Online Streaming Features Selection via Markov Blanket

Waqar Khan 1, Lingfu Kong 1,*, Brekhna Brekhna 2,3, Ling Wang 1 and Huigui Yan 1

1 School of Information Science and Engineering, Yanshan University, Qinhuangdao 06604, China; wzafarkhan89@gmail.com (W.K.); wangling@stumail.ysu.edu.cn (L.W.); rocky_yhg@163.com (H.Y.)
2 School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China; wj00012@sdufe.edu.cn or brekhna87@gmail.com
3 Department of Computer Science, Shaheed Benazir Bhutto University, Peshawar 25000, Pakistan
* Correspondence: lfkong@ysu.edu.cn

Abstract: Streaming feature selection has always been an excellent method for selecting the relevant subset of features from high-dimensional data and overcoming learning complexity. However, little attention is paid to online feature selection through the Markov Blanket (MB). Several studies based on traditional MB learning presented low prediction accuracy and used fewer datasets as the number of conditional independence tests is high and consumes more time. This paper presents a novel algorithm called Online Feature Selection Via Markov Blanket (OFSVMB) based on a statistical conditional independence test offering high accuracy and less computation time. It reduces the number of conditional independence tests and incorporates the online relevance and redundant analysis to check the relevancy between the upcoming feature and target variable T, discard the redundant features from Parents-Child (PC) and Spouses (SP) online, and find PC and SP simultaneously. The performance OFSVMB is compared with traditional MB learning algorithms including IAMB, STMB, HITON-MB, BAMB, and EEMB, and Streaming feature selection algorithms including OSFS, Alpha-investing, and SAOLA on 9 benchmark Bayesian Network (BN) datasets and 14 real-world datasets. For the performance evaluation, F1, precision, and recall measures are used with a significant level of 0.01 and 0.05 on benchmark BN and real-world datasets, including 12 classifiers keeping a significant level of 0.01. On benchmark BN datasets with 500 and 5000 sample sizes, OFSVMB achieved significant accuracy than IAMB, STMB, HITON-MB, BAMB, and EEMB in terms of F1, precision, recall, and running faster. It finds more accurate MB regardless of the size of the features set. In contrast, OFSVMB offers substantial improvements based on mean prediction accuracy regarding 12 classifiers with small and large sample sizes on real-world datasets than OSFS, Alpha-investing, and SAOLA but slower than OSFS, Alpha-investing, and SAOLA because these algorithms only find the PC set but not SP. Furthermore, the sensitivity analysis shows that OFSVMB is more accurate in selecting the optimal features.

Keywords: Bayesian network; Markov blanket; streaming feature; feature selection; big data; conditional independence test; sensitivity analysis

1. Introduction

In machine learning, several feature selection algorithms are essential for processing high-dimensional data. The optimal feature (feature/variable/node are same) set for a target variable T is the Markov blanket (MB) [1], composed of Parents-Child (PC), Spouses (SP), and direct causes and effects between them, as shown in Figure 1.

There are two major approaches for feature selection, i.e., traditional-based and streaming-based. Traditional-based MB discovery (discovery/learning are interchangeable in this article) assumes that all features are readily available from the beginning. This assumption, however, is often violated in real-world applications. For instance, for the problem of detecting Mars crater from high-resolution planetary images [2], it is impracticable to attain the complete feature set, which means to have a near-global coverage of
the Martian surface. Several MB discovery algorithms are proposed based on traditional concepts requiring all features and instances to be available before learning. Such algorithms include Incremental Association-Based Markov Blanket (IAMB) [3], HITON-MB (HITON-MB) [4], Simultaneous Markov Blanket (STMB) [5], Balanced Markov Blanket (BAMB) [6], and Efficient and Effective Markov Blanket (EEMB) [7]. Traditional-based MB discovery algorithms do not apply to real-world scenarios, such as users of the famous microblogging website Twitter who yield over 250 million tweets every single day, which includes words and abbreviations [8], personalized recommendations [9], malware scanning [10], and ecological inspection and analysis [11] with the quality of frequent features updates and dynamic space-changing [12]. A fascinating question is whether we should wait long for all features to become accessible before starting the learning process. We need a lot of computational efforts to build such features upfront. This raises an intriguing research challenge about constructing an effective feature selection process without knowing the entire feature space. However, the existing algorithms for Markov blanket discovery commonly assume that all features must be present in advance. Therefore to tackle such issues, streaming feature selection was introduced and applied to many applications such as biology, weather forecasting, transportation, stock markets, clinical research, etc. [8]. Several algorithms based on streaming features (SF) were proposed for real scenarios including Grafting [8], Alpha-investing (α−investing) [13], Scalable and Accurate Online Feature Selection (SAOLA) [14], and Online Streaming Feature Selection (OSFS) [15]. However, these algorithms only focus on obtaining PC sets and do not consider the Spouses, which causes them to lose the interpretability by ignoring the causal MB discovery.

Motivated by these observations and issues, this paper presents an Online Streaming Features Selection via Markov Blanket algorithm, based on a statistical conditional independence test. The features are no longer static but flow in one by one and are analyzed as it arrives. A null-conditional test motivated it to address the streaming features, feature relevance analysis to find the true positive PC and Spouses, and feature redundancy analysis to remove the false positive/irrelevant features.

The main contributions of this study are as follows:
- OFSVMB obtains Parent-Child and Spouse feature sets simultaneously and separates them in the MB set;
- OFSVMB reduces the impact of conditionally independent tests errors since it uses fewer conditionally independent tests to learn MB;
- Sensitivity analysis of OFSVMB using three different parameters α values concerning the Rate of Instance |RI| to analyze the performance with small and large sample sizes;
- Performance evaluation of OFSVMB algorithm on benchmark BN and real-world datasets.

The remainder of the paper is organized as follows: The related work is presented in Section 2. The preliminaries is discussed in Section 3. The proposed algorithm is presented in Section 4. The experimental findings are discussed in Section 5, and the paper is concluded in Section 6.
2. Related Work

In traditional MB discovery techniques, features must be present before learning begins. Different algorithms developed for MB learning, which is based on traditional concepts such as Incremental Association-Based Markov Blanket (IAMB) [16], Max-Min Markov Blanket (MMMB) [17], HITON-MB (HITON-MB) [18], Simultaneous Markov Blanket (STMB) [19], Iterative Parent-Child-based MB (IPCMB) [19], Balanced Markov Blanket (BAMB) [6], and Efficient and Effective Markov Blanket (EEMB) [7]. While MB is learning the Parents-Child and Spouses of the target feature, T cannot differentiate by the IAMB [16] algorithm. Additionally, when the sample size of the dataset is not quite large, the IAMB can not faithfully discover the Markov blanket of the target feature T. The Min-Max Markov blanket (MMMB) uses the Divide and Conquer strategy to minimize the size of data samples [17]. It needs to split the dilemma of finding MB into two sub-dilemmas to find the Parents-Child and the Spouses. MMMB has been modified to build HITON-MB, which excludes false-positive features from the Parents-Child set quickly by interleaving the growing and shrinking phases [18]. Under the Markov blanket assumption, MMMB [17] and HITON-MB [18] were conceptually flawed and needed a new mechanism for accurate MB discovery. The Iterative Parent-Child-based search of Markov Blanket (IPCMB) algorithm uses the same PC algorithm as PCMB [19] to identify the PC set and increases the efficiency without losing accuracy [19]. However, the symmetry constraint check makes this algorithm computationally slow. STMB [19] also followed the technique as IPCMB [19] for Parents-Child exploration. However, it consumes more time and avoids symmetry checks.

Moreover, the online feature selection methods receive features one by one dynamically. These methods include grafting [8], Alpha-investing ($\alpha$-investing) [8], Scalable and Accurate Online Feature Selection (SAOLA) [20], and Online Streaming Feature Selection (OSFS) [21]. Grafting was the first algorithm designed by Perkins and Theiler. It attempts the streaming features selection technique, which is a stage-wise technique for gradient descent. Alpha-investing used $p$-value and threshold to control the feature selection process and decide to select relevant features and remove redundant ones [8]. The benefit of alpha-investing is to handle features sets of unknown sizes even up to infinity, but it fails to investigate redundant features, causing an unpredictable and low prediction accuracy. SAOLA [20] examines two features simultaneously and analyzes redundancy under a single scenario. It fails to find an optimum relevance threshold value to remove all redundant features. In contrast, OSFS [21] removes unnecessary features which are not relevant/associated with the target feature, T, using conditional independence. It uses two steps to achieve the Parents-Child relevant to the target feature T. The first step analyzes the online relevance, and the second step analyzes online redundancy. It identifies the approximate MB without Spouses. The methods of online feature selection such as Alpha-investing, SAOLA, and OSFS only identify Parents-Child (PC) features and avoid the Spouses (SP) and the causality neglect by these algorithms. Based on online feature selection and causality, the Causal Discovery From Streaming Feature (CDFSf) is used [22]. The symmetric Causal Discovery From Streaming Feature (S-CDFSf) [22] is developed, which uses conditional independence test to identify the relevant features of the target feature, T, based on streaming, which belongs to the Parents-Child and Spouses.

