

Attribution 4.0 International (CC BY 4.0)

<https://creativecommons.org/licenses/by/4.0/>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.



Scalable and Flexible Two-Phase Ensemble Algorithms for Causality Discovery

Pei Guo^a, Yiyi Huang^b, Jianwu Wang^{a,*}

^a Department of Information Systems, University of Maryland, Baltimore County, Baltimore, MD, United States of America

^b Department of Hydrology & Atmospheric Sciences, University of Arizona, Tucson, AZ, United States of America

ARTICLE INFO

Article history:

Received 1 June 2020

Received in revised form 16 May 2021

Accepted 1 July 2021

Available online 16 August 2021

Keywords:

Causality discovery

Ensemble learning

Data parallelism

Granger causality

Dynamic Bayesian network

ABSTRACT

Causality study investigates cause-effect relationships among different variables of a system and has been widely used in many disciplines including climatology and neuroscience. To discover causal relationships, many data-driven causality discovery methods, e.g., Granger causality, PCML and Dynamic Bayesian Network, have been proposed. Many of these causality discovery approaches mine time-series data and generate a directed causality graph where each graph edge denotes a cause-effect relationship between the two connected graph nodes. Our benchmarking of different causality discovery approaches with real-world climate data show these approaches often generate quite different causality results with the same input dataset due to their internal learning mechanism differences. Meanwhile, there are ever-increasing available data in virtually every discipline, which makes it more and more difficult to use existing causality discovery algorithms to produce causality results within reasonable time. To address these two challenges, this paper utilizes data partitioning and ensemble techniques, and proposes a flexible two-phase causality ensemble framework. The framework first conducts phase 1 ensemble for partitioned data and then conducts phase 2 ensemble from phase 1 ensemble results. Based on the framework, we develop two ensemble approaches: i) data ensemble at phase 1 and algorithm ensemble at phase 2, and ii) algorithm ensemble at phase 1 and data ensemble at phase 2. To achieve scalability, we further parallelize the ensemble approaches via the Spark big data analytics engine. The proposed ensemble approaches are evaluated by synthetic and real-world datasets. Our experiments show that the proposed approaches achieve good accuracy through ensemble and high scalability through data-parallelization in distributed computing environments.

© 2021 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Causality [1] is a fundamental research topic studying cause-effect relationships among different components of a system and causality study can help explain why the system has certain behaviors. Causality learning/discovery has been widely studied and applied in many disciplines including climatology and neuroscience.

Many data-driven causality learning approaches have been proposed to mine time-series data [2], such as Granger causality [3], PCML [4], Dynamic Bayesian Network [5], and Convergent Cross Mapping [6]. These approaches take time-series data of two or more variables in a system as input and produce their predictions on cause-effect relationship among these variables. For instance, the work at [7] uses Granger causality to study cause-effect relationships among multiple climate variables and shows that sea

surface temperature changes at pacific ocean near equator, an indicator of the El Niño-Southern Oscillation (ENSO) climate phenomenon [8] can cause abnormal surface temperature, pressure and precipitation remotely.

One challenge with the variety of different causal discovery approaches is that these approaches often lead to divergent causality conclusions from the same dataset, which makes it difficult to explain and use data-driven causality discovery results. There have been some studies comparing different causality discovery methods [9,10]. For example, the experiments on comparing three causality discovery algorithms show there are only 83% overlapping among the results on average [10]. Yet there is still a lack of comprehensive framework to effectively integrate these diverse algorithms.

The other challenge to be tackled by this paper is the ever-increasing volume and dimension of available data for causality discovery. For instance, total worldwide climate data volume is projected to increase from 5 PB in 2010 to 350 PB in 2030 [11]. It is more and more difficult to use existing causality discovery al-

* Corresponding author.

E-mail address: jianwu@umbc.edu (J. Wang).

gorithms to handle the increasing dimensionality and resolution of these climate datasets. Meanwhile, data volume is just one factor for time complexity of many causality discovery algorithms. As an example, a popular Granger causality algorithm's execution time grows quadratically with the increase of either of the three factors: data record number, variable number and time lag number [12].

To address the above two challenges, this paper applies data partitioning and ensemble techniques to achieve scalable and accurate causality learning. Ensemble learning [13] is a meta machine learning algorithm which combines multiple base or individual learners in order to get better overall learning accuracy. In this paper, we propose a two-phase hybrid causality ensemble learning framework by first partitioning data into smaller sizes, and conducting a phase 1 ensemble for each data partition, then conducting phase 2 ensemble from phase 1 ensemble results. The framework can be easily parallelized through big data engines like Spark [14] or Flink [15], and is adaptable to different ensemble approaches. To the best of our knowledge, this study is the first supporting both scalable and ensemble learning for causality discovery. The software implementations of our work are open-sourced at [16].

The contributions of this paper are as follows.

- We propose a flexible two-phase causality ensemble framework by first conducting phase 1 ensemble for partitioned data and then conducting phase 2 ensemble from phase 1 ensemble results. The hybrid framework can combine learning results from different data partitions (namely data ensemble), and different algorithms (namely algorithm ensemble).
- Based on the above flexible framework, we propose two ensemble approaches for parallel causality ensemble learning via Spark [14] and the MapReduce programming model [17]. The first one conducts data ensemble in phase 1 and algorithm ensemble in phase 2. The second one conducts data ensemble and algorithm ensemble in the opposite order.
- We did experiments on synthetic and real-world datasets to evaluate our proposed scalable ensemble framework and approach, which show that our approaches can achieve both better accuracy and almost linear speedup.

This paper is an extension of our conference paper [18]. The major extensions include: 1) we expand the two-level hybrid causality framework to not only data-algorithm ensemble, but also algorithm-data ensemble; 2) we add experiments on both algorithm-ensemble and data-algorithm ensemble approaches and analyze their differences from the experiment results; 3) we apply our proposed methods to a real-world climate phenomenon: causality analysis of dynamics and thermodynamics variables near Arctic region.

The rest of the paper is organized as follows. The background is introduced in Section 2. The two-phase hybrid causality ensemble learning framework is explained in Section 3. Then, Section 4 describes the ensemble approaches and Section 5 explains how to parallelize these ensemble approaches. Experiment section 6 includes the data, experiments and evaluations. Related work discussion is in Section 7. Finally, Section 8 concludes our paper.

2. Background

2.1. Ensemble learning

Ensemble learning [13] is a meta machine learning algorithm which uses multiple learning methods to obtain better predictive performance than learning from any of the constituent methods. Since 1990, ensemble learning methods have become a major learning paradigm because of both empirical good performances

in real-world applications [19] and theoretical proof on its advantages [20]. Many state-of-art data mining approaches/packages, e.g., random forest [21] and XGBoost [22], are based on ensemble learning. Many ensemble learning algorithms have been proposed and they mainly vary in the following three aspects: 1) what are base/individual learners, 2) how each base learner learns from input data, 3) how to combine results of base learners. For base learner selection, if base learners used in an ensemble learning belong to the same type, e.g. decision tree or neural network, the ensemble algorithm is called homogeneous ensemble. Otherwise, it is called heterogeneous ensemble. On how each learner learns, there are three main approaches and they mostly differ in how input data is fed to base learner. The first approach, called stacking ensemble [23], uses the same input data for all base learners. Bagging ensemble [24], as the second approach, uses different sampling results from the original input data for different base learners. The third approach is boosting ensemble [25] which uses multiple base learners iteratively and, in each iteration, assigns higher weight to data whose learning accuracy was low in previous iterations. On base learner combination, common methods are majority voting and weighted majority voting [26].

Overall speaking, our proposed approach belongs to heterogeneous ensemble because the base/individual learners are different. Moreover, our work utilizes majority voting, while combining stacking ensemble and bagging ensemble in a flexible way. Traditional bagging ensemble takes sampling of the original input data. Since we are dealing with time-series data sets, random sampling removes chronologicity and breaks temporal causal relationships between the variables, so our approach divides dataset to several partitions on their index.

2.2. Causality discovery methods

Existing causal relationships discovery methods can be categorized into two types depending on the input data sets types: 1) learning from multivariate independent and identically distributed (i.i.d.) data and 2) learning from multivariate time-series data. The learning results from a multivariate causality approach can be denoted as a directed graph (see an example at Fig. 1) where each graph edge denotes a cause-effect relationship conditioned on all other variables in the graph.

In this subsection, we explain three multivariate causality discovery approaches towards time-series input data, namely multivariate (graphical) Granger causality [12], PCMCI [4] and dynamic Bayesian network [5] and their algorithm details. Because they all belong to the same causality discovery category and their learning results can be modeled as directed graphs, we could conduct ensemble learning using these algorithms as base learners which will be explained in later sections.

