMB	77	77	77	77.00	1.11	10
	BAMB	81	81	794	318.67	0.90	**5**
coil2000	T-OCD$_{MB}$	**96**	**98**	**98**	**97.33**	**0.95**	**9**
	HITON-MB	95	96	95	95.33	1.00	13
	STMB	94	95	95	94.67	1.22	22
	BAMB	95	96	95	95.33	0.99	15
colon	T-OCD$_{MB}$	**78**	**78**	**80**	**78.67**	**1.05**	**13**
	HITON-MB	–	–	–	–	–	–
	STMB	–	–	–	–	–	–
	BAMB	–	–	–	–	–	–
lung	T-OCD$_{MB}$	**95**	**95**	**95**	**95.00**	**1.25**	**9**
	HITON-MB	–	87	68	77.50	1.62	10
	STMB	–	–	–	–	–	–
	BAMB	–	63	61	62.00	1.35	14
sido0	T-OCD$_{MB}$	**88**	**96**	**99**	**94.33**	**2.80**	**9**
	HITON-MB	–	–	–	–	–	–
	STMB	–	–	–	–	–	–
	BAMB	–	–	–	–	–	–
prostate-GE	T-OCD$_{MB}$	**86**	**78**	**78**	**80.67**	**3.05**	5
	HITON-MB	72	73	73	72.67	3.65	**2**
	STMB	69	72	77	72.67	4.18	12
	BAMB	77	78	81	78.67	3.22	12
arcene	T-OCD$_{MB}$	**99**	**99**	**98**	**98.67**	**4.85**	9
	HITON-MB	70	73	69	70.67	6.88	**4**
	STMB	61	67	71	66.33	9.56	21
	BAMB	73	70	65	69.33	4.85	15
leukemia2	T-OCD$_{MB}$	**87**	**90**	**90**	**89.00**	**7.01**	**33**
	HITON-MB	77	83	83	81.00	11.2	45
	STMB	83	86	84	84.33	15.2	35
	BAMB	82	88	85	85.00	7.03	39
11Tumors	T-OCD$_{MB}$	**88**	**91**	**89**	**89.33**	**12.35**	12
	HITON-MB	77	81	79	79.00	18.12	38
	STMB	77	83	78	79.33	21.44	152
	BAMB	77	81	79	79.00	12.40	**11**
SMK-CAN-187	T-OCD$_{MB}$	**86**	**89**	**88**	**87.67**	**18.49**	**6**
	HITON-MB	82	88	85	85.00	25.5	50
	STMB	82	87	81	83.33	35.3	46
	BAMB	85	88	86	86.33	19.05	22

Continued

Dataset	Algorithm	Fine$_{Tree}$ Clf	SVM Clf	Cosine$_{KNN}$ Clf	Avg. Acc.	Time	Selected Features
GLI-85	T-OCD$_{MB}$	**92**	**96**	**95**	**94.33**	**24.72**	**6**
	HITON-MB	89	90	88	89.00	32.00	14
	STMB	89	89	88	88.67	41.44	152
	BAMB	90	90	89	89.67	26.00	181
Dorothea	T-OCD$_{MB}$	**95**	**97**	**93**	**95.00**	**41.54**	**10**
	HITON-MB	93	93	91	92.33	69.00	15
	STMB	–	–	–	–	–	–
	BAMB	93	93	91	92.33	49.85	23

Table 9. Prediction accuracy of T-OCD$_{MB}$ and its rivals using three classifiers with a significance level of 0.01.