3. Preliminaries

In this segment, the MB discovery through streaming features is defined. In addition, the specific aspects are discussed in detail. Table 1 summarized the notations used in this paper based on the new definitions.
**Table 1. Summary of Notation.**

| Notations | Mathematical Meanings |
|-----------|-----------------------|
| R         | A feature set under streaming features |
| S         | Conditional feature set within R |
| T         | Target attribute |
| C         | Over R, a DAG (Directed Acyclic Graph) |
| P         | Over R, a JPD (Joint Probability Distribution) |
| f         | False positive |
| X, Y, Z, N, E or \(X_i, Y_i, Z_i\) | A feature/a variable/a node, which belong to R |
| \(t_i\)  | A time point of the \(X_i\) arriving feature |
| \(PC_T\) | A set of Parents-Children of T |
| \(CPC_T\) | A candidate feature set of \(PC_T\) |
| non-\(pc_T\) | Non-Parents-Child set of target T |
| \(X_i \perp Y_i\) | Given S, X and Y are independent |
| \(X_i \perp\!\perp Y_i\) | Given S, X and Y are dependent |
| \(SP_T\) | A set of Spouses of T |
| \(SP_T\{X\}\) | A Candidate feature set of Spouse |
| \(CSP_{T}\{X\}\) | T's Spouses in relation to T's child X |
| \(Des_T\) | Descendant of Target (T) |
| \(Pa_T\) | Parents of target T |
| \(\alpha\) | Significance levels of 0.01 and 0.05 |
| \(\rho\) | \(p\)-value return by statistical \(G^2\) test and Fisher's z-test |
| \(MB_T\) | Markov blanket (MB) of Node T |

**Definition 1** (Streaming Features \[15\]). *Involves a feature vector \(\{X_1, X_2, X_3, \ldots, X_i\}\) that streams in one by one over time \(\{t_1, X_{i+1}, X_{i+2}, \ldots, X_{i+n}\}\) while a number of training samples remain constant.*

**Definition 2** (Conditional Independence \[6\]). *Feature (variable) \(X_i\) is conditionally independent of feature (variable) \(Y_i\) given S, if and only if \(P(X_i|Y_i, S) = P(X_i|S)\).*

**Definition 3** (Strong relevant \[23\]). *A feature \(X_i \in R\) is strongly relevant to the target feature \(T\), if and only if \(\forall S \subseteq R \setminus \{X_i\}, s.t. P(X_i|S) \neq P(X_i|S, T)\).*

**Definition 4** (Irrelevant \[23\]). *A feature \(X_i \in R\) is irrelevant to a target feature \(T\), if and only if it is \(\forall S \subseteq R \setminus \{X_i\}, s.t. P(X_i|S, T) = P(S|T)\).*

**Definition 5** (Redundant \[24\]). *A feature \(X_i\) is redundant to the target variable \(T\), if and only if it is weakly relevant to target variable \(T\) and has a Markov blanket, \(MB(X_i)\), then it is a subset of the Markov blanket of \(MB_T\).*

**Definition 6** (Faith-fullness Condition \[25\]). *\(G\) denotes a Bayesian network, and \(P\) represents a joint probability distribution through feature set \(R\). So, \(G\) is devoted to \(P\) if \(P\) captures all. Only the conditional independence is among features in \(G\).*

**Definition 7** (V-structure \[26\]). *If there is no an arrow between feature (variable) \(X_i\) and feature (variable) \(Y_i\), and feature (variable) \(Z_i\) has two incoming arrows from \(X_i\) and \(Y_i\), respectively, then \(X_i, Z_i, and Y_i\) form a \(V\)-structure \(X_i \rightarrow Z_i \leftarrow Y_i\).*

**Definition 8** (D-Separation \[26\]). *A path \(D\) between a feature (variable) \(X_i\) and feature (variable) \(Y_i\) is D-separated by set of features (variables) \(S\), if and only if:

- \(D\) includes a chain \(X_i \leftarrow Z_i \rightarrow Y_i\) such that the middle one features \(Z_i\) is in \(S\).
- \(D\) includes a collider \(X_i \rightarrow Z_i \leftarrow Y_i\) such that the middle one feature \(Z_i\) is not in \(S\) and none of \(Z_i\) successors are in \(S\).*

A feature set \(S\) is said to be D-separated \(X_i\) and \(Y_i\), if and only if \(S\) jammed each path \(D\) from a feature \(X_i\) to a feature \(Y_i\).*
Theorem 1. In a faithful Bayesian Network, an MB of the target feature T, MBₜ, in a set R is an optimal set of features, composed of Parents, Children, and Spouses. All other features are not conditionally dependent of target feature T given MBₜ, ∀Xᵢ ∈ R \ (MBₜ ∪ T), s.t. Xᵢ ⊥⊥ T|MBₜ.

4. Online Feature Selection via Markov Blanket

This section presents the proposed algorithm for implementing the framework for feature selection with streaming features called Online features selection via a Markov blanket (OFSVMB).

4.1. Framework of OFSVMB

The framework of the Online Feature Selection via Markov blanket is shown in Table 2. Two conditional tests are used to check the association between features. The first is a statistical G² test (for discrete data) and the second is the statistical Fisher’s z-test (for continuous data). Ind(Xᵢ, T|S) = (Xᵢ ⊥⊥ T|S) (Equal(=) sign means they are same in this article) denotes a conditional independence test between a feature Xᵢ and the target feature (variable) T, given a subset S. While Dep(Xᵢ, T|S) = (Xᵢ ∥ T|S) represents a conditional dependence test between a feature Xᵢ and the target feature T, given a subset S.

Table 2. The OFSVMB framework.

1. Initialization:
The target T; Candidate Parents-Child set CPCₜ = ∅; non-Parents-Child features set non_p_cₜ = ∅; Candidate Spouse set CSPᵣ = ∅; MB = ∅;
2. Check relevancy between feature Xᵢ arrived at time tᵢ and target feature T through null conditional independence test; if Dep(Xᵢ, T|∅), add in CPCₜ; if not, remove non-relevant feature through Proposition 1 and add in non_p_cₜ set, then enter step 3.
3. After null-conditional independence, check for relevance and redundant features, e.g., Dep(Xᵢ, T|S), through Proposition 2. If it is relevant, add to CPCₜ and jointly find a spouse and add in CSPᵣ. Otherwise, add it to the non_p_cₜ set.
   • It looks for non-MB successors in CPCₜ set through Theorem 2.
   • Discard it from CPCₜ set if it is not dependent on target T.
4. After step 3, find Spouse in non_p_cₜ set, which is redundant in step 2 and 3, and simultaneously remove false positive spouse from spouse set through Proposition 3.
5. Steps 2–4 are repeated until no features remain.
6. Output: Selected MB of target T.

For a new incoming feature, the OFSVMB algorithm performs the Null conditional independence analysis [26], relevance analysis, and redundancy analysis. The pseudocode of OFSVMB is shown in Algorithm 1. In Algorithm 1 RecogNC, line 5 performs null independence analysis [26], relevance analysis, and redundancy analysis. The pseudocode

Proposition 1. Using null conditional independence, check the feature Xᵢ relevancy or irrelevancy with target feature T.

Proof of Proposition 1. Assuming that Xᵢ ∈ R and Yᵢ ∈ R, the following hold:
Xᵢ ⊥⊥ Yᵢ|[] ⇒ P(Xᵢ|Yᵢ[]) = P(Xᵢ) ⇒ P(Xᵢ|Yᵢ) = P(Xᵢ) ⇒ P(Xᵢ,Yᵢ) = P(Xᵢ)P(Yᵢ) ⇒ P(Xᵢ,Yᵢ) = P(Xᵢ)P(Yᵢ) ⇒ Xᵢ ⊥⊥ Yᵢ □

Therefore, (Xᵢ ⊥⊥ Yᵢ[]) represents that Xᵢ and Yᵢ are not relevant to each other.

Through the relevancy analysis based on Proposition 2, the OFSVMB analyzes features and adds them to the candidate Parents-Child set CPCₜ and the candidate Spouse set CSPᵣ, which are the candidate set of the Parents-Child and Spouse set. If feature Xᵢ is related to
Theorem 2. via Markov blanket, given in Algorithm 1. The proposed algorithm can derive the MB Y given in V-structure.

Proof of Proposition 3. is removed from CSP. Proof. The proof is as follows: If ∃X_i ∈ CPC_T, i.e., X_i ⊥⊥ T|S, then X_i ∈ CPC_T.

Proof. The proof is as follows: If ∃X_i ∈ CPC, s.t. X_i ⊥⊥ T|S, if feature X_i is related to target feature T under R condition by Theorem 1, then add X_i to CPC_T.

Based on Theorem 2 and Proposition 3, using the redundancy analysis, the OFSVMB removes false-positive features separately from the candidate Parent-Child set CPC_T and the candidate Spouse set CSP_T, which are the candidate sets of the Parents-Child and Spouse set. However, it also looks for non-MB descendant features in CPC_T and discards them if they have been through Theorem 2, which makes the OFSVMB different from the other existing algorithms. Removing false positive Spouses through redundant analysis if Y_i ∈ CPC_T, if Y_i is not dependent to the target feature (variable) T, giving subset S, then Y_i is removed from CPC_T. By removing the irrelevant Spouse such that N ∈ CPC_T and E ∈ CSP_T\{N\}, if E is not relevant to target feature T under the condition of S ∪ N, then E is removed from CSP_T\{N\}.

Theorem 2. (PC recognizes false-positives): Only successors of the target T, standing for Des_T, make up false-positives f ∈ PC_T.

Proof. CPC_T is composed of a whole PC and a few false positives. We show that f ⊆ Des_T. CPC_T is a candidate super-set of all true positive PCs because it should hold the true positive PCs. After an exhaustive search for the PC set, the entire parents set of target T is represented by Pa_T. According to Definition 5, all non-successor nodes (features) are independent of target T provided Pa_T. f ⊥⊥ T|Pa_T is any non-descendant node if f ∈ f is any non-successor nodes (feature). As a result, f will be omitted from CPC_T due to the Markov condition f ⊆ Des_T.

Proposition 3. (Remove false-positives from a Spouse). In a BN, R is a feature set, assuming that X_i is neighboring to Y_i, Y_i is neighboring to T, and X_i is not neighboring to T (e.g., X_i → Y_i). However, once feature M enters CSP_T\{Y_i\}, for any existing feature X in CSP_T\{Y_i\} \ N, ∃S ⊆ CPC_T ∪ CSP_T\{Y_i\} \ X_i, X_i ⊥⊥ T|S ∪ \{Y_i\}, then X is a false-positives and removed from CSP_T\{Y_i\}.

Proof of Proposition 3. V-structure (X_i → Y_i ← T) illustrates that if X_i is target T’s Spouse and Y_i is the mutual Child, there occurs a subset S ⊆ R \ \{X_i, T, Y_i, N\} so that features X and T are independent given feature S. Still, they are dependent given S ∪ Y_i. If there is another feature, M exists to block the path between X_i and Y_i. M has a direct effect on Y_i that is not satisfied by V-structure (X_i → Y_i ← T), so in this condition, X is removed from CSP_T\{Y_i\} and is considered as a false-positives Spouse.

4.2. The Proposed OFSVMB Algorithm and Analysis

This paper segment explains the proposed algorithm for Online Feature Selection via Markov blanket, given in Algorithm 1. The proposed algorithm can derive the MB_T by deleting all redundant features. The OFSVMB algorithm based on null-conditional independence test, relevance analysis, and redundant analysis. First, the null-conditional independence [26] (line 5) of Algorithm 1 is used to tackle the streaming feature. It analyze
the new feature $X_i$ arrived at time $t_i$. If $X_i$ is dependent of target $T$ given empty set, (the empty set is equal to $\emptyset$), then include feature $X_i$ to $CPC_T$; otherwise, add $X_i$ to non_pC_T. After null conditional independence analysis, check if either feature $X_i$ is a true positive PC (line 6–9) given subset $S$. If it is not a true positive, remove from $CPC_T$ (line 10); otherwise, analyze whether it is a candidate Spouse CSP_T\{X_i\} (line 11–19).

Algorithm 1 OFSVMB.