2.2.1. Multivariate Granger causality

Granger causality was proposed in 1969 as a predictive model in economics by Nobel Laureate Clive W. Granger. The Granger causality is defined as follows. One time series x Granger causes another time series y , if and only if the regression for y based on past values of both x and y is statistically significant than the regression of y only based on past values of y . Let the lagged variable x be x_{t-i} for i from 1 to maximum lag P , and similarly, the lagged y be y_{t-i} . To evaluate Granger causality, it first does the following two linear regressions:

$$y_t = a_{11} \cdot y_{t-1} + a_{12} \cdot y_{t-2} + \dots + a_{1P} \cdot y_{t-P} + \varepsilon_1 \quad (1)$$

$$y_t = a_{21} \cdot y_{t-1} + \dots + a_{2P} \cdot y_{t-P} + b_{21} \cdot x_{t-1} + \dots + b_{2P} \cdot x_{t-P} + \varepsilon_2 \quad (2)$$

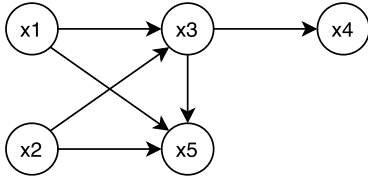


Fig. 1. Causality graph example.

Then it compares whether the regression function in Equation (1) performs better in accuracy than Equation (2) when predicting y_t . To decide which regression has better accuracy, Granger causality often uses a statistical hypothesis test method such as F -test or Chi-squared (χ^2) test to get a p -value to determine statistical significance.

To address the limitations of generating spurious causal relationships by the above pair-wise Granger causality method, multivariate Granger causality discovery, a.k.a. graphical Granger causality discovery fits a vector autoregressive model (VAR) to time-series data [27]. To demonstrate conditional Granger causality in VAR model, let $X_{l=1}^p$ denote $\{x_1, x_2, \dots, x_p\}$, from time $t = 1$ to $t = p$ and similarly, $Y_{l=1}^p$ as $\{y_1, y_2, \dots, y_p\}$, with coefficient matrix A_1, B_1, C_1, D_1 , the first joint VAR model is as follows.

$$\begin{cases} y_t = A_1 \cdot Y_{l=1}^p + B_1 \cdot X_{l=1}^p + \varepsilon_{1t} \\ x_t = C_1 \cdot X_{l=1}^p + D_1 \cdot Y_{l=1}^p + \varepsilon_{2t} \end{cases} \quad (3)$$

with the prediction error covariance matrix being:

$$\text{CovMatrix} = \begin{bmatrix} \text{var}(\varepsilon_{1t}) & \text{cov}(\varepsilon_{1t}, \varepsilon_{2t}) \\ \text{cov}(\varepsilon_{2t}, \varepsilon_{1t}) & \text{var}(\varepsilon_{2t}) \end{bmatrix} \quad (4)$$

Besides lagged variables $X_{l=1}^p$ and $Y_{l=1}^p$, when a new variable z is taken into account, with $Z_{l=1}^p$ representing $\{z_1, z_2, \dots, z_p\}$, the new VAR model is:

$$\begin{cases} y_t = A_2 \cdot Y_{l=1}^p + B_2 \cdot Z_{l=1}^p + C_2 \cdot X_{l=1}^p + \varepsilon_{3t} \\ z_t = D_2 \cdot Y_{l=1}^p + E_2 \cdot Z_{l=1}^p + F_2 \cdot X_{l=1}^p + \varepsilon_{4t} \\ x_t = G_2 \cdot Y_{l=1}^p + H_2 \cdot Z_{l=1}^p + I_2 \cdot X_{l=1}^p + \varepsilon_{5t} \end{cases} \quad (5)$$

Correspondingly, the prediction error covariance matrix of VAR model in (5) is:

$$\Sigma = \begin{bmatrix} \text{var}(\varepsilon_{3t}) & \text{cov}(\varepsilon_{3t}, \varepsilon_{4t}) & \text{cov}(\varepsilon_{3t}, \varepsilon_{5t}) \\ \text{cov}(\varepsilon_{4t}, \varepsilon_{3t}) & \text{var}(\varepsilon_{4t}) & \text{cov}(\varepsilon_{4t}, \varepsilon_{5t}) \\ \text{cov}(\varepsilon_{5t}, \varepsilon_{3t}) & \text{cov}(\varepsilon_{5t}, \varepsilon_{4t}) & \text{var}(\varepsilon_{5t}) \end{bmatrix} \quad (6)$$

Similar to the pairwise Granger causality testing, we care about whether introducing z can improve the prediction of y and how significant the improvement is. From the VAR model in Equation (3) of variable y and x , and the VAR model in Equation (5) of variable y , z , and x , the conditional Granger causality test from z to y conditioned on x , denoted as $(z \rightarrow y|x)$, is:

$$F\text{-test}(\text{var}(\varepsilon_{1t}), \text{var}(\varepsilon_{3t})) \quad (7)$$

From F -test in Equation (7), we can get a p -value and compare the p -value to a threshold to conclude whether z Granger causes y conditioned on x .

2.2.2. PCMCI

PCMCI is a causal discovery method described in [4] which identifies relevant variables for conditioning and estimates causality graph from time-series data. The method makes use of a "time series graph" made of nodes representing the state variables at different time-lags. If the time lag is denoted by l , a causal link is notated $x_{t-l} \rightarrow y_t$, and this link exists if x_{t-l} is not conditionally independent of y_t given the past of all variables. Assuming

the causal structure does not change over time, the same links are present at each time step.

The parents $\mathcal{P}(x)$ of a variable x are defined as the set of all nodes with a link towards x . However, estimating these parents directly by testing for conditional independence on the whole past is problematic due to high-dimensionality and because conditioning on irrelevant variables leads to biases.

PCMCI estimates causal links by a two-step procedure [4]:

1. Condition-selection: For every variable α , estimate a superset of parents $\tilde{\mathcal{P}}(\alpha_t)$ with an iterative Markov discovery algorithm [28] such as PC_1 algorithm. The condition-selection step reduces the dimensionality and avoids conditioning on irrelevant variables.

2. Momentary conditional independence (MCI): To test whether $x_{t-l} \rightarrow y_t$ with MCI, it evaluates:

$$x_{t-l} \perp y_t \mid \tilde{\mathcal{P}}(y_t), \tilde{\mathcal{P}}(x_{t-l}) \quad (8)$$

Equation (8) checks momentary conditional independence conditions between x_{t-l} and y_t , and makes null hypothesis of x_{t-l} and y_t are conditionally independent given $\tilde{\mathcal{P}}(y_t)$ and $\tilde{\mathcal{P}}(x_{t-l})$. To draw a causal link, if the null hypothesis is rejected, we say that x_{t-l} causes y_t .

2.2.3. Dynamic Bayesian network

Bayesian network [29] is one of many probabilistic graphical models which consists of a directed acyclic graph (DAG) and conditional probability distributions (CPDs) associated with each node in the model. Bayesian network can be used to make predictions, and decision making under uncertainty.

Dynamic Bayesian network [5] is similar to Bayesian network but with temporal extension, making it an appropriate graphical model to use for temporal datasets. The two main steps to creating a probabilistic graphical model are structure learning and parameter learning.

In this paper, we adopt the approach in [9] for dynamic Bayesian network learning. The approach first expands variable set by adding new variables for each original variable through time lagging. For instance, P new variables can be created from original variable x : x_{t-i} for i from 1 to maximum lag P . With the expanded variable set, the K2 algorithm [30] is used to search through all possible causality graph structures and identify which structure has the highest possibility to produce the data. In this score-based structure learning approach, Bayesian information criterion (BIC) scoring is used. Next, after causality graph is generated for expanded variable set, the causality graph is simplified by removing lagged variable and combining the causality edges. For instance, two edges $x_{t-2} \rightarrow y_{t-1}$ and $x_{t-3} \rightarrow y_t$ are combined to one edge $x \rightarrow y$ in the final graph.

Moreover, for the sake of computational time, the time-series data is partitioned into bins. Each bin defines a set of sub ranges, then the data is assigned to each labeled bin. For example, if the lowest value of the dataset is -5, and the highest value is 5. With the total bin number 10, a value of 1.2351 can be placed in a bin labeled 7, whose range is [1, 2). This approach increases the state counts of each variable and allows for faster computation.