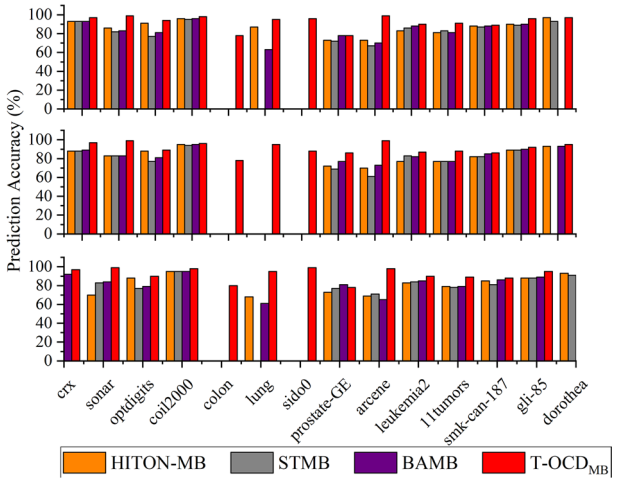


Figure 6. Prediction accuracy of the T-OCD$_{MB}$ and its rivals using 14 datasets.

STMB (21.44 s). These results confirm T-OCD$_{MB}$'s robustness and scalability in achieving faster computation while maintaining feature selection quality.

Statistical analysis

To evaluate and compare the performance of the algorithms, we conducted Friedman's test followed by Nemenyi's post hoc test to calculate the average ranks and the critical difference (CD), respectively. The Friedman test is expressed in Eq. 3:

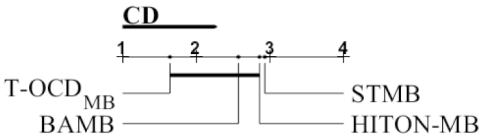
$$Fd_f = \frac{(n-1)\,\chi_f^2}{n\,(k-1) - \chi_f^2}, \tag{3}$$

where

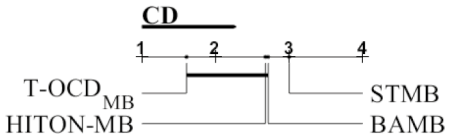
$$\chi_f^2 = \frac{12n}{k\,(k+1)} \left(\sum_{i=1}^{k} r_i^2 - \frac{k\,(k+1)^2}{4} \right), \tag{4}$$

where n and k are the numbers of real-world datasets and algorithms, respectively. The mean rank can be represented as r_i, where $i = 1, 2, 3, \ldots, k$, corresponds to the i-th algorithm across all real-world datasets. The null hypothesis is rejected by utilizing Friedman's test at a significance level of 0.01, which implies that the performance of the algorithms is not equivalent. Once the null hypothesis is rejected, we proceed with the Nemenyi post-hoc test, which identifies a significant difference between algorithms when the average ranks (r_i) differ by at least the critical difference, as presented in the following:

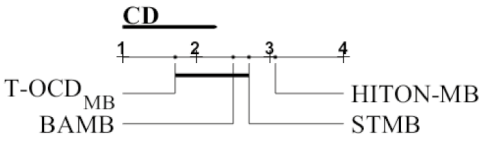
$$CD = q_\alpha \sqrt{\frac{k\,(k+1)}{6n}}, \tag{5}$$



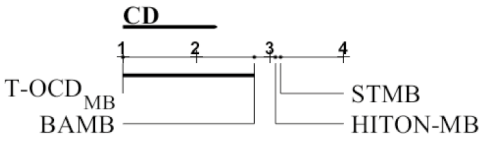
(a) Fine Tree



(b) SVM



(c) Consine KNN



(d) Average Performance of three Classifiers

Figure 7. Critical Difference (CD) diagrams of the Nemenyi post-hoc test for algorithm performance based on three classifiers: (**a**) Fine Tree, (**b**) SVM, (**c**) Cosine KNN, and (**d**) the average ranking across all three classifiers.

where q_α denotes the critical value from the statistical table. If the average ranks (r_i) of two algorithms are within one CD, they are not significantly different.

The T-OCD$_{MB}$ algorithm was evaluated against its rivals (HITON-MB, BAMB, STMB) using classifiers. The statistical tests were performed on the results from three different classifiers: Fine Tree, SVM, Cosine KNN, and their average. The resulting CD diagrams are shown in Fig. 7.

The Friedman test, performed at a significance level of $\alpha = 0.01$, rejected the null hypothesis for all three classifiers, indicating statistically significant differences in performance. The p-values were $p = 0.014934$ for Fine Tree, $p = 0.012599$ for SVM, and $p = 0.024563$ for Cosine KNN. Furthermore, the average performance across all three classifiers also showed a highly significant difference ($p = 1.0269 \times 10^{-06}$).