1: Require: Data $D$; target node, $T$; $X$
2: Ensure: Markov blanket of target node $T$.
3: $CPC_T \leftarrow \emptyset$; $CSP_T \leftarrow \emptyset$; non_pC_T \leftarrow \emptyset;
4: repeat
5: \hspace{1em} $[CPC_T, non_pC_T] \leftarrow$ RecogNC $(T, X_i, D)$
6: \hspace{2em} if $Dep(X_i, T|S)$ then
7: \hspace{3em} $CPC_T \leftarrow CPC_T \cup \{X_i\}$
8: \hspace{3em} for every $Y_i \in CPC_T$ do
9: \hspace{4em} if $Y_i \perp\!\!\!\!\!\perp T|S$ then
10: \hspace{5em} $CPC_T \leftarrow CPC_T \setminus \{Y_i\}$
11: \hspace{5em} $CSP_T\{Y_i\} \leftarrow \emptyset$
12: \hspace{5em} $Sep_T\{Y_i\} \leftarrow S$
13: \hspace{5em} for every $Z_i \in CPC_T$ do
14: \hspace{6em} if $Y_i \not\perp\!\!\!\!\!\perp T|Sep_T\{Y_i\} \cup \{Z_i\}$ then
15: \hspace{7em} $CSP_T\{Z_i\} \leftarrow CSP_T\{Z_i\} \cup \{Y_i\}$
16: \hspace{6em} end if
17: \hspace{5em} end for
18: \hspace{3em} end if
19: \hspace{2em} end for
20: if $CPC_T$ & CSP_T\{Z_i\} is nonempty then
21: \hspace{2em} $M \leftarrow CPC_T$
22: \hspace{2em} for every $X_i \in M$ do
23: \hspace{3em} if $X_i \not\perp\!\!\!\!\!\perp T|CPC_T \cup CSP_T\{X_i\}$ then
24: \hspace{4em} $CPC_T \leftarrow CPC_T \setminus \{X_i\}$
25: \hspace{3em} end if
26: \hspace{2em} end for
27: \hspace{2em} end if
28: else
29: \hspace{2em} non_pC_T \leftarrow non_pC_T \cup \{X_i\}
30: \hspace{2em} if $CPC_T$ & non_pC_T is nonempty then
31: \hspace{3em} for every $N \in CPC_T$ do
32: \hspace{4em} for every $X_i \in non_pC_T$ do
33: \hspace{5em} if $X_i \not\perp\!\!\!\!\!\perp T|Sep_T\{Y_i\} \cup \{N\}$ then
34: \hspace{6em} $CSP_T\{N\} \leftarrow CSP_T\{N\} \cup \{X_i\}$
35: \hspace{6em} for every $E \in CSP_T\{N\}$ do
36: \hspace{7em} for $S \subset \{CPC_T \cup CSP_T\{N\} \setminus \{E\}\}$ do
37: \hspace{8em} if $E \not\perp\!\!\!\!\!\perp T|S \cup \{N\}$ then
38: \hspace{9em} $CSP_T\{N\} \leftarrow CSP_T\{N\} \setminus \{E\}$
39: \hspace{8em} end if
40: \hspace{7em} end for
41: \hspace{6em} end if
42: \hspace{5em} end for
43: \hspace{4em} end if
44: \hspace{3em} end for
45: \hspace{2em} end if
46: \hspace{2em} end if
47: until no more features are left
48: Output $MB_T$
OFSVMB then checks non-MB successors $X_i$ in the $CPC_T$ that may have several pathways to the target variable $T$. If $X_i$ and $T$ are independent, $X_i$ is removed from the $CPC_T$ (line 24). Algorithm 1 finds the Parents-Child (line 5–10) and computes the Spouses from the non_pC set that is discarded during the null-conditional independence step (line 5) and from false-positive PC (line 6–10).

Moreover, after checking the conditional independence test at (line 6), if the feature $X_i$ is not dependent on target feature $T$, it comes to (line 29), to identify the spouse from the non_pc set. If $X_i \not\perp \perp T|\{X_j\} \cup N$, include the CSP set $\{N\}$ (line 31–34). Through redundant analysis (line 35–45) check whether the selected spouse is redundant, if yes, then discard the false positive/redundant feature from CSP set.

### 4.3. Statistical Conditional Independence Terminology in OFSVMB

The conditional independence test is used to classify irrelevant and redundant features [15], denoted by the notations $Ind(T, X_i|S)$ and $Dep(T, X_i|S)$ in Algorithm 1. The $G^2$ test, equivalent to the $\chi^2$ test (discrete data), and Fisher’s $z$-test are used.

#### 4.3.1. Statistical $G^2$ Test for Discrete Data

The $G^2$ with three features (variables), $X_i, X_j$, and $X_k$, set $V_{ijk}^{abc}$ as the number of checks satisfying $X_i = x, X_j = y,$ and $X_k = z$ in a dataset. $V_{ij}^{xy}, V_{jk}^{xz},$ and $V_{ik}^x$ are all described in the same way. If $X_i$ and $X_j$ are conditionally not dependent given $X_k$, thus, in Equation (1) below, the $G^2$ test is shown:

$$
G^2 = 2 \sum_{x, y, z} V_{ijk}^{abc} \ln \frac{V_{ijk}^{abc}}{V_{ij}^{xy} V_{jk}^{xz} V_{ik}^x}.
$$

With sufficient degrees of freedom, $G^2$ is asymptotically distributed as $\chi^2$. In general, when checking the conditional independence of $X_i$ and $X_j$ given $S$, the amount of degrees of freedom $df$ used during the test is measured as:

$$
df = (c_i - 1)(c_j - 1) \prod_{c_k \in S} c_k \tag{2}
$$

where, in Equation (2), $c_i$ represents the number of distinct values of $X_i$.

#### 4.3.2. Statistical Fisher’s $z$-Test for Continuous Data

Fisher’s $z$-test, on the other hand, calculates the degree of correlation between features as given in Equation (3). After the feature subset $S$ is provided, the partial correlation coefficient $r(X_i, T|S)$ between the feature $X_i$ and the target variable $T$ is expressed in the Gaussian distribution $N(\mu, \Sigma)$ [14]:

$$
r_{(X_i, T|S)} = \frac{\hat{r}(X_i, T) - \hat{r}(X_i, T)\hat{r}(S,T)}{\sqrt{1 - \hat{r}^2(X_i, T)}} \sqrt{1 - \hat{r}^2(S,T)} \tag{3}
$$

Under the null hypothesis of conditional independence between the feature $X_i$ and the target variable $T$ of the current feature subset $S$, $r_{(X_i, T|S)} = 0$, according to Fisher’s $z$-test. Assume that $\alpha = \{0.01, 0.05\}$ is a given level of significance and $p$ is the $p$-value obtained by Fisher’s $z$-test.

Supposing $p > \alpha$, $X_i$ and target variable $T$ are not related to each other when the subset $S$ is given, according to the null hypothesis of the conditional independence of $X_i$ and target variable $T$. If $p \leq \alpha$, then $X_i$ and target variable $T$ are both relevant.

#### 4.4. Correctness of OFSVMB

OFSVMB produces the MB of the target feature $T$ truly and accurately, under the faithful assumption. According to Theorem 1, in the beginning, OFSVMB finds true
positive or relevant features which belong to the \( CPC_T \). It is a candidate Parents-Child set of target feature \( T \) using null-conditional independence and adds them in the candidate parent child (CPC) set, e.g., \( CPC_T \). In contrast, add the false positive or redundant feature in the non-Parents-Child set (line 5). Then, in the candidate Parents-Child (CPC) set, e.g., \( CPC_T \), OFSVMB searches for a false positive or redundant feature and discards it from the candidate parent-child (CPC) set (line 9) and (line 10). The \( CPC_T \) contains all relevant features dependent on target feature \( T \) given any subset \( S \in R \).

Additionally, OFSVMB finds the Spouse (line 13–19) of target \( T \) at the same time as looking for a redundant feature in the \( CPC_T \) set. Non-MB successors in \( CPC_T \) are discarded by conditioning on the true positive Spouse in \( CSP_T \{ X_i \} \), combined with the true positive PC in \( CPC_T \) in the joint set \( CPC_T \) and \( CSP_T \{ X_i \} \) (line 20–27). Suppose the feature \( X_i \) (line 6) is independent of target relevant feature \( T \), so the OFSVMB considers searching Spouse from the non-pc set (line 29–34) and simultaneously removes the redundant feature from \( CSP_T \) through redundant analysis (line 35–38). At last, if there is no feature left, OFSVMB finds the Markov blanket (MB) of target feature \( T \) at (line 48), which is the union of true positive \( PC_T = CPC_T \) and true positive \( SP_T = CSP_T \).

4.5. Time Complexity Analysis

The time complexity, as presented in Table 3 of the state-of-the-art MB discovery algorithms, counts on how many CI tests have been used in the algorithm. OFSVMB identifies the MB of target feature \( T \) through online relevance and redundancy analysis. It is assumed that \( R \) represents the total number of features that appeared with time \( t \), where \( R' \) is the number of features relevant to target feature \( T \) in \( R \). The remaining number of features in \( R \), not relevant to target \( T \), is represented by \( R'' \). \( CSP_T \{ X_i \} \) represents the Candidate Parents-Child set, and \( CSP_T \{ X_i \} \) represents a subset of the Spouses of target \( T \) with regards to target \( T \)’s child \( X_i \). When the feature \( X_i \) appears at time \( t \), the OFSVMB time complexity is explained below: The null-conditional independence has a time complexity of \( O(1) \). The Parents-Child identification takes \( O(R(R')|CPC_T|k^{CPC_T}) \) and the discovery of Candidate Spouses is \( O(R''|CPS_T \{ X_i \}|K^{CPC_T}) \), where \( k \) is the maximum limit of a conditioning set that might increase. The OFSVMB is based on streaming discard redundant features in the streaming scenario, and both CPC and CSP become smaller. The approximate time complexity of the proposed OFSVMB algorithm is \( O(|R|2^C) \), where \( C \) is the is equal to \( CPC_T \) and \( CSP_T \). The OFSVMB is somehow more efficient than the state-of-the-art because it handles features in real-time scenarios.