3. A flexible two-phase causality ensemble learning framework

To deal with both increasing volume of available input data and increasing variety of different causality discovery algorithms, we propose a flexible two-phase causality ensemble framework that achieves ensemble of both multiple causality discovery algorithms as base learners and multiple data partitions as base learner input data. Before diving into the details of this hybrid framework, we first explain how ensemble could be done with only data ensemble and algorithm ensemble. We note most causality discovery algorithms generate not only cause-effect relationships, but also time

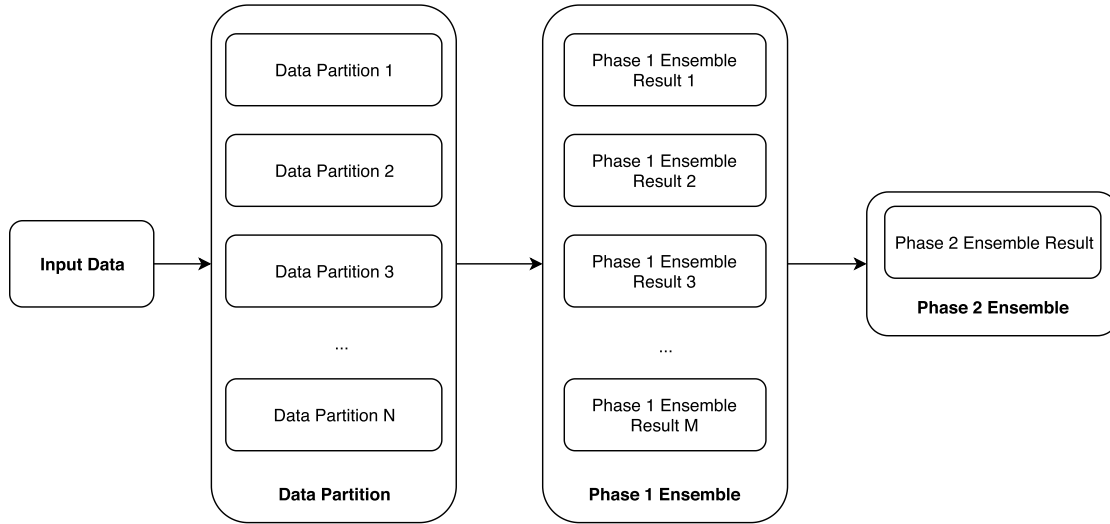


Fig. 2. A flexible two-phase ensemble framework for causality discovery.

lag and probability of each relationship. In this paper, we only focus on structure causality ensemble, namely how multiple directed graphs can be combined into one, and leave the ensemble of time lag and probability of causal edges for future work.

3.1. Algorithm ensemble for causality discovery

Algorithm ensemble approach deals with algorithm variety by applying different causality discovery algorithms as base learners with the same input data and later combining all base learner results. Each causality discovery algorithm mines the same time-series dataset and produces its own directed graph where nodes denote time series variables and each directed edge denotes a cause-effect relationship between the two connected variables. Because each base learner works on the same input data, the nodes of result graphs are the same for different base learners. But different base learners could produce different causality edges. Then by applying a certain base learner combination method, such as majority voting, we can derive a new directed graph as ensemble result. The nodes in the ensemble graph are the same with the results in each base learner. For graph edges, we can iterate all possible edges of the graph and decide whether this edge should be in the ensemble graph by combining corresponding edges in base learner graph result. If we use majority voting as combination method, an edge will be in ensemble graph only if the edge appears in more than half of base learner graphs.

By applying algorithm ensemble, the ensemble result is often more accurate than utilizing only one single causality discovery algorithm. However, when the size of input time-series dataset gets larger, the execution time of algorithm level ensemble increases dramatically because every base learner will take longer time to finish. Thus, a non-scalable algorithm ensemble approach is not enough to meet the challenge of dealing with the increasing data size.

3.2. Data ensemble for causality discovery

Data ensemble approach deals with data volume challenge by first partitioning data into smaller datasets, then using the same causality discovery algorithm as base learners with data partitions, and later combining all base learner results. Data partitioning is often done horizontally, not vertically, so that each data partition can still have all variables needed for multivariate causality learning. For time-series data, we need to preserve time dependency and

lagging within each data partition. So, instead of doing sampling in common bagging ensemble methods, data partitioning can be done by splitting the overall time ranges into smaller time ranges. Similar to algorithm ensemble, the nodes of resulting causality graph are the same for different base learners and edges of the graphs might be different. Then we can derive ensemble graph using the same base learner combination method in the previous subsection. The limitation of this approach is that it does not deal with variety of causality learning algorithms.

3.3. Two-phase hybrid ensemble for causality discovery

To address the challenges of diverse causality discovery results and increasing data size, we further integrate data ensemble and algorithm ensemble into one hybrid framework. As illustrated in Fig. 2, it conducts two-phase ensemble. In the hybrid ensemble framework, the input data is first partitioned into N data slices. Then phase 1 causality discovery is done based on the N data slices to get phase 1 ensemble result. In the end, phase 1 ensemble results are reduced to one final output through phase 2 ensemble.

This generic and flexible framework can be implemented in two ways. The first way, called **data-algorithm hybrid ensemble**, conducts data ensemble first in phase 1 and then algorithm ensemble in phase 2. The second way, called **algorithm-data hybrid ensemble**, conducts algorithm ensemble first in phase 1 and then data ensemble in phase 2. For data-algorithm hybrid ensemble, the count of discovered causality graphs, namely M in Fig. 2, is the same with the count of base learners because we get one causality graph from each causality discovery algorithm via data ensemble at phase 1. For algorithm-data hybrid ensemble, M equals N because it applies different algorithms and ensembles their results to get one causality graph for each data partition at phase 1.

4. Two-phase hybrid ensemble causality discovery approaches

Based on the flexible framework explained in previous section, two ensemble approaches are developed: 1) data-algorithm hybrid ensemble and 2) algorithm-data hybrid ensemble, as illustrated in Figs. 3 and 4, respectively. These two approaches are designed to effectively learn causal relationships from three data-driven causality learning approaches: Multivariate Granger causality (MGC), PCMCI and Dynamic Bayesian Network (DBN).

The data-algorithm hybrid ensemble approach (see Fig. 3) denotes that data ensemble happens in phase 1, then algorithm ensemble happens in phase 2. In this approach, the input data is

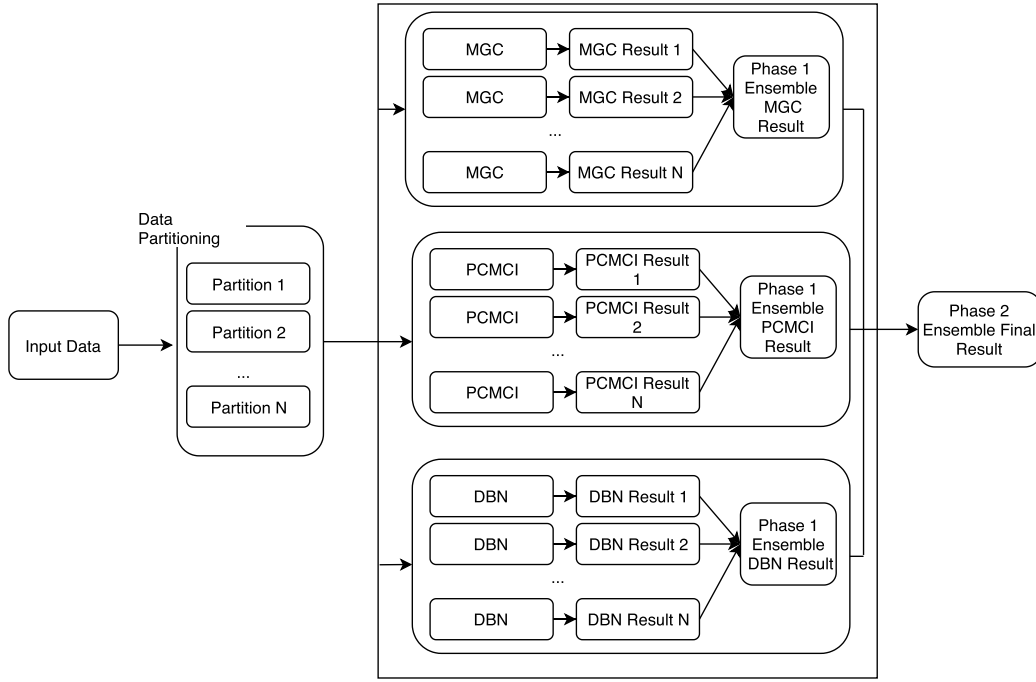


Fig. 3. Illustration of data-algorithm hybrid causality ensemble learning approach.