Following this, we conducted the Nemenyi post-hoc test. The resulting CD diagrams (Fig. 7) highlight the following key findings:

1. Superior Performance of T-OCD$_{MB}$: In all three individual classifier tests and the average ranking, T-OCD$_{MB}$ consistently achieved the highest rank (closest to 1). Its performance was statistically superior to several rivals, as its average rank was consistently outside the critical difference line of other algorithms.
2. Classifier-Dependent Variations: The precise ranking order between the rival algorithms (HITON-MB, BAMB, STMB) varied depending on the classifier used. For instance, the performance of HITON-MB and BAMB was more competitive when evaluated using the SVM and Cosine KNN classifiers compared to the Fine Tree classifier.
3. Overall Significance: The highly significant p-value ($p = 1.0269 \times 10^{-06}$) for the average performance across classifiers confirms that T-OCD$_{MB}$'s overall superiority is not an artifact of a specific classification model but is robust across different evaluation methods.

Overall, the statistical analysis confirms that T-OCD$_{MB}$ achieves significantly better performance than its rivals, as evidenced by its consistently highest average rank in the CD diagrams.

Stability analysis

The stability of T-OCD$_{MB}$ and its competitors (HITON-MB, STMB, and BAMB) was evaluated with respect to parameters α and $|RI|$. This analysis used high-dimensional real-world healthcare datasets (lung cancer, yeast, and 11 tumors) with α values of 0.1, 0.01, 0.05 and sample size ratios $|RI| = 0.1, 0.8, 0.9$. Here, α represents the