Table 3. Time complexity of Markov blanket (MB) discovery Algorithm.

| Algorithm   | Time Complexity                                      |
|-------------|------------------------------------------------------|
| IAMB        | \( O(|R|^2) \)                                        |
| STMB        | \( O(|R|^2R') \)                                      |
| HITON-MB    | \( O(2^{CPC_T}|R||CPC_T|) \)                          |
| BAMB        | \( O(|R|2^{CPC_T}) \)                                 |
| EEMB        | \( O(|R|2^{CPC_T}) \)                                 |
| OFSVMB      | \( O(|R|2^C) \)                                       |

The proposed OFSVMB and EEMB accuracy are comparable. However, OFSVMB performs the streaming feature selection where features come one by one inflow, so removing the redundant features from \( CPC_T \) and \( CSP_T \) jointly takes a few CI tests. The STMB uses a backward strategy during PC learning and identifies the separated PC set from all other subsets of \( R \) at any repetition. It makes the STMB slower than the proposed OFSVMB algorithm, as presented in Table 3.
5. Results and Discussion

In this segment, the results of the proposed OFSVMB are discussed in detail. The results are conducted through extensive experiments and comparing them with the traditional-based MB discovery algorithms such as Iterative Associative Markov Blanket (IAMB), Simultaneous MB (STMB), HITON-MB (HITON-MB), Balanced Markov Blanket (BAMB), an Efficient and Effective MB discovery (EEMB), and streaming-based algorithms such as Alpha-investing (α-investing), Scalable and Accurate Online Feature Selection (SAOLA), and Online Streaming Feature Selection (OSFS).

5.1. Datasets and Experiment Setup

The experimental results are computed on 9 benchmark BN datasets (https://pages.mtu.edu/lebrown/supplements/mmhc_paper/mmhc_index.html, accessed on 1 July 2021) of small and large sample-size, as shown in Table 4, and 14 real-world feature selection datasets as shown in Table 5. The real-world datasets are selected from different domains, such as sets from the UCI machine learning repository [27]; frequently studied public microarray (wdbc) [28], ionosphere, colon, arcene, leukemia, and madelon are from the NIPS 2003 feature selection competition [29]; lung and medical belongs to biomedical [30]; lymphoma, reged1, and marti1 [31,32]; and prostate-GE and sido0 [33,34].

The OFSVMB algorithm is implemented in Matlab R2017b. All the experimental work is conducted on Windows 10 with an Intel Core i5-6500U with 8 GB RAM. The two conditional independence (CI) tests, including the $G^2$ test (for discrete data) and the Fisher’s z-test (for continuous data) with the significance levels of 0.01 and 0.05 are used.

5.2. Evaluation Metrics

The performance of the proposed OFSVMB algorithm is evaluated using three evaluation metrics on the benchmark BN datasets. The evaluation metrics are as follows:

- **Precision**: The number of true positives in an algorithm’s output (e.g., features that belong to the true MB of a target feature in a DAG) divided by the number of features in the algorithm’s output yields this metric.
- **Recall**: This metric is calculated by dividing the number of true positives in the output by the number of true positives (the true MB of a target feature in a DAG).
• **F1 = 2 * ((Precision * Recall)/(Precision + Recall))**: The harmonic mean of the precision and recall is used to calculate the F1 score. In the best-case scenario, F1 = 1 if precision and recall are both excellent. In the worst-case scenario, F1 = 0.

Each benchmark BNs contains two groups of sample size, i.e., 500 and 5000.

**Results and Discussion on Benchmark BN**

The efficiency and efficacy of the proposed OFSVMB algorithm is computed and compared with other state-of-the-art traditional-based MB discovery algorithms, such as IAMB, STMB, HITON-MB, BAMB, and EEMB, on 9 benchmark BN datasets. The F1, precision, recall, and running time in seconds of OFSVMB and other state-of-the-art are shown in Table 6 for 500 samples and Table 7 for 5000 samples. The sign “/” represents the separation of significant level, e.g., on the left side the α = 0.01 and on the right side the α = 0.05, where α is the significant level. The F1, precision, and recall of sample sizes 500 and 5000 (a) shows significance levels of 0.01 and (b) shows significance levels of 0.05 in Figures 2 and 3, respectively. Moreover, Figure 4 shows the running time of OFSVMB and five other state-of-the-art algorithms with different sample sizes. In the figures, the x-axis are denoted as the number of datasets (see Table 4), and the y-axis represent the F1, precision, and recall, together with running time, respectively.

According to Table 6, OFSVMB is the most accurate and fastest algorithm among the other five state-of-the-art algorithms because it is based on streaming features, which compute the features in an online manner. Meanwhile, on Child, Child3, and Insurance datasets with a sample size of 500 and significance levels of 0.01 and 0.05, OFSVMB is accurate in terms of F1, precision, and recall compared to IAMB, STMB, HITON-MB, BAMB, and EEMB, while faster than STMB, HITON-MB, BAMB, and EEMB. STMB and BAMB do not perform the symmetry check, but they must conduct an exhaustive subset analysis for the PC learning, making these two algorithms relatively slow and inefficient. The OFSVMB is slower than IAMB on Child and Child3 datasets because in each iteration, IAMB uses the entire set of presently selected features as conditioning set to identify whether or not to add or remove a feature from the currently selected features. IAMB is computationally efficient on datasets with small sample sizes, as presented in Table 6 in bold and shown in Figures 2a,b and 4. On Child10 datasets, the OFSVMB is faster than other state-of-the-art algorithms, and IAMB is the second fastest, as shown in Table 6 in bold and shown in Figure 2a,b and the running time in Figure 4. Moreover, on the Child10 dataset, the HITON-MB’s accuracy is comparable with the OFSVMB in terms of precision with a significance level of 0.05, as shown in Table 6 in bold.

On the Alarm10 dataset with a sample size of 500, the EEMB is more accurate in terms of F1 and recall than the OFSVMB, IAMB, STMB, HITON-MB, and BAMB. At the same time, the OFSVMB is accurate than EEMB in terms of precision and the running time is faster as well, as presented in Table 6 in bold and shown in Figures 2a,b and 4. On extensive datasets such as Pig and Gene with a sample size of 500 using significance levels of 0.01 and 0.05, OFSVMB is accurate in terms of F1, precision, and recall compared to its rivals, as shown in Figure 2a,b, while it is faster than the state-of-the-art as presented in Table 6 in bold and in Figure 4. In a dense dataset such as Barley, the HITON-MB is more accurate because HITON-MB locates the target’s Spouses, and must find the PC of each feature in the target’s discovered PC set. This method drastically reduces the number of sample data required and increases MB discovery performance, especially when dealing with high-dimensional and small data samples. When the size of the PC set of the features within the target’s PC set is enormous, therefore, this type of technique is computationally costly. The HITON-MB accuracy is higher in terms of F1 and precision than IAMB, STMB, BAMB, EEMB, and OFSVMB, but at a significance level of 0.01, the OFSVMB has comparable accuracy with HITON-MB in terms of F1. Moreover, in terms of recall, the EEMB has higher accuracy than IAMB, STMB, HITON-MB, BAMB, and OFSVMB, but the OFSVMB has higher running time than others, as shown in Table 6 in bold and Figures 2a,b and 4. On Mildew, the OFSVMB shows higher accuracy than other algorithms. However, HITON-MB is more
accurate than the OFSVMB and the others such as IAMB, STMB, BAMB, and EEMB at a significance level of 0.05 in terms of precision. In contrast, at a significance level of 0.01, the OFSVMB is more accurate than HITON-MB, as shown in Table 6. The OFSVMB on the Mildew dataset runs faster than the state-of-the-art as presented in Table 6 in bold and shown in Figures 2a,b and 4.

Figure 2. F1, Precision, Recall using 9 benchmark BN datasets: (a) significance level = 0.01; (b) significance level = 0.05 with sample size of 500.

Figure 3. Precision, Recall, F1 using 9 benchmark BN datasets: (a) significance level = 0.01; (b) significance level = 0.05, with sample size of 5000.
Table 6. F1, Precision, Recall, and Running time (s) on small and large-size benchmark BN datasets with sample size = 500.

| Dataset | Algorithm | F1    | Precision | Recall | Time(s) |
|---------|-----------|-------|-----------|--------|---------|
| Child   | IAMB      | 0.78/0.67 | 0.86/0.67 | 0.77/0.77 | 0.13    |
|         | STMB      | 0.73/0.70 | 0.91/0.85 | 0.67/0.67 | 0.20    |
|         | HITON-MB  | 0.82/0.81 | 0.94/0.86 | 0.77/0.77 | 0.24    |
|         | BAMB      | 0.86/0.81 | 0.93/0.82 | 0.81/0.84 | 0.18    |
|         | EEMB      | 0.86/0.81 | 0.92/0.81 | 0.82/0.83 | 0.15    |
|         | OFSVMB    | 0.89/0.88 | 0.97/0.94 | 0.83/0.85 | 0.14    |
| Child3  | IAMB      | 0.65/0.70 | 0.70/0.70 | 0.61/0.71 | 0.42    |
|         | STMB      | 0.65/0.58 | 0.82/0.70 | 0.62/0.60 | 0.86    |
|         | HITON-MB  | 0.76/0.71 | 0.85/0.83 | 0.69/0.71 | 1.27    |
|         | BAMB      | 0.77/0.78 | 0.89/0.86 | 0.70/0.73 | 0.74    |
|         | EEMB      | 0.57/0.57 | 0.67/0.67 | 0.50/0.50 | 0.74    |
|         | OFSVMB    | 0.89/0.87 | 0.95/0.93 | 0.83/0.85 | 0.48    |
| Child10 | IAMB      | 0.60/0.62 | 0.75/0.75 | 0.51/0.53 | 2.11    |
|         | STMB      | 0.54/0.56 | 0.75/0.76 | 0.47/0.49 | 0.30    |
|         | HITON-MB  | 0.71/0.72 | 0.90/0.84 | 0.62/0.65 | 3.08    |
|         | BAMB      | 0.71/0.70 | 0.78/0.81 | 0.66/0.63 | 0.28    |
|         | EEMB      | 0.80/0.79 | 0.90/0.85 | 0.75/0.87 | 0.35    |
|         | OFSVMB    | 0.65/0.66 | 0.91/0.88 | 0.56/0.58 | 0.25    |
| Alarm10 | IAMB      | 0.55/0.56 | 0.79/0.85 | 0.43/0.42 | 0.17    |
|         | STMB      | 0.46/0.46 | 0.81/0.82 | 0.34/0.35 | 0.24    |
|         | HITON-MB  | 0.56/0.57 | 0.89/0.89 | 0.44/0.45 | 0.33    |
|         | BAMB      | 0.55/0.56 | 0.73/0.71 | 0.43/0.45 | 0.43    |
|         | EEMB      | 0.62/0.67 | 0.75/0.76 | 0.58/0.59 | 0.56    |
|         | OFSVMB    | 0.78/0.79 | 0.90/0.90 | 0.70/0.71 | 0.15    |
| Insurance | IAMB     | 0.74/0.75 | 0.78/0.80 | 0.72/0.72 | 11.07   |
|          | STMB      | 0.79/0.74 | 0.91/0.86 | 0.74/0.72 | 11.94   |
|          | HITON-MB  | 0.91/0.76 | 0.88/0.84 | 0.90/0.90 | 14.78   |
|          | BAMB      | 0.86/0.84 | 0.88/0.84 | 0.86/0.86 | 27.3    |
|          | EEMB      | 0.92/0.80 | 0.93/0.72 | 0.90/0.89 | 18.30   |
|          | OFSVMB    | 0.93/0.91 | 0.96/0.91 | 0.92/0.92 | 9.25    |
| Pig     | IAMB      | 0.63/0.68 | 0.60/0.62 | 0.68/0.68 | 20.09   |
|          | STMB      | 0.56/0.18 | 0.44/0.12 | 0.73/0.73 | 30.24   |
|          | HITON-MB  | 0.80/0.66 | 0.79/0.56 | 0.90/0.92 | 23.34   |
|          | BAMB      | 0.82/0.68 | 0.79/0.59 | 0.90/0.91 | 23.18   |
|          | EEMB      | 0.82/0.67 | 0.78/0.57 | 0.90/0.92 | 28.50   |
|          | OFSVMB    | 0.87/0.89 | 0.85/0.86 | 0.91/0.93 | 14.88   |
| Gene    | IAMB      | 0.34/0.35 | 0.72/0.75 | 0.24/0.23 | 0.18    |
|          | STMB      | 0.12/0.11 | 0.30/0.24 | 0.08/0.08 | 0.20    |
|          | HITON-MB  | 0.34/0.39 | 0.74/0.77 | 0.24/0.27 | 0.19    |
|          | BAMB      | 0.28/0.31 | 0.30/0.35 | 0.28/0.28 | 0.21    |
|          | EEMB      | 0.33/0.35 | 0.40/0.27 | 0.50/0.50 | 0.21    |
|          | OFSVMB    | 0.34/0.34 | 0.73/0.73 | 0.24/0.25 | 0.17    |
| Barley  | IAMB      | 0.30/0.32 | 0.79/0.75 | 0.19/0.21 | 0.11    |
|          | STMB      | 0.13/0.09 | 0.29/0.19 | 0.09/0.06 | 0.14    |
|          | HITON-MB  | 0.34/0.34 | 0.81/0.83 | 0.21/0.22 | 0.20    |
|          | BAMB      | 0.33/0.34 | 0.81/0.81 | 0.21/0.22 | 0.16    |
|          | EEMB      | 0.29/0.25 | 0.50/0.33 | 0.20/0.20 | 0.17    |
|          | OFSVMB    | 0.35/0.35 | 0.83/0.82 | 0.23/0.23 | 0.9     |
Table 7. F1, Precision, Recall, and Running time (s) on small and large-size benchmark BN datasets with sample size = 5000.