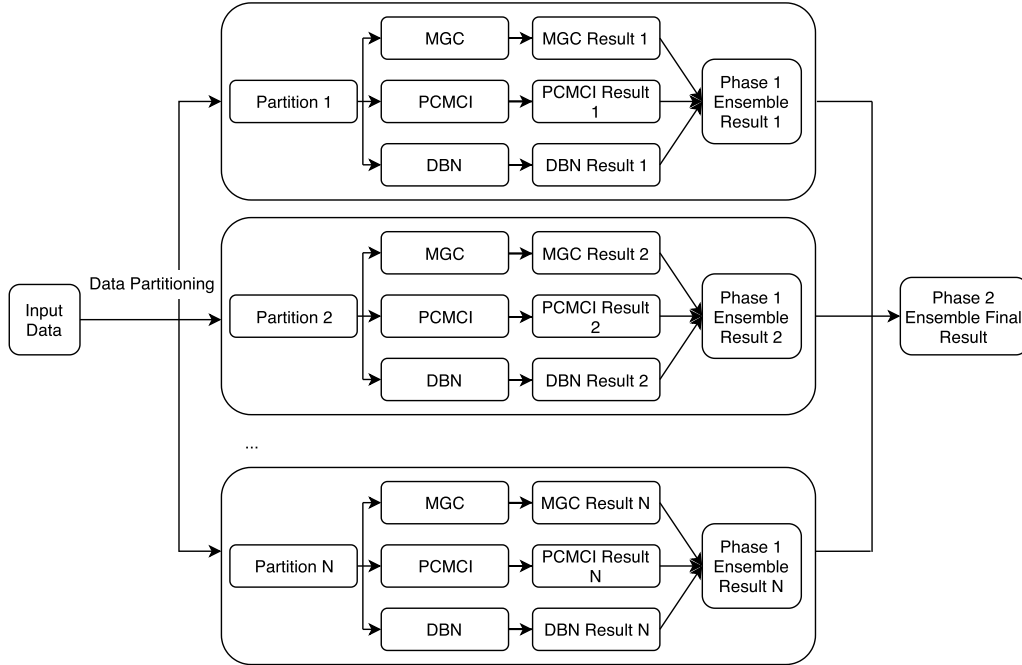


Fig. 4. Illustration of algorithm-data hybrid causality ensemble learning approach.

first partitioned into N slices. Then, each of the causality discovery method (*MGC*, *PCMCi* and *DBN*) is executed on all the partitioned data to get one causality output directed graph for each data slices. For example, *MGC* outputs *MGC_Result₁*, *MGC_Result₂*, ... *MGC_Result_N*. Different methods are executed in serial in the order of *MGC*, *PCMCi*, *DBN*. The outputs from all partitioned data slices corresponding to each causality method are collected for phase 1 ensemble. The phase 1 ensemble results are computed by majority voting. In its final step, the phase 1 ensemble results of each causality method (*MGC_Ensemble*, *PCMCi_Ensemble* and *DBN_Ensemble*) are reduced using ensemble

methods again to get phase 2 ensemble causality result as final output.

The algorithm-data hybrid ensemble approach (Fig. 4) has different workflow from the previous approach: algorithm ensemble happens in phase 1, and data ensemble happens in phase 2. In the beginning, the whole data set is still partitioned into N partitions. However, for each partition of the data, different causality methods are executed to output the result for this specific partition. For instance, the *data_partition_1* is executed by *MGC*, *PCMCi* and *DBN* to get *MGC_Result_1*, *PCMCi_Result_1* and *DBN_Result_1*. Then the results of all causality discovery methods for each data

Algorithm 1: Data-Algorithm Ensemble (*Data-Algorithm_Ensemble*).

Input: Different causality discovery methods: Multivariate Granger causality: *MGC*, PCMCI: *PCMCI*, Dynamic Bayesian Network: *DBN*, Time-series data: *D*, Number of data partitions: *N*

Output: A directed causality graph: $G = (V, E)$

- 1: Partition data *D* into *N* partitions as $\{d\} = d_1, d_2, \dots, d_N$
- 2: Get $E_{MGC} = \text{Data-Algorithm_Phase_1}(MGC, \{d\})$
- 3: Get $E_{PCMCI} = \text{Data-Algorithm_Phase_1}(PCMCI, \{d\})$
- 4: Get $E_{DBN} = \text{Data-Algorithm_Phase_1}(DBN, \{d\})$
- 5: ## Phase 2 edge ensemble:
- 6: **for** unique edges $\{e_i\}$ in E_{MGC} , E_{PCMCI} and E_{DBN} **do**
- 7: Count e_i appearance in E_{MGC} , E_{PCMCI} and E_{DBN} as n_i
- 8: **if** $n_i \geq 2$ **then**
- 9: Add e_i to final graph *G*
- 10: **end if**
- 11: **end for**
- 12: Output $G = (V, E)$

Algorithm 2: Phase 1 Ensemble for Data-Algorithm Ensemble (*Data-Algorithm_Phase_1*).

Input: Causality discovery method: *Causality*, Data partition set: $\{d\}$

Output: A set of directed edges in Graph corresponding to causality discovery method: $E_{causality}$

- 1: **for** each data partition d_i in $\{d\}$ **do**
- 2: Get Causality edge set from causality computation: $E_i = \text{Causality}(d_i)$
- 3: **end for**
- 4: ## Phase 1 edge ensemble:
- 5: **for** unique edges $\{e_j\}$ in all E_i **do**
- 6: Count e_j appearance in all E_i as n_j
- 7: **if** $n_j > N/2$ **then**
- 8: Add e_j to $E_{causality}$
- 9: **end if**
- 10: **end for**
- 11: Output $E_{causality}$

partition are combined through phase 1 ensemble. In phase 2 ensemble, the final output causality results are generated using all the phase 1 ensemble results.

4.1. Data-algorithm hybrid ensemble

The data-algorithm ensemble approach includes two algorithms. Algorithm 1 (*Data-Algorithm_Ensemble*) is for the full two-phase hybrid ensemble approach and illustrated in Fig. 3. Algorithm 2 (*Data-Algorithm_Phase_1*) is for phase 1 data ensemble corresponding to each phase 1 ensemble block in Fig. 3.

The input of the *Data-Algorithm_Ensemble* (Algorithm 1) includes different causality discovery methods, which are multivariate Granger causality (*MGC*), PCMCI (*PCMCI*) and Dynamic Bayesian Network (*DBN*), time series input data *D*, and the number of data partitions *N*. The logic of Algorithm 1 for the whole ensemble process is as follows. In line 1, the input dataset *D* is first partitioned into *N* slices by its timestamp as $\{d\} = d_1, d_2, \dots, d_N$ where the time interval of each slice is only $1/N$ of the original time series. Then it calls Algorithm 2 (*Data-Algorithm_Phase_1*) to execute each causality discovery method to get phase 1 ensemble causality edge set E_{MGC} , E_{PCMCI} and E_{DBN} from all the data partitions in lines 2-4. Finally, in lines 6-11, phase 2 ensemble result is computed by majority voting on edge set of all causality mining methods, E_{MGC} , E_{PCMCI} and E_{DBN} , that if two or more causality ensemble edge sets contain the same edge, this edge is added into final output graph $G = (V, E)$ with *V* denoting nodes and *E* as edges in line 12.

The phase 1 data ensemble in the data-algorithm ensemble approach, namely *Data-Algorithm_Phase_1* is shown in Algo-

Algorithm 3: Algorithm-Data Ensemble (*Algorithm-Data_Ensemble*).

Input: Different causality discovery methods: Multivariate Granger causality: *MGC*, PCMCI: *PCMCI*, Dynamic Bayesian Network: *DBN*, Time-series data: *D*, Number of data partitions: *N*

Output: A directed causality graph: $G = (V, E)$

- 1: Partition data *D* into *N* partitions as $\{d\} = d_1, d_2, \dots, d_N$
- 2: **for** each data partition d_i in $\{d\}$ **do**
- 3: Run phase 1 ensemble on data partition d_i to get edge set $E_{d_i} = \text{Algorithm-Data_Phase_1}(MGC, PCMCI, DBN, d_i)$
- 4: **end for**
- 5: ## Phase 2 edge ensemble:
- 6: **for** unique edges $\{e_j\}$ in all E_{d_i} **do**
- 7: Count e_j appearance in all E_{d_i} as n_j
- 8: **if** $n_j > N/2$ **then**
- 9: Add e_j to final graph *G*
- 10: **end if**
- 11: **end for**
- 12: Output $G = (V, E)$

gorithm 2. Its inputs include the specific causality discovery method *Causality*, and the partitioned time-series data set $\{d\}$. In lines 1-3, the causality discovery method executes for each data partition d_i in $\{d\}$ to output a causality edge set E_i from *Causality*(d_i). Since this causality edge set contains edges from each partition, in lines 5-10, phase 1 ensemble method loops to check if the number of a given edge e_j appears in more than half of the partition edge set. For instance, if there are 10 partitions, and a causality edge (x_1, x_2) appears 6 times in all the partition edge set, it is added to the phase 1 ensemble output $E_{causality}$ as in line 8 then be output as in line 11.