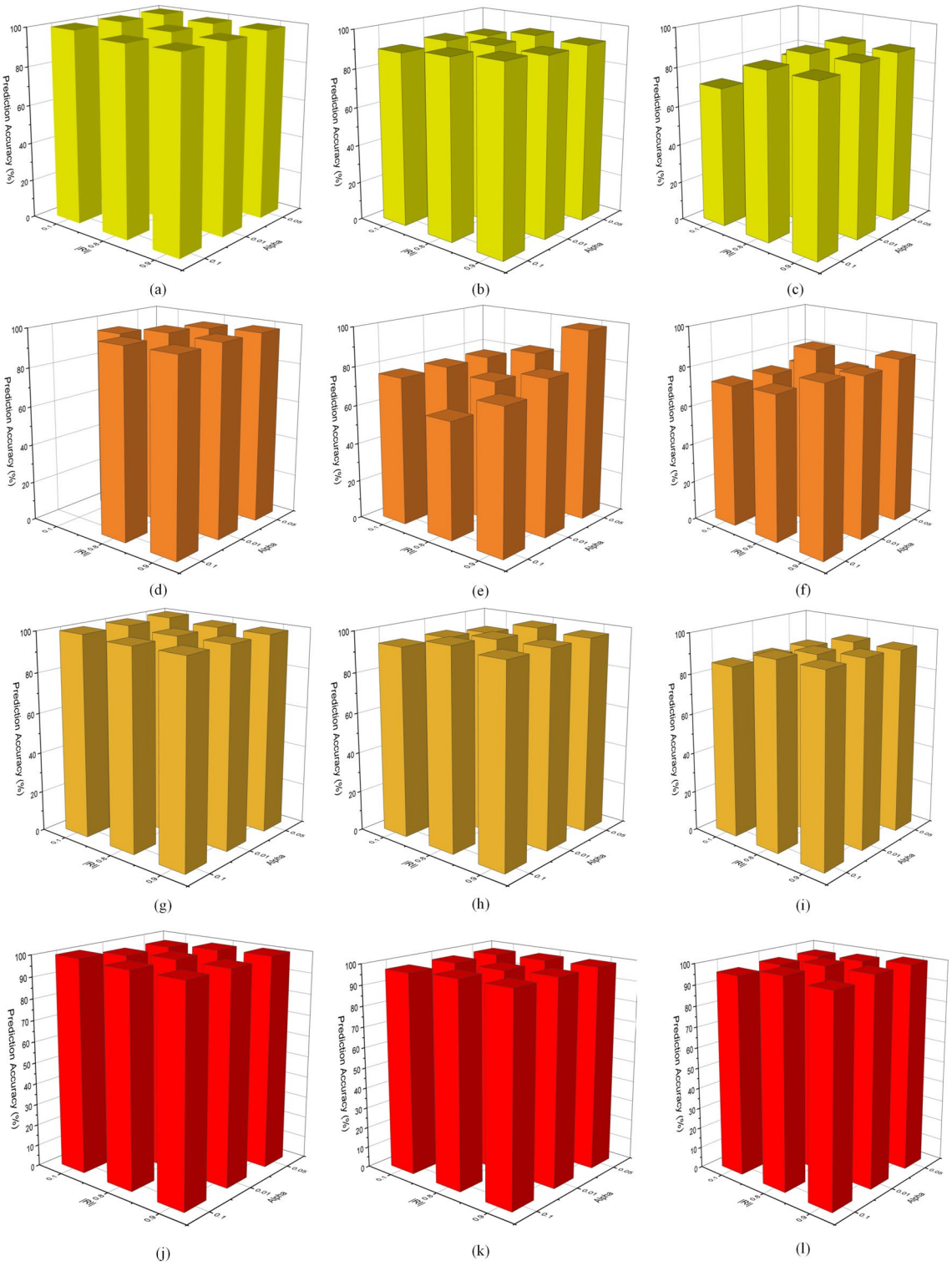


Figure 8. Stability analysis of T-OCD$_{MB}$ and its competitors: (**a**–**c** HITON-MB; **d**–**f** STMB; **g**–**i** BAMB; **j**–**l** T-OCD$_{MB}$) on three real-world datasets: lung cancer, yeast, and 11 tumors.

significance level for conditional independence tests, while $|RI|$ indicates the proportion of data instances used. Prediction accuracy was assessed using a Linear SVM classifier in MATLAB R2021a.

As shown in Fig. 8, T-OCD$_{MB}$ maintained consistent prediction accuracy across parameter variations, performing comparably to or better than the other algorithms. In high-dimensional datasets with a fixed number of features, T-OCD$_{MB}$ demonstrated improved stability in certain scenarios. Specifically, T-OCD$_{MB}$

outperformed its competitors by achieving an average stability improvement of approximately 5% to 10% compared to HITON-MB and 8% to 12% over STMB and BAMB.

Conclusion

This paper addresses the challenges and limitations encountered by existing OC-CFS algorithms, which significantly degrade their performance. To overcome these issues, we propose T-OCD$_{MB}$, a novel and robust triplet framework for offline constraint-based Markov Blanket (MB) discovery. The framework integrates a three-stage strategy–utilizing HITON-PC for PC discovery, BAMB for spouse discovery, and STMB for the removal of non-MB descendants. This integration not only improves and stabilizes prediction accuracy but also achieves a better balance between accuracy and computational efficiency, resulting in a faster running time. The performance of T-OCD$_{MB}$ is extensively evaluated and compared against state-of-the-art OC-CFS algorithms on benchmark BNs and real-world datasets. Quantitatively, the superiority of our approach is evident: on small sample sizes (n=500), T-OCD$_{MB}$ achieved the highest recall in 5 out of 7 datasets, representing an average improvement of over 20% compared to its closest competitors. On large sample sizes (n=5000), it excelled in precision, ranking first in 4 out of 7 datasets with an exceptional average precision of 94%. Furthermore, the framework is computationally highly efficient, operating as the second-fastest method overall and running 35% faster than the average competitor on large datasets. Future work could focus on validating the framework across a wider range of BN and real-world datasets and extending it to local causal discovery.

Data availability

The research data supporting this study are available from the corresponding author upon appropriate request.