| Dataset | Algorithm | F1   | Precision | Recall | Time(s) |
|---------|-----------|------|-----------|--------|---------|
| Child   | IAMB      | 0.75/0.67 | 0.83/0.64 | 0.70/0.71 | 11.05   |
|         | STMB      | 0.82/0.73 | 0.85/0.73 | 0.80/0.77 | 69.21   |
|         | HITON-MB  | 0.89/0.83 | 0.94/0.84 | 0.87/0.87 | 41.33   |
|         | BAMB      | 0.93/0.86 | 0.96/0.83 | 0.91/0.91 | 23.25   |
|         | EEMB      | 0.93/0.88 | 0.95/0.82 | 0.93/0.95 | 17.15   |
|         | OFSVMB    | 0.95/0.90 | 0.97/0.89 | 0.98/0.99 | 11.13   |
| Child3  | IAMB      | 0.79/0.71 | 0.75/0.71 | 0.79/0.72 | 5.29    |
|         | STMB      | 0.87/0.75 | 0.90/0.79 | 0.88/0.87 | 9.13    |
|         | HITON-MB  | 0.89/0.83 | 0.94/0.84 | 0.87/0.87 | 38.54   |
|         | BAMB      | 0.60/0.59 | 0.61/0.65 | 0.63/0.64 | 14.2    |
|         | EEMB      | 0.70/0.70 | 0.67/0.67 | 0.74/0.74 | 12.06   |
|         | OFSVMB    | 0.95/0.91 | 0.97/0.91 | 0.95/0.95 | 8.50    |
| Child10 | IAMB      | 0.76/0.75 | 0.78/0.75 | 0.75/0.75 | 8.95    |
|         | STMB      | 0.77/0.80 | 0.81/0.90 | 0.71/0.75 | 47.33   |
|         | HITON-MB  | 0.87/0.86 | 0.88/0.85 | 0.85/0.87 | 9.85    |
|         | BAMB      | 0.75/0.79 | 0.83/0.85 | 0.70/0.74 | 11.28   |
|         | EEMB      | 0.79/0.78 | 0.89/0.85 | 0.77/0.77 | 9.02    |
|         | OFSVMB    | 0.91/0.89 | 0.97/0.91 | 0.87/0.88 | 8.86    |
| Alarm10 | IAMB      | 0.60/0.61 | 0.61/0.61 | 0.60/0.61 | 8.05    |
|         | STMB      | 0.65/0.65 | 0.92/0.90 | 0.54/0.56 | 45.62   |
|         | HITON-MB  | 0.73/0.74 | 0.93/0.93 | 0.63/0.64 | 12.64   |
|         | BAMB      | 0.62/0.63 | 0.87/0.88 | 0.50/0.52 | 25.46   |
|         | EEMB      | 0.79/0.78 | 0.89/0.85 | 0.75/0.70 | 11.86   |
|         | OFSVMB    | 0.90/0.83 | 0.94/0.94 | 0.77/0.75 | 7.90    |
| Insurance | IAMB    | 0.63/0.61 | 0.62/0.62 | 0.65/0.70 | 15.22   |
|         | STMB      | 0.38/0.15 | 0.28/0.38 | 0.32/0.38 | 15.94   |
|         | HITON-MB  | 0.64/0.72 | 0.75/0.82 | 0.76/0.85 | 17.78   |
|         | BAMB      | 0.96/0.84 | 0.93/0.74 | 1.00/1.00 | 22.2    |
|         | EEMB      | 0.96/0.87 | 0.94/0.79 | 1.00/1.00 | 18.27   |
|         | OFSVMB    | 0.90/0.90 | 0.90/0.91 | 0.88/0.89 | 15.07   |
| Pig     | IAMB      | 0.60/0.60 | 0.53/0.53 | 0.89/0.89 | 24.01   |
|         | STMB      | 0.30/0.11 | 0.20/0.20 | 0.75/0.78 | 35.24   |
|         | HITON-MB  | 0.83/0.68 | 0.77/0.57 | 0.85/0.85 | 30.34   |
|         | BAMB      | 0.82/0.69 | 0.76/0.59 | 0.94/0.94 | 26.03   |
|         | EEMB      | 0.82/0.71 | 0.77/0.61 | 0.94/0.94 | 24.50   |
|         | OFSVMB    | 0.93/0.91 | 0.87/0.88 | 0.95/0.95 | 23.09   |
| Gene    | IAMB      | 0.35/0.36 | 0.63/0.65 | 0.25/0.25 | 9.70    |
|         | STMB      | 0.35/0.35 | 0.70/0.69 | 0.27/0.28 | 10.15   |
|         | HITON-MB  | 0.54/0.54 | 0.75/0.75 | 0.42/0.42 | 9.89    |
|         | BAMB      | 0.26/0.24 | 0.23/0.21 | 0.29/0.28 | 17.21   |
|         | EEMB      | 0.67/0.63 | 0.50/0.22 | 1.00/1.00 | 15.63   |
|         | OFSVMB    | 0.48/0.49 | 0.78/0.77 | 0.34/0.35 | 9.19    |
| Barley  | IAMB      | 0.48/0.49 | 0.60/0.63 | 0.41/0.41 | 9.87    |
|         | STMB      | 0.21/0.20 | 0.39/0.38 | 0.15/0.14 | 11.45   |
|         | HITON-MB  | 0.42/0.40 | 0.77/0.76 | 0.41/0.41 | 12.09   |
|         | BAMB      | 0.16/0.17 | 0.41/0.41 | 0.11/0.11 | 14.72   |
|         | EEMB      | 0.29/0.27 | 0.19/0.18 | 0.60/0.60 | 9.89    |
|         | OFSVMB    | 0.53/0.53 | 0.77/0.75 | 0.42/0.42 | 9.66    |
On small Child and Child3 datasets with a sample size of 5000 using significance levels of 0.01 and 0.05, OFSVMB is more accurate in terms of F1, precision, and recall compared to others, as presented in Table 7 in bold and shown in Figure 3a,b. However, on the Child dataset, the OFSVMB runs faster than the HITON-MB, STMB, BAMB, and EEMB but slower than IAMB. While on the Child3 dataset, the IAMB runs faster than the OFSVMB and the other four algorithms as shown in Table 7, and the running time is shown in Figure 4. On the Child10 dataset with a sample size of 5000, HITON-MB outperforms on OFSVMB, IAMB, STMB, BAMB, and EEMB in terms of F1, precision, and recall, while OFSVMB is run faster than the other 5 algorithms as shown in Table 7 in bold. Figure 3a,b shows the accuracy of OFSVMB and other algorithms, while the running time is given in Figure 4. On the Alarm10, Insurance datasets with a sample size of 5000, OFSVMB is more accurate than IAMB, STMB, HITON-MB, BAMB, and EEMB in terms of F1, precision, and recall as shown in Table 7 in bold. Meanwhile, OFSVMB shows higher accuracy and better running time than others, as shown in Figure 4. On a large-size dataset with a sample size of 5000, such as the Pig, the BAMB and EEMB are more accurate in terms of F1 and precision at a significance level of 0.01 and for recall at significance levels of 0.01 and 0.05, as shown in Table 7 in bold and in Figure 3a,b. The OFSVMB is more accurate in terms of F1 and precision at significance levels of 0.05 and runs faster than its rivals as shown in Figure 4. Moreover, on an extensive Gene dataset, the OFSVMB is still more accurate and runs faster than other state-of-the-art algorithms as shown in Table 7 in bold and in Figure 3a,b, and the running time is shown in Figure 4.