4.2. Algorithm-data hybrid ensemble

The algorithm-data hybrid ensemble approach also contains two algorithms: Algorithm 3 (*Algorithm-Data_Ensemble*) for the full two-phase hybrid ensemble approach as illustrated in Fig. 3 and Algorithm 4 (*Algorithm-Data_Phase_1*) for its phase 1 algorithm ensemble.

The *Algorithm-Data_Ensemble* (Algorithm 3) algorithm requires the same input as the *Data-Algorithm_Ensemble* algorithm: the causality discovery methods of *MGC*, *PCMCI*, *DBN*, the time series input data *D* and the number of data partitions *N*. In line 1 of Algorithm 3, the input time-series data *D* is first partitioned into *N* slices as $\{d\} = d_1, d_2, \dots, d_N$. In next step, from lines 2-4, for each data partition d_i , the algorithm for phase 1 ensemble (*Algorithm-Data_Phase_1*) is called with input of the causality discovery methods (*MGC*, *PCMCI*, *DBN*) and the current data partition d_i to get phase 1 ensemble output of edge set E_{d_i} . Then, in lines 6-11, phase 2 ensemble is executed on all edge set $E_{d_1}, E_{d_2}, \dots, E_{d_N}$. In these edge set, every unique causality edge in the $\{e_j\}$ is checked to see if it appears in more than half of the partition number *N* as majority voting in lines 8-10. Finally, all voted causality edges are added to final graph and output as a directed graph result $G = (V, E)$.

The phase 1 ensemble for algorithm-data hybrid ensemble approach is called *Algorithm-Data_Phase_1* and shown in Algorithm 4. The input of this phase 1 ensemble algorithm contains the three causality mining methods (*MGC*, *PCMCI* and *DBN*) and the specific data partition (d_i). From line 1 to line 5 of the algorithm, for each causality discovery method, this data partition is fed to the method to get its causality edge result set. Majority voting is applied on all causality edges result set E_{MGC} , E_{PCMCI} and E_{DBN} in lines 7-12. For instance, if a causality edge $e = (x_1, x_2)$ appears in E_{MGC} and E_{DBN} but not in E_{PCMCI} , it still passes majority voting, and is saved to the output ensemble of this data partition as E_{d_i} in line 10. Finally, the algorithm outputs phase 1 edge ensemble result E_{d_i} in line 13.

Algorithm 4: Phase 1 Ensemble for Algorithm-Data Ensemble (*Algorithm-Data_Phase_1*).

Input: Causality discovery methods: *MGC*, *PCMC1*, *DBN*
Time-series data partition: d_i
Output: A set of directed edges in Graph corresponding to this specific partition d_i : E_{d_i}

```

1: for MGC, PCMC1, DBN do
2:   Get  $E_{MGC} = MGC(d_i)$ 
3:   Get  $E_{PCMC1} = PCMC1(d_i)$ 
4:   Get  $E_{DBN} = DBN(d_i)$ 
5: end for
6: ## Phase 1 edge ensemble:
7: for unique edges  $\{e_j\}$  in  $E_{MGC}$ ,  $E_{PCMC1}$ ,  $E_{DBN}$  do
8:   Count  $e_j$  appearance in  $E_{MGC}$ ,  $E_{PCMC1}$  and  $E_{DBN}$  as  $n_j$ 
9:   if  $n_j \geq 2$  then
10:    Add  $e_j$  to  $E_{d_i}$ 
11:   end if
12: end for
13: Output  $E_{d_i}$ 

```

5. Parallel two-phase hybrid causality ensemble learning via spark big data engine

When facing large volume of datasets, like research done for big data machine learning [31,32], scalable algorithms are also crucial to reduce computation time for causality discovery. To achieve scalability for our approaches, the above two-phase hybrid causality ensemble approaches are further implemented in parallel via Spark [14] to deal with big data in two aspects: 1) automatic data partitioning and 2) parallel function mapping.

Regarding the data partitioning part in our parallel implementation, the data is first load into Spark as resilient distributed dataset (RDD), then it is automatically partitioned by timestamp of each record, as in the phase 2 ensemble for both data-algorithm hybrid ensemble and algorithm-data hybrid ensemble approaches, in Algorithm 1 line 1 and Algorithm 3 line 1, respectively. More specifically, every data partition, as a chunk of the large distributed dataset, is assigned an index for phase 1 ensemble in next step.

For parallel function mapping, the parallelization of data-algorithm hybrid ensemble is implemented in its phase 1 ensemble, as in Algorithm 2 lines 1-3. With Spark RDD partitioning, now each data partition d_i becomes an RDD partition. Then these RDD partitions are mapped to be transformed by the causality discovery method *Causality* in parallel then be reduced as edge set E_i for later phase 2 ensemble computation. For algorithm-data hybrid ensemble approach, the parallelization is in its phase 2, located in lines 2-4 of Algorithm 3. Algorithm 4 (*Algorithm-Data_Phase_1*) is mapped to execute in parallel with each of the partitioned RDD $\{d_i\}$ with index i from its input. Then, all phase 1 ensemble results are reduced for phase 2 ensemble to generate the final output graph.

6. Experiments**6.1. Experiments setup****6.1.1. Environment**

The experiments were conducted on top of the HPCF2018 cluster at the University of Maryland, Baltimore County [33], where each computing node has two 18-core Intel Skylake CPUs, 384 GB of memory and a 120 GB SSD disk. For our experiment environment, each cluster contained one master node and several worker nodes. Moreover, the Spark programs are managed by Slurm workload manager [34] in standalone cluster mode.

For software, we used Python (version 3.6.8), Spark (version 2.4). In our experiments, each node contains one executor, each driver/executor's memory is 200 GB, and partition number is set as 48.

6.1.2. Baseline approaches and parameter settings

We employed seven baseline approaches in our experiments. The first three were single causality discovery approaches: Multivariate Granger causality (*MGC*), *PCMC1* and Dynamic Bayesian Network (*DBN*). The next three were corresponding data ensemble approaches for each of the three single causality discovery approaches following the way described in Section 3.2. The last one was an algorithm ensemble approach by combining all the three single causality discovery approaches following the way described in Section 3.1. For experiment parameter settings, we set the maximum time lagging as 3 for synthetic data and 14 for real-world data. We also set the p -value threshold as 0.05 for both *MGC* and *PCMC1* tests. Besides, the total bin number for *DBN* was set to 5 to reduce computation time. In *PCMC1* method, we utilize its different conditional independence tests for linear and nonlinear causality discovery. For nonlinear conditional independence tests, as we have large dataset, RCOT test is applied.

6.2. Experiments with synthetic datasets**6.2.1. Dataset description**

The datasets used in our experiments are mainly in two categories: synthetic data and real-world data. For synthetic data, we generated four datasets based on predefined equations to evaluate our proposed algorithms' performance. One important reason for synthetic dataset generation is to know causality ground truth so we could evaluate learning result accuracy. Similar to the synthetic datasets generation approach for Granger causality and *DBN* evaluation in [9], we generated our synthetic datasets based on linear and nonlinear causal dependency Equation (9) and Equation (10), where ε s are random noises. All the x s are initialized as 0, and the ε s are drawn from normal distribution and not dependent on time t . The causality graphs for the equations can be found at Fig. 1 and Fig. 5. The linear and nonlinear datasets with different sizes (namely 1 million and 10 million for row numbers) were generated using the same equations correspondingly.

$$\begin{cases} x_1(t) = 0.95 \cdot \sqrt{2} \cdot x_1(t-1) - 0.90 \cdot x_1(t-2) + \varepsilon_1 \\ x_2(t) = 0.5 \cdot x_2(t-1) + \varepsilon_2 \\ x_3(t) = -0.5 \cdot x_1(t-1) + 0.25 \cdot \sqrt{2} \cdot x_3(t-1) \\ \quad + 0.25 \cdot \sqrt{2} \cdot x_2(t-1) + \varepsilon_3 \\ x_4(t) = -0.95 \cdot x_4(t-1) - 0.25 \cdot \sqrt{2} \cdot x_3(t-1) + \varepsilon_4 \\ x_5(t) = 0.5 \cdot x_1(t-1) + 0.95 \cdot x_2(t-2) - 0.25 \cdot \sqrt{2} \cdot x_3(t-1) \\ \quad + 0.5 \cdot x_5(t-1) + \varepsilon_5 \end{cases} \quad (9)$$

$$\begin{cases} x_1(t) = 0.125 \cdot \sqrt{2} \cdot \exp(-x_1(t-1)^2/2) + \varepsilon_1 \\ x_2(t) = 1.2 \cdot \exp(-x_1(t-1)^2/2) + \varepsilon_2 \\ x_3(t) = -1.05 \cdot \exp(-x_1(t-1)^2/2) \\ \quad + 0.2 \cdot \sqrt{2} \exp(-x_2(t-2)^2/2) + \varepsilon_3 \\ x_4(t) = -1.15 \cdot \exp(-x_1(t-2)^2/2) \\ \quad + 0.2 \cdot \sqrt{2} \cdot \exp(-x_4(t-1)^2/2) \\ \quad + 1.35 \cdot \exp(-x_3(t-1)^2/2) + \varepsilon_4 \\ x_5(t) = -1.15 \cdot \exp(-x_2(t-1)^2/2) + \varepsilon_5 \end{cases} \quad (10)$$

6.2.2. Consistency evaluation

Result consistency with different data partitions. We tested the consistency of our two-phase ensemble algorithms with different data partitions as 48, 100, 150 and 200. The experiments show both two-phase algorithm-data ensemble and two-phase data-algorithm ensemble generate the same results under these different data partition settings. It demonstrates the stableness of our ensemble approaches. We only chose 48 for the following experiments.