Received: 31 May 2025; Accepted: 13 October 2025
Published online: 18 November 2025

References

1. Morid, M. A., Sheng, O. R. L. & Dunbar, J. Time series prediction using deep learning methods in healthcare. *ACM Trans. Manag. Inf. Syst.* **14**, 1–29 (2023).
2. Korial, A. E., Gorial, I. I. & Humaidi, A. J. An improved ensemble-based cardiovascular disease detection system with chi-square feature selection. *Computers* **13**, 126 (2024).
3. Shah, A., Ramanathan, A., Hayot-Sasson, V. & Stevens, R. Causal discovery and optimal experimental design for genome-scale biological network recovery. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, 1–11 (2023).
4. Gao, X.-E., Hu, J.-G., Chen, B., Wang, Y.-M. & Zhou, S.-B. Causal discovery approach with reinforcement learning for risk factors of type ii diabetes mellitus. *BMC Bioinformatics* **24**, 296 (2023).
5. Kaltenpoth, D. & Vreeken, J. Nonlinear causal discovery with latent confounders. In *International Conference on Machine Learning*, 15639–15654 (PMLR, 2023).
6. Bronstein, M., Meyer-Kalos, P., Vinogradov, S. & Kummerfeld, E. Causal discovery analysis: a promising tool for precision medicine. *Psychiatr. Ann.* **54**, e119–e124 (2024).
7. Mbogu, H. M. & Nicholson, C. D. Data-driven root cause analysis via causal discovery using time-to-event data. *Comput. Ind. Eng.* 109974 (2024).
8. Moreau, C. et al. Dynamic personalized prediction of the individual liver-related risk after sustained viral response in hcv patients. *J. Viral Hepatitis* **30**, 567–577 (2023).
9. Bales, M. et al. Pathways between risk/protective factors and maternal postnatal depressive symptoms: the elfe cohort. *J. Clin. Med.* **12**, 3204 (2023).
10. Arlegui, H. *et al.* Impact of the first wave of the covid-19 pandemic on the treatment of psoriasis with systemic therapies in france: Results from the psobioteq cohort. In *Annales de Dermatologie et de Vénéréologie*, vol. 150, 101–108 (Elsevier, 2023).
11. Assaad, C. K., Ez-Zejjari, I. & Zan, L. Root cause identification for collective anomalies in time series given an acyclic summary causal graph with loops. In *International Conference on Artificial Intelligence and Statistics*, 8395–8404 (PMLR, 2023).
12. Ferreira, S. & Assaad, C. K. Identifiability of direct effects from summary causal graphs. *In Proceedings of the AAAI Conference on Artificial Intelligence* **38**, 20387–20394 (2024).
13. Rehak, J., Youssef, S. & Beyerer, J. Root cause analysis using anomaly detection and temporal informed causal graphs. In *ML4CPS–Machine Learning for Cyber-Physical Systems* (2024).
14. Dalloo, A. M. & Humaidi, A. J. Optimizing machine learning models with data-level approximate computing: the role of diverse sampling, precision scaling, quantization and feature selection strategies. *Res. Eng.* **24**, 103451 (2024).
15. Lu, C., Wu, Y., Hernández-Lobato, J. M. & Schölkopf, B. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations* (2021).
16. Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference* (Elsevier, 2014).
17. Ling, Z., Yu, K., Zhang, Y., Liu, L. & Li, J. Causal learner: A toolbox for causal structure and markov blanket learning. *Pattern Recogn. Lett.* **163**, 92–95 (2022).
18. Ling, Z. et al. A light causal feature selection approach to high-dimensional data. *IEEE Trans. Knowl. Data Eng.* **35**, 7639–7650 (2022).
19. Srivastava, A., Chockalingam, S. P. & Aluru, S. A parallel framework for constraint-based bayesian network learning via markov blanket discovery. *IEEE Trans. Parallel Distrib. Syst.* **34**, 1699–1715 (2023).
20. Yu, K. et al. Causality-based feature selection: Methods and evaluations. *ACM Comput. Surveys (CSUR)* **53**, 1–36 (2020).
21. Bishop, C. M. & Frey, B. J. (eds.). *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research (PMLR).
22. Wu, X., Jiang, B., Yu, K., Miao, c. & Chen, H. Accurate markov boundary discovery for causal feature selection. *IEEE Trans. Cybern.* **50**, 4983–4996. https://doi.org/10.1109/TCYB.2019.2940509 (2020).
23. Khan, W., Kong, L., Noman, S. M. & Brekhna, B. A novel feature selection method via mining markov blanket. *Appl. Intell.* **53**, 8232–8255 (2023).
24. Yu, K., Liu, L., Li, J. & Chen, H. Mining markov blankets without causal sufficiency. *IEEE Trans. Neural Netw. Learn. Syst.* **29**, 6333–6347. https://doi.org/10.1109/TNNLS.2018.2828982 (2018).
25. Liu, X.-Q. & Liu, X.-S. Markov blanket and markov boundary of multiple variables. *J. Mach. Learn. Res.* **19**, 1–50 (2018).
26. Yu, K., Liu, L. & Li, J. A unified view of causal and non-causal feature selection. *ACM Trans. Knowl. Discov. Data (TKDD)* **15**, 1–46 (2021).