In a dense dataset with a sample size of 5000, such as the Barley, the EEMB is more accurate in terms of F1 and recall at a significant level of 0.01 and 0.05, as shown in Table 7 in bold and in Figure 3a,b. Still, OFSVMB is more accurate in terms of precision at a significance level of 0.01 and 0.05 than EEMB and runs faster than others, as shown in Figure 4. While on the Mildew dataset, the OFSVMB is more accurate than IAMB, STMB, BAMB, and EEMB. The HITON-MB is more accurate than OFSVMB in terms of precision at a significance level of 0.05, while comparable accuracy in terms of precision at a significance level of 0.01, as shown in Table 7 in bold and in Figure 3a,b and runs faster than other algorithms as shown in Figure 4.

5.3. Evaluation Classifiers

The number of selected features and the prediction accuracy of OFSVMB on 14 real-world datasets with low to high dimensionality is conducted using 12 classifiers including C1 = Coarse Gaussian SVM, C2 = Coarse KNN, C3 = Coarse Tree, C4 = Cosine KNN, C5 = Fine Gaussian SVM, C6 = Fine Tree, C7 = Linear Discriminant, C8 = Linear SVM, C9 = Medium KNN, C10 = Medium Tree, C11 = Subspace Discriminant, and C12 = Subspace KNN, where C standards for classifier and saving space in Tables 8–11, the abbreviation is used with numbers. For all the datasets, cross-validation is used 10-fold to prevent bias in error estimation. Moreover, the selected number of features and running time in seconds are also reported to show the efficiency of the algorithms.
Table 8. Comparison of prediction accuracy under C1–C6 classifiers at significance level = 0.01.

| Dataset     | Algorithm  | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 |
|-------------|------------|-----|-----|-----|-----|-----|-----|
| arcene      | OSFS       | 80  | 56  | 76  | 82  | 71  | 74  |
|             | Alpha-investing | 57  | 55  | 70  | 70  | 62  | 65  |
|             | SAOLA      | –   | –   | –   | –   | –   | –   |
|             | OFSVMB     | 80  | 65  | 76  | 82  | 71  | 74  |
| spect       | OSFS       | 70  | 70  | 70  | 46.4| 96  | 70  |
|             | Alpha-investing | 70  | 65.5| 69.3| 69.3| 96  | 69.7|
|             | SAOLA      | 70  | 70  | 70  | 46.4| 92  | 70  |
|             | OFSVMB     | 70  | 82.6| 72.7| 91.3| 97  | 71.9|
| wdbc        | OSFS       | 96.3| 93.3| 92.1| 96  | 87  | 93.1|
|             | Alpha-investing | 90  | 94  | 93  | 96.3| 88  | 93.1|
|             | SAOLA      | 91  | 91  | 91.6| 90.5| 88  | 90.5|
|             | OFSVMB     | 97.3| 95.8| 95.5| 96.1| 94  | 93.3|
| ionosphere  | OSFS       | 88.6| 87.2| 90  | 90.6| 88  | 89.2|
|             | Alpha-investing | 88  | 65.5| 77.4| 74.2| 79  | 71  |
|             | SAOLA      | 88.9| 73.2| 90.3| 88.3| 88  | 89.7|
|             | OFSVMB     | 94  | 68.7| 92  | 96.9| 89  | 91.5|
| colon       | OSFS       | 80.6| 64.5| 83.9| 80.6| 88  | 82.3|
|             | Alpha-investing | 66.1| 64.5| 77.4| 74.2| 79  | 71  |
|             | SAOLA      | 87.1| 64.5| 80.6| 90.3| 85  | 82.3|
|             | OFSVMB     | 88.7| 80  | 88.7| 98.3| 88  | 88.7|
| lung        | OSFS       | 83.7| 68.5| 77.3| 87.2| 82  | 81.8|
|             | Alpha-investing | 78.3| 68.5| 81.8| 93.6| 79  | 82.8|
|             | SAOLA      | 84.2| 68.5| 80.3| 89.2| –   | 84.2|
|             | OFSVMB     | 83.7| 75.5| 75.4| 97.7| 85  | 82.8|
| prostate-GE | OSFS       | 94.1| 51  | 92.2| 94.1| 99  | 92.2|
|             | Alpha-investing | 94.1| 51  | 90.2| 93.1| 88  | 90.2|
|             | SAOLA      | 96.1| 51  | 86.3| 96.1| 95  | 86.3|
|             | OFSVMB     | 94.1| 80  | 93.1| 98.1| 99  | 91.2|
| reged1      | OSFS       | 98.2| 88.4| 97.4| 94.8| 90  | 96.8|
|             | Alpha-investing | 88.2| 88.2| 88  | 88.2| 94  | 86.4|
|             | SAOLA      | 99.4| 88.2| 96.8| 98  | 93  | 96.6|
|             | OFSVMB     | 99.2| 82.2| 94.6| 98.4| 91  | 95.8|
| medalon     | OSFS       | 62  | 61.9| 61.7| 56  | 94  | 93  |
|             | Alpha-investing | 49.8| 50.5| 49.4| 49.2| 96  | 50  |
|             | SAOLA      | 62.4| 61.7| 60.9| 57.7| 96  | 58.3|
|             | OFSVMB     | 62.4| 91  | 61.5| 98  | 96  | 95.5|
| lycomphia   | OSFS       | 98.4| 67.7| 95.2| 98.4| 83  | 95.2|
|             | Alpha-investing | 93.5| 67.7| 96.8| 93.5| 88  | 96.8|
|             | SAOLA      | 98.4| 67.7| 87.1| 85.5| 80  | 87.1|
|             | OFSVMB     | 100 | 92.9| 96.5| 96.6| 70  | 97  |
| martil      | OSFS       | 88.2| 88.2| 87.8| 88.2| 87  | 79.2|
|             | Alpha-investing | 88.2| 88.2| 86.4| 88.8| 66  | 82.4|
|             | SAOLA      | 88.2| 88.2| 87.8| 88.2| 81  | 79.2|
|             | OFSVMB     | 95.2| 95.2| 89.6| 96.8| 89  | 79.4|
| medical     | OSFS       | 99.4| 95.6| 99.4| 99.4| 62  | 99.4|
|             | Alpha-investing | 99.1| 95.6| 99.2| 98.7| 49  | 99.4|
|             | SAOLA      | 98.7| 95.6| 99.4| 98.2| 61  | 99.4|
|             | OFSVMB     | 99.4| 97.6| 99.4| 99  | 62  | 99.7|
| leukemia    | OSFS       | 83.7| 68.5| 78.3| 88.2| 92  | 82.8|
|             | Alpha-investing | 78.3| 68.5| 77.8| 94.1| 50  | 81.8|
|             | SAOLA      | 84.2| 68.5| 79.3| 89.7| 81  | 85.2|
|             | OFSVMB     | 90.8| 80  | 81.8| 88.2| 94  | 85.9|
| sidol       | OSFS       | 83.7| 68.5| 74.3| 88.2| 88  | 86.9|
|             | Alpha-investing | 78.3| 68.5| 66.8| 94.1| 78  | 81.8|
|             | SAOLA      | 80.2| 68.5| 72.3| 89.7| 85  | 85.2|
|             | OFSVMB     | 81.8| 79.5| 79.8| 98.2| 90  | 89.3|
Table 9. Comparison of prediction accuracy under C7–C12 classifiers at significance level = 0.01.

| Dataset   | Algorithm | C-7 | C-8 | C-9 | C-10 | C-11 | C-12 |
|-----------|-----------|-----|-----|-----|------|------|------|
| arcene    | OSFS      | 84  | 85  | 73  | 74   | 85   | 74   |
|           | Alpha-investing | 76  | 74  | 67  | 65   | 72   | 73   |
|           | SAOLA     | -   | -   | -   | -    | -    | -    |
|           | OFSVMB    | 84  | 85  | 75  | 74   | 85   | 74   |
| spect     | OSFS      | 70  | 70  | 95  | 70   | 69.7 | 89.6 |
|           | Alpha-investing | 70.8| 67.8| 96  | 69.7 | 70   | 48.3 |
|           | SAOLA     | 70  | 70  | 91  | 70   | 70   | 70   |
|           | OFSVMB    | 71.2| 80  | 96  | 71.9 | 91   | 89.6 |
| wdbc      | OSFS      | 95.6| 96.5| 95  | 93.1 | 95.1 | 93   |
|           | Alpha-investing | 95.1| 97.4| 66  | 93.1 | 95.3 | 93.1 |
|           | SAOLA     | 92.1| 91.4| 74  | 91.2 | 92.6 | 75.2 |
|           | OFSVMB    | 95.6| 96.8| 87  | 93.3 | 95.4 | 94.9 |
| ionosphere| OSFS      | 87.2| 87.7| 77  | 89.2 | 86.3 | 88.9 |
|           | Alpha-investing | 86.6| 87.7| 85  | 71   | 85.8 | 93.2 |
|           | SAOLA     | 87.2| 87.7| 88  | 90   | 86.6 | 90.3 |
|           | OFSVMB    | 85.5| 87.9| 89  | 91.5 | 89.7 | 93.3 |
| colon     | OSFS      | 83.9| 83.9| 83  | 82.3 | 80.6 | 77.4 |
|           | Alpha-investing | 83.9| 82.3| 65  | 82.8 | 80.6 | 77.5 |
|           | SAOLA     | 87.1| 85.5| 65  | 82.3 | 87.1 | 79   |
|           | OFSVMB    | 88.7| 88.7| 88  | 88.7 | 88.7 | 78.2 |
| lung      | OSFS      | 94.6| 91   | 84.2| 93.1 | 89.7 |
|           | Alpha-investing | 97   | 90.2| 97  | 93.1 |
|           | SAOLA     | 94.1| 94.1| 82  | 82.8 | 91.6 | 90.6 |
|           | OFSVMB    | 95.1| 95.1| 99  | 92.2 | 94.1 | 81.4 |
| prostate-GE | OSFS     | 95.1| 95.1| 99  | 92.2 | 94.1 | 81.4 |
|           | Alpha-investing | 95.1| 96.1| 88  | 86.8 | 95.1 | 91.2 |
|           | SAOLA     | 95.1| 95.1| 98  | 86.3 | 60.8 | 95.1 |
|           | OFSVMB    | 94.1| 95.1| 99  | 91.2 | 94.1 | 94.2 |
| reged1    | OSFS      | 98.6| 98.9| 67  | 96.8 | 98.8 | 98.8 |
|           | Alpha-investing | 88.2| 88.2| 68  | 50.7 | 88.2 | 93.2 |
|           | SAOLA     | 99   | 99.2| 68  | 96.6 | 99   | 99.2 |
|           | OFSVMB    | 97.8| 98.8| 98  | 98.5 | 99.2 | 98.2 |
| medalon   | OSFS      | 61.3| 61.6| 51  | 61.5 | 61.2 | 55.4 |
|           | Alpha-investing | 50  | 50.4| 51  | 96.8 | 50   | 49.3 |
|           | SAOLA     | 61.8| 62.1| 51  | 60.6 | 61.8 | 57.2 |
|           | OFSVMB    | 61.5| 75.2| 95  | 75   | 61.4 | 57.8 |
| lycomphia | OSFS      | 98.4| 96.8| 83  | 95.2 | 96.8 | 98.4 |
|           | Alpha-investing | 98.4| 96.8| 72  | 83.8 | 96.8 | 93.5 |
|           | SAOLA     | 98.4| 98.4| 81  | 87.1 | 100  | 98.4 |
|           | OFSVMB    | 98.4| 100 | 79  | 97.5 | 100  | 98.7 |
| martil    | OSFS      | 88.2| 88.2| 87  | 84.6 | 88.2 | 77.2 |
|           | Alpha-investing | 94   | 94.1| 50  | 99.4 | 90.4 | 85   |
|           | SAOLA     | 88.2| 88.2| 75  | 84.6 | 88.2 | 77.2 |
|           | OFSVMB    | 87.6| 88.2| 88  | 92   | 94.5 | 85.2 |
| madical   | OSFS      | 99.5| 99.4| 62  | 99.4 | 99.4 | 96   |
|           | Alpha-investing | 99.4| 99.4| 62  | 99.4 | 98.1 | 95.8 |
|           | SAOLA     | 91.3| 99.4| 62  | 99.4 | 98.1 | 95.6 |
|           | OFSVMB    | 89.2| 91.6| 92  | 82.8 | 91.1 | 89.7 |
| leukemia  | OSFS      | 81.2| 91.8| 63  | 81.8 | 96.6 | 92.6 |
|           | Alpha-investing | 85.5| 93.1| 80  | 85.2 | 85.2 | 91.6 |
|           | SAOLA     | 89.2| 94.6| 91  | 84   | 91.1 | 88.7 |
|           | OFSVMB    | 89.2| 95.5| 89  | 87   | 90.6 | 92.6 |
| sid0      | OSFS      | 89.2| 91.6| 90  | 82.8 | 91.1 | 89.7 |
|           | Alpha-investing | 85.5| 91.8| 82  | 82   | 87.6 |
|           | SAOLA     | 87.1| 93.1| 85.2| 85.2 | 90.6 | 90.6 |
|           | OFSVMB    | 89.2| 95.5| 89  | 87   | 90.6 | 92.6 |
### Table 10. Comparison of mean prediction accuracy under C1–C6 classifiers at significance level = 0.01.