Table 1
Result matrix similarity of linear 10M synthetic dataset.

Matrix Similarities	MGC	PCMC	DBN	Data-level Ensemble MGC	Data-level Ensemble PCMC	Data-level Ensemble DBN	Algorithm-level Ensemble	Two-phase Algorithm-data Ensemble	Two-phase Data-algorithm Ensemble
MGC	1.000	0.750	0.600	1.000	0.857	0.500	0.857	0.857	0.857
PCMC	0.750	1.000	0.600	0.750	0.857	0.500	0.857	0.857	0.857
DBN	0.600	0.600	1.000	0.600	0.667	0.889	0.667	0.667	0.667
Data-level Ensemble MGC	1.000	0.750	0.600	1.000	0.857	0.500	0.857	0.857	0.857
Data-level Ensemble PCMC	0.857	0.857	0.667	0.857	1.000	0.556	1.000	1.000	1.000
Data-level Ensemble DBN	0.500	0.500	0.889	0.500	0.556	1.000	0.556	0.556	0.556
Algorithm-level Ensemble	0.857	0.857	0.667	0.857	1.000	0.556	1.000	1.000	1.000
Two-phase Algorithm-data Ensemble	0.857	0.857	0.667	0.857	1.000	0.556	1.000	1.000	1.000
Two-phase Data-algorithm Ensemble	0.857	0.857	0.667	0.857	1.000	0.556	1.000	1.000	1.000

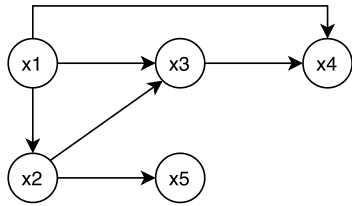


Fig. 5. Nonlinear synthetic data ground truth causal graph.

Result consistency with different individual causality discovery methods. Since the causality discovery methods produce different results, to quantify the similarity coefficients, we used the well-known matrix distance calculation to measure Jaccard coefficients for different combinations of the model results. All pair-wise matrix similarities among different methods for linear 10M and nonlinear 10M synthetic data are presented in Table 1 and Table 2, respectively. From the two tables, we have the following four observations. First, the results from the three base learners are quite divergent (0.650 for linear dataset average similarity and 0.478 for nonlinear dataset average similarity). Second, by doing data-level ensembles for the three base learners, the results are a little more divergent for linear dataset (0.637 for average similarity), but more similar for nonlinear dataset (0.738 for average similarity). Third, the results from data-level ensembles are still different from those algorithm-level ensembles (0.804 for linear dataset average similarity and 0.469 for nonlinear dataset average similarity). Fourth, the results from the two-phase ensembles are identical (1.000 for both linear and nonlinear dataset average similarity). It shows our two-phase ensemble approach is an effective way to achieve consistent results.

6.2.3. Accuracy evaluation

We employ Structural Hamming Distance (SHD) metric [35] to compare accuracy of different approaches. SHD is a common metric to measure the difference between two directed graphs with the same node set. SHD value is defined as the total step count of three types of actions needed to transform from one direct graph to another direct graph: 1) reversing an edge's direction, 2) removing an extra edge, 3) adding a missing edge. We calculate SHD between ground truth graph and each learned graph. The lower SHD value means the more similarity between the two graphs, so the algorithm that generates the learned graph is more accurate.

Single Causality Discovery Methods. Our first experiment com-

pared the correctness of single causality discovery method of *MGC*, *PCMC* and *DBN*. The comparison results for 1M and 10M linear and nonlinear data sets are in left three columns of Table 3. It shows that different causality discovery methods had quite different results with these four data sets.

Single-phase Causality Ensemble Methods. We also measured the accuracy of the three data ensemble baseline approaches and the algorithm ensemble causality ensemble approach. The results were shown in the middle four columns of Table 3. For linear datasets, we could see from the table that both data ensemble and algorithm approach could achieve the same or better accuracy than single causality discovery approaches. For nonlinear datasets, data ensemble approaches still performs better in accuracy; however, algorithm ensemble could perform a little bit worse due to two algorithms making the same wrong prediction on certain edges.

Two-phase Hybrid Causality Ensemble Methods. The results of the two hybrid causality ensemble approaches, namely algorithm-data ensemble and data-algorithm ensemble, are shown in the right two columns of Table 3. In the experiments, our hybrid causality ensemble approaches achieve perfect accuracy because their SHD values are all zero. In linear experiments, compared to data ensemble and algorithm ensemble baseline approaches, our two-phase hybrid causality ensemble approaches could get the same or better results. In nonlinear experiments, two-phase hybrid ensemble approaches achieve better accuracy than both data ensemble and algorithm ensemble. They both perform better than all the baseline approaches in accuracy for all the data sets.

6.2.4. Comparison with other causality discovery methods

We compare our two-phase hybrid ensemble methods with two state-of-art causality discovery methods: Directed Acyclic Graph Neural Networks (DAG-GNN) [36] and Temporal Causality Discovery Framework (TCDF) [37] using their public GitHub implementations. In the comparison, 1M linear and nonlinear synthetic datasets are used. Moreover, since these two methods are both neural network methods and their results differ with different hyperparameter settings, we generate outcomes using different hyperparameter settings. The results are shown in Tables 4 and 5 for DAG-GNN and TCDF respectively.

In Table 4, two sets of hyperparameters are tuned in our experiments: t and τ . t is a threshold for causal edges where an edge will not be in final causal graph if its corresponding adjacency matrix value is smaller than t . τ is a regularization parameter. The default values are $t = 0.3$ and $\tau = 0.0$. We note that DAG-GNN method is originally for causal discovery for iid data, not

Table 2

Result matrix similarity of nonlinear 10M synthetic dataset.

Matrix Similarities	MGC	PCMC	DBN	Data-level Ensemble MGC	Data-level Ensemble PCMC	Data-level Ensemble DBN	Algorithm-level Ensemble	Two-phase Algorithm-data Ensemble	Two-phase Data-algorithm Ensemble
MGC	1.000	0.550	0.583	0.667	0.583	0.417	0.883	0.500	0.500
PCMC	0.550	1.000	0.300	0.421	0.368	0.263	0.450	0.316	0.316
DBN	0.583	0.300	1.000	0.667	0.75	0.714	0.545	0.857	0.857
Data-level Ensemble MGC	0.667	0.421	0.667	1.000	0.875	0.625	0.500	0.750	0.750
Data-level Ensemble PCMC	0.583	0.368	0.750	0.875	1.000	0.714	0.545	0.857	0.857
Data-level Ensemble DBN	0.417	0.263	0.714	0.625	0.714	1.000	0.363	0.833	0.833
Algorithm-level Ensemble	0.883	0.450	0.545	0.500	0.545	0.363	1.000	0.455	0.455
Two-phase Algorithm-data Ensemble	0.500	0.316	0.857	0.750	0.857	0.833	0.455	1.000	1.000
Two-phase Data-algorithm Ensemble	0.500	0.316	0.857	0.750	0.857	0.833	0.455	1.000	1.000

Table 3

Structural Hamming Distance (SHD) comparison of single causality discovery approach, baseline ensemble approach, and our proposed two-phase hybrid causality approach (best results are in bold).

	MGC	PCMC	DBN	Data-level Ensemble MGC	Data-level Ensemble PCMC	Data-level Ensemble DBN	Algorithm-level Ensemble	Two-phase Algorithm-data Ensemble	Two-phase Data-Algorithm Ensemble
Linear 1M	1	1	4	1	1	4	0	0	0
Linear 10M	1	1	3	1	1	4	0	0	0
Nonlinear 1M	5	13	1	4	3	3	4	0	0
Nonlinear 10M	6	6	1	2	1	1	3	0	0

Table 4

Structural Hamming Distance (SHD) comparison of DAG-GNN method static and temporal graphs with different hyperparameters (best results are in bold).