27. Cavique, L. Causality: The next step in artificial intelligence. In *Philosophy of Artificial Intelligence and Its Place in Society*, 1–17 (IGI Global, 2023).
28. Ferreira, A. J. & Figueiredo, M. A. Efficient feature selection filters for high-dimensional data. *Pattern Recogn. Lett.* **33**, 1794–1804 (2012).
29. Wang, N., Liu, H., Zhang, L., Cai, Y. & Shi, Q. Loose-to-strict markov blanket learning algorithm for feature selection. *Knowl.-Based Syst.* **283**, 111216 (2024).
30. Wu, X., Jiang, B., Wang, X., Ban, T. & Chen, H. Feature selection in the data stream based on incremental markov boundary learning. *IEEE Trans. Neural Netw. Learn. Syst.* (2023).
31. Wang, L. et al. A survey of causal discovery based on functional causal model. *Eng. Appl. Artif. Intell.* **133**, 108258 (2024).
32. Guo, X., Yu, K., Cao, F., Li, P. & Wang, H. Error-aware markov blanket learning for causal feature selection. *Inf. Sci.* **589**, 849–877 (2022).
33. Rodrigues de Morais, S. & Aussem, A. A novel scalable and data efficient feature subset selection algorithm. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 298–312 (Springer, 2008).
34. Ling, Z. et al. Bamb: A balanced markov blanket discovery approach to feature selection. *ACM Trans. Intell. Syst. Technol. (TIST)* **10**, 1–25 (2019).
35. Cheng, L. et al. Evaluation methods and measures for causal learning algorithms. *IEEE Trans. Artif. Intell.* **3**, 924–943 (2022).
36. Frank, A. Uci machine learning repository. http://archive.ics.uci.edu/ml (2010).
37. Statnikov, A., Tsamardinos, I., Dosbayev, Y. & Aliferis, C. F. Gems: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *Int. J. Med. Informatics* **74**, 491–503 (2005).
38. Zhao, Z. et al. Advancing feature selection research. *ASU feature selection repository* 1–28 (2010).

Acknowledgements
I would like to express my gratitude to all the co-authors for their valuable contributions and the time they dedicated to our discussions. We are also grateful to Fuzhou University of International Studies and Trade for their support and collaboration in this work.

Author contributions
Waqar Khan contributed to conceptualization, methodology, software development, formal analysis, data curation, validation, funding, and original draft preparation. Brekhna Brekhna participated in formal analysis, investigation, supervision, and original draft writing. Jianqiong Huang, Yajun Xie, Muhammad Suhail Shaikh, Muhammad Sadiq Hassan Zada, and Yifan Zheng contributed to formal analysis and manuscript review and editing. All authors have reviewed and approved the final version of the manuscript.

Funding
This work is supported by Fujian Provincial Natural Science Foundation of China No. 2024J08242 and No. 2024J01982.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to W.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.