| Dataset | Algorithm    | C-1 | C-2 | C-3 | C-4 | C-5 | C-6 |
|---------|--------------|-----|-----|-----|-----|-----|-----|
| Mean    | OSFS         | 86.2| 73.5| 82.5| 85.7| 86.4| 86.8|
|         | Alpha-investing | 80.0| 70.8| 76.2| 85.0| 78.5| 81.5|
|         | SAOLA        | −   | −   | −   | −   | −   | −   |
|         | OFSVMB       | 88.4| 83.2| 85.4| 95.1| 86.7| 88.2|

### Table 11. Comparison of mean prediction accuracy under C7–C12 classifiers at significance level = 0.01.

| Dataset | Algorithm    | C-7 | C-8 | C-9 | C-10 | C-11 | C-12 |
|---------|--------------|-----|-----|-----|------|------|------|
| Mean    | OSFS         | −   | −   | −   | −    | −    | −    |
|         | Alpha-investing | 85.8| 69.6| 81.1| 85.6| 82.2| −    |
|         | SAOLA        | −   | −   | −   | −    | −    | −    |
|         | OFSVMB       | 87.5| 91.3| 87  | 87.6| 90.7| 87.9|

### Results and Discussion on Real-World Dataset

This segment shows the comparison of selected features, prediction accuracy, and efficiency of the OFSVMB algorithm with other state-of-the-art streaming-based algorithms, such as Alpha-investing, SAOLA, and OSFS, on 14 real-world datasets with a significance level of 0.01. The following Tables 8–12 describe the prediction accuracy, mean prediction accuracy based on 12 classifiers, and the number of selected features and running time in seconds.

### Table 12. #F (number of selected features) and running time (s) at significance level = 0.01.

| Algorithm    | OSFS #F | Time(s) | Alpha-Investing #F | Time(s) | SAOLA #F | Time(s) | OFSVMB #F | Time(s) |
|--------------|---------|---------|---------------------|---------|----------|---------|-----------|---------|
| arcene       | 5       | 3.02    | 7                   | 0.25    | −        | −       | 6         | 9.05    |
| spect        | 3       | 0.40    | 18                  | 2.02    | 5        | 0.30    | 3         | 0.16    |
| wdbc         | 3       | 0.27    | 20                  | 8.65    | 2        | 0.50    | 9         | 7.01    |
| ionosphere   | 5       | 0.2     | 10                  | 0.6     | 5        | 0.9     | 12        | 19.02   |
| colon        | 2       | 2.93    | 4                   | 0.17    | 4        | 1.78    | 13        | 84.2    |
| lung         | 8       | 51.82   | 45                  | 0.71    | 30       | 1.24    | 10        | 52.88   |
| Prostate-GE  | 2       | 2.10    | 12                  | 0.59    | 12       | 0.60    | 5         | 3.75    |
| reged1       | 11      | 34.28   | 1                   | 0.06    | 15       | 0.43    | 10        | 38.05   |
| madelon      | 6       | 1.05    | 1                   | 0.12    | 7        | 0.08    | 9         | 0.25    |
| lymphoma     | 3       | 5.13    | 5                   | 0.30    | 35       | 1.99    | 33        | 70.55   |
| mart1        | 3       | 4.99    | 28                  | 5.85    | 1        | 0.50    | 2         | 2.23    |
| medical      | 17      | 190.5   | 28                  | 1.15    | 16       | 5       | 6         | 212.8   |
| leukemia     | 5       | 1.5     | 2                   | 0.01    | 17       | 1.02    | 13        | 10.55   |
| sido         | 2       | 12.50   | 39                  | 5.83    | 40       | 3       | 9         | 6.80    |

Figure 5 shows the prediction accuracy of the algorithms based on 12 classifiers where the x-axis denotes the dataset, and the y-axis represents the prediction accuracy. In contrast, Figure 6 shows the running time in seconds of the algorithms. The sign “−” denotes that the algorithm fails to select any number of features in the corresponding dataset and takes longer. The better outcomes are underlined in bold text, presented in the following Tables 8–12.
Tables 8 and 9 describe the prediction accuracy of OFSVMB against OSFS, Alpha-investing, and SAOLA using 12 classifiers. As described in the tables, OFSVMB performs better than the other algorithms on most datasets. Its mean prediction accuracy is higher than others using the 12 classifiers as shown in Tables 10 and 11. The OFSVMB searches for PC and Spouse of the target feature T. In contrast, the OSFS, Alpha-investing, and SAOLA only search for PC and ignore the Spouse set of the target feature T, which causes them to lose the interpretability.
On the arcene dataset, SAOLA fails to obtain any features, as shown in Tables 8 and 9. OSFS and Alpha-investing have comparable accuracy with OFSVMB on very few datasets. OFSVMB somehow includes false positives, not worse than its rivals on real-world feature selection datasets. From Figure 5, the OFSVMB has better prediction accuracy compared with other algorithms on many datasets under 12 classifiers. Table 12 describes the number of selected features and the running time of the algorithms. Alpha-investing selects more features on many datasets because it does not re-evaluate the redundant features, making it ineffective but time-efficient. The OSFS selects fewer features than Alpha-investing and SAOLA and OFSVMB; when the new feature arrives, the OSFS first checks the relevancy, and then the arrived features’ redundancy, which improves accuracy against Alpha-investing and SAOLA, and selects fewer features.

The running time of OSFS is slower than Alpha-investing and SAOLA on many datasets because it considers features repeatedly to add or remove them from the feature set. The OFSVMB selects many features from many datasets and causes its running time to be slower than other algorithms, as shown in Figure 6. The OFSVMB searches for the PC and Spouse of the target feature T, while OSFS, Alpha-investing, and SAOLA only search the PC set of the target feature T.

5.4. Sensitivity Analysis

The OFSVMB algorithm is governed by \( \alpha \) (Alpha (\( \alpha \)) and significance level are similar) and \(|RI|\). We conducted a sensitivity analysis to investigate the effect of parameter values on the model’s accuracy. This section provides the details of sensitivity experiments. Concerning different parameters, including \( \alpha \) values and with different Rate of Instance \(|RI|\), the analysis is practiced on six real-world feature selection datasets, where the Rate of Instance \(|RI|\) is the subset of instances from the dataset. We choose three values of \( \alpha \) such as \((0.1, 0.01, 0.05)\) concerning three different values of the \(|RI|\) such as \((0.1, 0.8, 0.9)\).

The parameter \( \alpha \) determines how well the reconstruction function preserves the original, observable feature’s values. The \( \alpha=(0.1, 0.01, 0.05) \) is used to see the effect of OFSVMB prediction accuracy. In contrast, investigating \(|RI| = (0.1, 0.8, 0.9)\) can keep important features along the instance to affect the prediction accuracy of the OFSVMB. In Figure 7, the x-axis represents the Rate of Instance \(|RI|\), the y-axis represents parameter \( \alpha \), and the z-axis represents the prediction accuracy (%) on six real-world datasets including (a) spect, (b) sylva, (c) madelon, (d) marti1, (e) ionosphere, and (f) reged1. These six datasets contain small and large sample sizes and sparse data.