		$t = 0.3, \tau = 0.0^*$	$t = 0.2, \tau = 0.0$	$t = 0.3, \tau = 1e-07$	$t = 0.2, \tau = 1e-07$
Static	Linear 1M	11	8	12	7
	Nonlinear 1M	6	9	6	7
Temporal	Linear 1M	10	9	13	9
	Nonlinear 1M	13	11	11	9

Table 5

Structural Hamming Distance (SHD) comparison of TCDF method with different hyperparameters (best results are in bold).

	Hidden layers = 0 Kernel size = 4 *	Hidden layers = 0 Kernel size = 6	Hidden layers = 1 Kernel size = 4	Hidden layers = 1 Kernel size = 6
Linear 1M	3	3	5	4
Nonlinear 1M	3	2	2	4

for time-series data. Directly applying iid data methods on time-series dataset might introduce complicated dependencies between contemporaneous data X_t and Y_t [38]. One way to utilize the iid causal method for time-series data is to treat each variable with time lag as a new variable. Thus, in Table 4, we show *static* results and *temporal* results. Static results indicate that the data points are treated as iid records and directly fed into the model. As a comparison, for the temporal results, the time-series data are first augmented by time lag, then the full augmented lagged dataset is used as the input. Moreover, the initial result graph is reduced to an output graph by shrinking all the lagged variables to its original variables. For instance, the edge $x_{t-3} \rightarrow y_{t-1}$ is reduced to $x \rightarrow y$. Finally the edges with too small probabilities are filtered out. We find that DAG-GNN algorithm is sensitive to the hyperparameters and the results are quite different in terms of

SHD. For the linear synthetic dataset, the best result is the static graph with $t = 0.2$ and $\tau = 1e - 07$. And for nonlinear synthetic datasets, the best graphs are static graphs with $t = 0.3, \tau = 0.0$ and $t = 0.3, \tau = 1e - 07$.

For the TCDF method, since it is originally developed for time-series dataset, we directly utilized the authors' original implementation. As shown in Table 5, we configured two hyperparameters, which are *hidden layers* and *kernel size*. The default values are *hidden layers* = 0 and *kernel size* = 4. Based on the TCDF paper, its *kernel size* should be $\text{maxlag} + 1$. We use *kernel size* = 4, which means $\text{maxlag} = 3$ and is the same to the maxlag setting in previous experiments. In addition, we also tried *kernel size* = 6 to check its performance difference. Our experiments show that with different hyperparameters combination choices, the output graphs are also quite different in terms of SHD. By looking into the internal

Table 6

Execution time table for baseline serial experiments on 1M and 10M row linear data (H:MM:SS.SS).

Linear Synthetic Dataset	MGC	PCMCI	DBN	Data-level Ensemble MGC	Data-level Ensemble PCMCI	Data-level Ensemble DBN	Algorithm-level Ensemble
1M	0:00:08.16	0:07:23.28	0:27:14.83	0:01:49.78	0:06:35.17	0:28:40.29	0:31:59.37
10M	0:01:21.88	1:31:06.78	5:58:52.31	0:18:52.88	0:54:17.20	3:45:56.39	6:45:01.24

Table 7

Execution time table for baseline serial experiments on 1M and 10M row nonlinear data (H:MM:SS.SS).

Nonlinear Synthetic Dataset	MGC	PCMCI	DBN	Data-level Ensemble MGC	Data-level Ensemble PCMCI	Data-level Ensemble DBN	Algorithm-level Ensemble
1M	0:00:07.83	0:24:07.11	0:24:39.91	0:01:13.25	0:28:00.59	0:27:17.02	0:45:26.36
10M	0:01:24.59	4:33:06.51	5:15:30.22	0:18:15.43	3:57:54.57	2:56:27.57	8:47:18.09

Table 8

Execution time table for parallel experiments on 1M linear data.

Linear 1M	Data-level Parallel Ensemble MGC	Data-level Parallel Ensemble PCMCI	Data-level Parallel Ensemble DBN	Two-phase Ensemble Algorithm-Data	Two-phase Ensemble Data-Algorithm
4 Worker Nodes	0m19.037s	10m51.091s	2m11.193s	3m44.671s	3m55.372s
6 Worker Nodes	0m20.068s	8m57.787s	1m12.336s	2m15.632s	2m54.335s
8 Worker Nodes	0m20.030s	6m46.573s	1m1.355s	2m4.638s	2m4.477s

logic of how causal edges are produced, we find that TCDF method does not find any wrong causes and its attention mechanism is able to find the most important cause regarding each effect. But it is less accurate in identifying other causes, which is one reason for missing some cause-effect edges. The best hyperparameter configuration is *hidden layers* = 6 and *kernel size* = 4, which gives SHD value as 3 for linear 1M dataset and SHD value as 2 for nonlinear 1M dataset.

Since SHD values of our two-phase ensemble approaches are 0 for the datasets, it shows our approach has better learning accuracy than these state-of-art methods on these two synthetic datasets.

6.2.5. Scalability evaluation

We conducted scalability experiments for our proposed two-phase hybrid ensemble causality approaches given different sizes of data sets at a distributed computing environment mentioned above with 5, 7 and 9 compute nodes.

Execution Time. The execution time of all the baseline algorithms is shown in Table 6 and Table 7 for linear and nonlinear, 1M and 10M dataset testing. The execution time for parallel experiments is shown as in Table 8 and Table 9 for 1M and 10M records of linear dataset. For nonlinear dataset, the execution time is recorded as in Table 10 and Table 11 for 1M and 10M data correspondingly.

The execution time for parallel experiments is in Table 8, and Table 9 for linear experiments. The nonlinear experiments execution time tables are as Table 10 and Table 11. The Spark based parallel implementations of the three data-ensemble baseline approaches use the same techniques in Section 5. We measured their execution time as in the left part of parallel experiments execution time tables. We also recorded the execution time of algorithm-data ensemble and data-algorithm ensemble showing in right part of all execution time tables. The experiments are executed under different environment settings using 4, 6 and 8 worker nodes.

For data-level parallel ensemble MGC and DBN, their execution time were shorter than those of our hybrid ensemble approaches because each of them only employed one causality learning algorithm. But for data-level parallel ensemble PCMCI, it is slower than the two-phase ensemble because at the runtime the spark session

encountered idle time for executors in the cluster, thus the computation time is fairly long. However, we did not see the same behavior in the two-phase ensemble experiments. It is the reason our two-phase ensemble execution time is faster than PCMCI based data-level parallel ensemble. The inner reason for this unexpected result will be further investigated. As shown in Table 3, the advantages of our hybrid ensemble approaches over these baseline approaches are overall learning accuracy.

Speedup. By comparing the execution time our parallel two-phase approaches in Tables 8, 9, 10, 11 with the execution time of our serial algorithm ensemble baseline approach in Table 6 and Table 7, we evaluate the speedups of our parallel hybrid ensemble approaches. The algorithm ensemble baseline was executed on a single node and called the three single causality learning algorithms one after another. Since experiments show our data-algorithm ensemble implementation and algorithm-data ensemble implementation have very similar execution time, we only use the execution time of algorithm-data ensemble results to compute speedup. As shown in Figs. 6 and 7, both sets of experiments achieved near linear speed up. Fig. 6 shows the speedups of algorithm-data ensemble in comparison to algorithm ensemble baseline for 10M row linear dataset. With 8 worker nodes, the speed up is more than 32 times. Similarly, Fig. 7 shows speedups of algorithm-data ensemble over algorithm ensemble baseline for 10M nonlinear dataset. Its speedup, when running with 8 worker nodes, reaches 35 times compared to the baseline. Our approaches can achieve better than linear speedup because the time complexity of each baseline algorithm is worse than $O(n)$. For instance, Granger causality algorithm's execution time grows quadratically with the increase of the data record number [12]. By splitting data into N partitions, the execution time for each data partition is less than $1/N$ of the baseline serial approach.

6.3. Experiments with real-world dataset

6.3.1. Dataset description

For real-world dataset, we focus on the problem of warming in the Arctic and polar sea ice declining. Regarding the climate data, we chose the ERA-5 global reanalysis product [39] from 1999 to 2018. ERA-5 was the 1.5 ERA-Interim modeling system, including

Table 9

Execution time table for parallel experiments on 10M linear data.