From the results given in Figure 7c, we observe that when \( \alpha \) is fixed with different \(|RI|\) values, the prediction accuracy of the OFSVMB algorithm increases for \(|RI| = 0.8 \) and 0.9. For some datasets given in Figure 7b,f, when the \(|RoI| = 0.1\), it can still keep the important feature along with instances and increase the prediction accuracy. We observe that the most optimal values of parameter \( \alpha \) are 0.01 and 0.05, and under these values with different \(|RI|\), the prediction accuracy of OFSVMB is higher. Still, the parameter \( \alpha = 0.1 \) with different \(|RI|\) also has an optimal contribution to the given six datasets regarding prediction accuracy. Thus, such an empirical value can be adopted in practice for future methods based on the statistical conditional independence test. These observations conclude that the OFSVMB is more accurate in selecting the optimal features regarding different \( \alpha \) with different \(|RI|\).
6. Conclusions

This paper proposes an Online Feature Selection via Markov Blanket (OFSVMB) algorithm that uses conditional independence $G^2$ and Fisher’s $z$-tests to find the MB based on streaming features. Once a feature is included in the PC and SP sets using online relevance analysis, it examines and checks for a true or false-positive feature using the online redundant analysis. OFSVMB tries to make the candidate features set of both PC and SP as small as possible, reducing the number of conditional independence tests. The proposed OFSVMB jointly identifies the Parents-Child and Spouse and separates them in streaming. The evaluation metrics such as F1, precision, and recall are used for evaluating the proposed algorithm on benchmark BN datasets and real-world datasets with 12 classifiers. Additionally, it also obtains the MB set with the highest accuracy. The results demonstrate that the OFSVMB is better than the traditional-based MB discovery algorithms such as IAMB, STMB, HITON-MB, BAMB, and EEMB on most benchmark BN datasets such as Child, Child3, Insurance, Pig, Gene with a sample size of 500 and Child3, Alarm10, Insurance, Gene, and Mildew with a sample size of 5000 with significance levels of 0.01 and 0.05. The OFSVMB also performs better on mean prediction accuracy regarding real-world datasets as compared to other streaming-based algorithms such as OSFS, Alpha-investing, and SAOLA using 12 classifiers including Fine Tree, Medium Tree, Coarse Tree, Linear Discriminant, Linear SVM, Fine Gaussian SVM, Coarse Gaussian SVM, Medium KNN, Coarse KNN, Cosine KNN, Subspace Discriminant, Subspace KNN. Furthermore, Searching feature strategies including the PC and Spouses of OFSVMB makes it a little time-consuming than OSFS, Alpha-investing, and SAOLA because these algorithms only search for the PC feature set and ignore the Spouses. OFSVMB is based on MB discovery, so it considers both PC and Spouses of the target feature $T$. In addition, the sensitivity analysis using two parameters $\alpha$ and Rate of Instance $|RI|$ also shows the OFSVMB performs better.

On a large and dense network with many features, statistical hypothesis-based tests for conditional independence causes performance inconsistency in OFSVMB and reduce accuracy. Using a V-structure cause the OFSVMB running time to be slower against OSFS,
Alpha-investing, and SAOLA because Alpha-investing and SAOLA select many features. However, they are still faster than our proposed algorithm.

In future work, we plan to overcome the limitations of the proposed algorithm and extend our work to address the direct causes and effects of obtaining the local causal discovery in the streaming feature of the target variable $T$ using mutual information or Neighborhood mutual information with a combination of conditional independence tests using different structure instead of V-structure. It will help to compute more true positive PCs and Spouses. In addition, the focus will be to improve the accuracy and consistently examine the impact of causal faith-fullness violations in the streaming feature selection.

**Author Contributions:** Conceptualization, W.K., L.K. and B.B; methodology, software, formal analysis, validation, data curation, writing—original draft preparation, W.K. and L.K.; investigation, resources, supervision, project administration, funding acquisition, L.K.; writing—review and editing, visualization, W.K., B.B., L.W. and H.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** There is no conflict of interest.

**Abbreviations**

The following abbreviations are used in this manuscript:

- MB Markov Blanket
- BN Bayesian Network
- PC Parents-Child
- SP Spouse
- SF Streaming Feature

**References**

1. Wu, D.; He, Y.; Luo, X.; Zhou, M. A Latent Factor Analysis-Based Approach to Online Sparse Streaming Feature Selection. *IEEE Trans. Syst. Man Cybern. Syst.* 2021. [CrossRef]
2. DeLatte, D.; Crites, S.T.; Guttenberg, N.; Yairi, T. Automated crater detection algorithms from a machine learning perspective in the convolutional neural network era. *Adv. Space Res.* 2019, 64, 1615–1628. [CrossRef]
3. Tsamardinos, I.; Aliferis, C. Towards Principled Feature Selection: Relevancy, Filters and Wrappers. In Proceedings of the International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 3–6 January 2003.
4. Aliferis, C.; Tsamardinos, I.; Statnikov, A. HITON: A Novel Markov Blanket Algorithm for Optimal Variable Selection. In *Annual Symposium Proceedings. AMIA Symposium*; AMIA: Bethesda, MD, USA, 2003; pp. 21–25.
5. Gao, T.; Ji, Q. Efficient Markov Blanket Discovery and Its Application. *IEEE Trans. Cybern.* 2017, 47, 1169–1179. [CrossRef]
6. Ling, Z.; Yu, K.; Wang, H.; Liu, L.; Ding, W.; Wu, X. BAMB: A Balanced Markov Blanket Discovery Approach to Feature Selection. *ACM Trans. Intell. Syst. Technol.* 2019, 10, 52:1–52:25. [CrossRef]
7. Wang, H.; Ling, Z.; Yu, K.; Wu, X. Towards efficient and effective discovery of Markov blankets for feature selection. *Inf. Sci.* 2020, 509, 227–242. [CrossRef]
8. Alnuaimi, N.; Masud, M.; Serhani, M.A.; Zaki, N. Streaming feature selection algorithms for big data: A survey. *Appl. Comput. Inform.* 2020. [CrossRef]
9. Pan, W.; Chen, L.; Ming, Z. Personalized recommendation with implicit feedback via learning pairwise preferences over item-sets. *Knowl. Inf. Syst.* 2019, 58, 295–318. [CrossRef]
10. Yang, S.; Wang, H.; Hu, X. Efficient Local Causal Discovery Based on Markov Blanket. *arXiv* 2019, arXiv:abs/1910.01288.
11. Sowmya, R.; Suneeetha, K. Data mining with big data. In *Proceedings of the 2017 11th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India, 5–6 January 2017; pp. 246–250.
12. Boulesnane, A.; Meshoul, S. Effective Streaming Evolutionary Feature Selection Using Dynamic Optimization. In *IFIP International Conference on Computational Intelligence and Its Applications*; Springer: Berlin, Germany, 2018; pp. 329–340.
13. Zhou, J.; Foster, D.; Stine, R.; Ungar, L. Streaming feature selection using alpha-investing. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in data Mining, Chicago, IL USA, 21–24 August 2005; pp. 384–393.

14. Yu, K.; Wu, X.; Ding, W.; Pei, J. Scalable and accurate online feature selection for big data. *ACM Trans. Knowl. Discov. Data (TKDD)* 2016, 11, 1–39. [CrossRef]

15. Wu, X.; Yu, K.; Ding, W.; Wang, H.; Zhu, X. Online feature selection with streaming features. *IEEE Trans. Pattern Anal. Mach. Intell.* 2012, 35, 1178–1192.

16. Yu, K.; Guo, X.; Liu, L.; Li, J.; Wang, H.; Ling, Z.; Wu, X. Causality-based feature selection: Methods and evaluations. *ACM Comput. Surv. (CSUR)* 2020, 53, 1–36. [CrossRef]

17. Wu, X.; Jiang, B.; Yu, K.; Chen, H.; Miao, C. Multi-label causal feature selection. *Proc. Aasi Conf. Artif. Intell.* 2020, 34, 6430–6437. [CrossRef]

18. Liu, C.; Yang, S.; Yu, K. Markov Boundary Learning With Streaming Data for Supervised Classification. *IEEE Access* 2020, 8, 102222–102234. [CrossRef]

19. Ling, Z.; Yu, K.; Wang, H.; Li, L.; Wu, X. Using feature selection for local causal structure learning. *IEEE Trans. Emerg. Top. Comput. Intell.* 2020, 5, 530–540. [CrossRef]

20. Zhou, P.; Wang, N.; Zhao, S. Online group streaming feature selection considering feature interaction. *Knowl.-Based Syst.* 2021, 226, 107157. [CrossRef]

21. You, D.; Wang, Y.; Xiao, J.; Lin, Y.; Pan, M.; Chen, Z.; Shen, L.; Wu, X. Online Multi-label Streaming Feature Selection with Label Correlation. *IEEE Trans. Knowl. Data Eng.* 2021. [CrossRef]

22. Li, L.; Lin, Y.; Zhao, H.; Chen, J.; Li, S. Causality-based online streaming feature selection. In *Concurrency and Computation: Practice and Experience; Wiley Online Library*; Hoboken, NJ, USA, 2021; p. e6347.

23. Wang, H.; You, D. Online Streaming Feature Selection via Multi-Conditional Independence and Mutual Information Entropy†. *Int. J. Comput. Syst. 2020, 13, 479–487. [CrossRef]*

24. You, D.; Wu, X.; Shen, L.; He, Y.; Yuan, X.; Chen, Z.; Deng, S.; Ma, C. Online Streaming Feature Selection via Conditional Independence. *Appl. Sci.* 2018, 8, 2548. [CrossRef]

25. Spirtes, P.; Glymour, C.; Scheines, R. Causation, prediction, and search. In *Causation, Prediction, and Search*; Springer: New York, NY, USA, 1993; pp. 238–258.

26. You, D.; Li, R.; Sun, M.; Ou, X.; Liang, S.; Yuan, F. Online Markov Blanket Discovery With Streaming Features. In Proceedings of the 2020 IEEE International Conference on Knowledge Graph (ICKG), Nanjing, China, 9–11 August 2020; pp. 92–99.

27. Singh, A.; Kumar, R. Heart Disease Prediction Using Machine Learning Algorithms. In Proceedings of the 2020 International Conference on Electrical and Electronics Engineering (ICE3), Gorakhpur, India, 14–15 February 2020; pp. 452–457.

28. Shen, Z.; Chen, X.; Garibaldi, J. Performance Optimization of a Fuzzy Entropy Based Feature Selection and Classification Framework. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 1361–1367.

29. Wu, D.; Luo, X.; Shang, M.; He, Y.; Wang, G.; Wu, X. A data-characteristic-aware latent factor model for web services QoS prediction. *IEEE Trans. Knowl. Data Eng.* 2020. [CrossRef]

30. Ucar, M.K.; Nour, M.; Sindi, H.F.; Polat, K. The Effect of Training and Testing Process on Machine Learning in Biomedical Datasets. *Math. Probl. Eng.* 2020, 2020, 2836236. [CrossRef]

31. Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catala, M.; Leskovec, J. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv* 2020, arXiv:abs/2005.00687.

32. He, Y.; Wu, B.; Wu, D.; Beyazit, E.; Chen, S.; Wu, X. Toward Mining Capricious Data Streams: A Generative Approach. *IEEE Trans. Neural Networks Learn. Syst.* 2021, 32, 1228–1240. [CrossRef][PubMed]

33. Ge, C.; Zhang, L.; Liao, B. Abstract 5327: KPG-121, a novel CRBN modulator, potently inhibits growth of metastatic castration resistant prostate cancer as a single agent or in combination with androgen receptor signaling inhibitors both in vitro and in vivo. *Cancer Res.* 2020, 80, 5327.

34. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liang, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11618–11628.