Linear 10M	Data-level Parallel Ensemble MGC	Data-level Parallel Ensemble PCMCi	Data-level Parallel Ensemble DBN	Two-phase Ensemble Algorithm-Data	Two-phase Ensemble Data-Algorithm
4 Worker Nodes	2m02.216s	51m33.383s	10m46.239s	23m57.703s	24m03.187s
6 Worker Nodes	1m46.703s	35m48.498s	7m55.188s	18m21.925s	18m39.600s
8 Worker Nodes	1m35.964s	22m34.472s	6m46.441s	12m21.128s	12m50.270s

Table 10

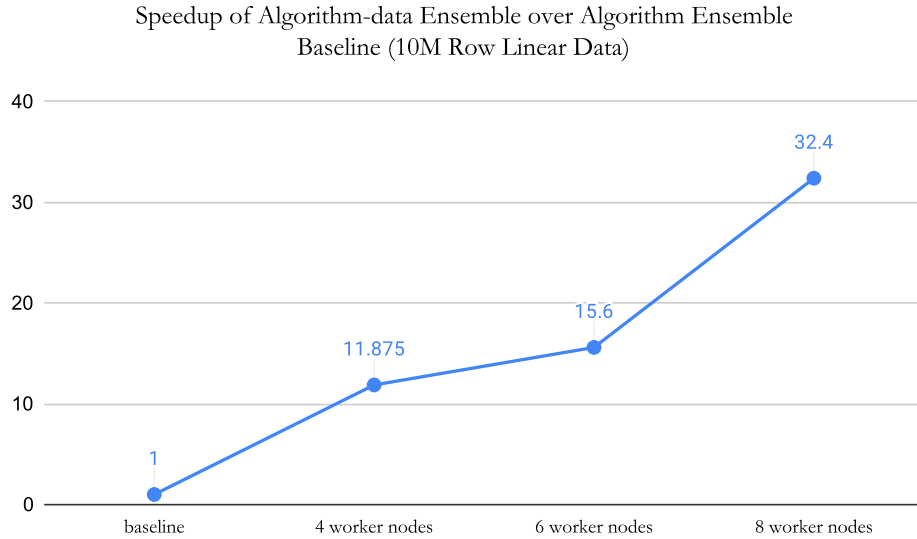
Execution time table for parallel experiments on 1M nonlinear data.

Nonlinear 1M	Data-level Parallel Ensemble MGC	Data-level Parallel Ensemble PCMCi	Data-level Parallel Ensemble DBN	Two-phase Ensemble Algorithm-Data	Two-phase Ensemble Data-Algorithm
4 Worker Nodes	0m20.195s	13m3.590s	2m19.261s	7m29.080s	7m25.565s
6 Worker Nodes	0m19.066s	11m5.580s	1m28.426s	5m30.104s	5m31.534s
8 Worker Nodes	0m18.492s	8m1.691s	1m1.877s	4m6.277s	4m13.503s

Table 11

Execution time table for parallel experiments on 10M nonlinear data.

Nonlinear 10M	Data-level Parallel Ensemble MGC	Data-level Parallel Ensemble PCMCi	Data-level Parallel Ensemble DBN	Two-phase Ensemble Algorithm-Data	Two-phase Ensemble Data-Algorithm
4 Worker Nodes	2m02.023s	39m37.026s	10m46.367s	23m8.639s	24m15.382s
6 Worker Nodes	1m50.979s	28m45.792s	7m52.148s	18m26.094s	18m1.137s
8 Worker Nodes	1m37.207s	25m41.998s	6m54.563s	15m7.396s	16m0.709s

**Fig. 6.** Speedup of algorithm-data ensemble compared to algorithm ensemble baseline for 10M row linear dataset. With 8 worker nodes, the speedup is more than 32 times.

4-dimensional variational analysis (4D-Var) with a running Integrated Forecast System (IFS) model cycle of 31R2 [40]. From ERA-5, users can request lots of atmospheric, land and oceanic climate variables from hourly estimates. The data cover the Earth on a 30 km grid and resolve the atmosphere using 137 levels from the surface up to a height of 80 km. We obtained values of seven variables including 10 meter U wind component, 10 meter V wind component, total cloud cover, mean sea level pressure, total precipitation, shortwave radiation, and longwave radiation [41]. We further pre-process the variables by combining shortwave radiation and long wave radiation as total radiation variable, and compute the 10 meter U wind component, 10 meter V wind component to get the wind speed variable. Each variable has 3 dimensions (7307 x 360 x 180) for day, longitude and latitude respectively. The analysis was focused on the daily data spanning 20 years, with more than 300 GB data. Then we computed the average at Arctic region (latitude > 60°N) for each variable based on the global data. One

more variable, the total sea ice extent variable was selected as the Arctic sea ice index from National Snow & Ice Data Center [42]. In total, our real-world experiment input dataset contains six variables. In the experiments, the maximum lag is chosen as 14, as two weeks.

6.3.2. Consistency evaluation

In experiments with our real-world dataset, the data partition number is set to 4, which means 5 years as a chunk. The reason is that the causal relationships of the selected atmospheric variables should be valid in a long-term and stable environment. By doing this, we can also mute the impacts of extreme weather events on those causal relations in a specific year (e.g., El Niño).

Result consistency with different individual causality discovery methods. Similar to the result consistency analysis for synthetic datasets, Table 12 shows the result consistency using matrix similarities on our real-world dataset experiments. From this ta-

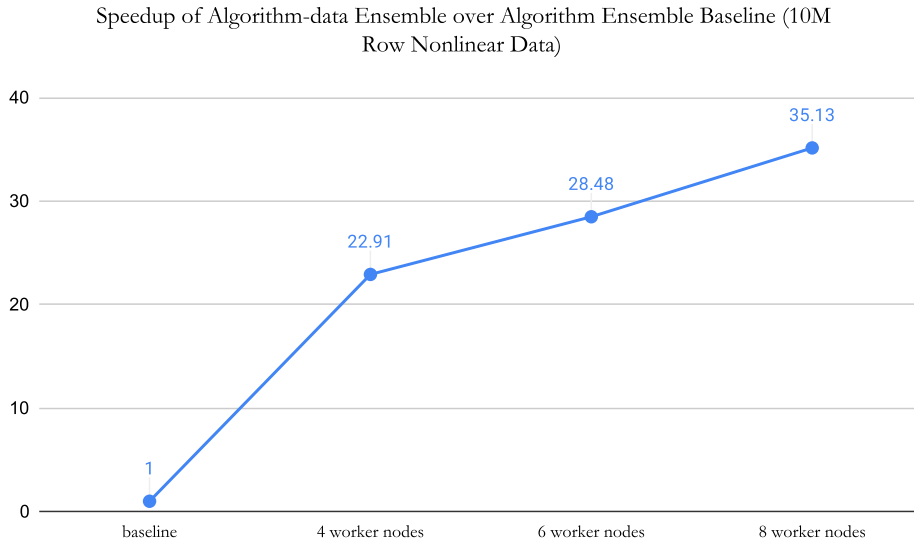


Fig. 7. Speedup of algorithm-data ensemble compared to algorithm ensemble baseline for 10M row nonlinear dataset. With 8 worker nodes, the speedup is more than 35 times.

Table 12
Result matrix similarity of real world dataset.

Matrix Similarities	GC	PCMCi	DBN	Data-level Ensemble GC	Data-level Ensemble PCMCi	Data-level Ensemble DBN	Algorithm-level Ensemble	Two-phase Algorithm-data Ensemble	Two-phase Data-algorithm Ensemble
GC	1.000	0.600	0.111	0.889	0.621	0.105	1	0.842	0.842
PCMCi	0.600	1.000	0.067	0.533	0.967	0.100	0.6	0.567	0.567
DBN	0.111	0.067	1.000	0.125	0.069	0.667	0.111	0.118	0.118
Data-level Ensemble GC	0.889	0.533	0.125	1.000	0.552	0.118	0.889	0.941	0.941
Data-level Ensemble PCMCi	0.621	0.967	0.069	0.552	1.000	0.103	0.621	0.587	0.587
Data-level Ensemble DBN	0.105	0.010	0.667	0.118	0.103	1.000	0.105	0.176	0.176
Algorithm-level Ensemble	1.000	0.600	0.111	0.889	0.620	0.105	1.000	0.842	0.842
Two-phase Algorithm-data Ensemble	0.842	0.567	0.118	0.941	0.586	0.176	0.842	1.000	1.000
Two-phase Data-algorithm Ensemble	0.842	0.567	0.118	0.941	0.586	0.176	0.842	1.000	1.000

ble, our observations are as follows. First, the individual causality discovery methods still generate quite different results with only 0.259 similarities on average. Second, the data-level ensemble has different results from the algorithm-level ensemble, with the average similarity only being 0.538. Third, our proposed two-phase ensemble methods output identical results, which meets the same observation with synthetic dataset experiments.

6.3.3. Accuracy evaluation

To discover causal relations between different variables, the climate scientists normally conduct interventions or real-life experiments by manipulating the value of a target variable. Since the atmosphere is a highly chaotic and non-linear system, it would be quite challenging to quantify these relationships, making the accuracy evaluation for each method much more difficult. Here, we qualitatively determine whether those results are reasonable or not.

Fig. 8 demonstrates the domain knowledge graph based on the literature in Earth Science. First, there are two-way interactions between large-scale circulation variability and sea ice changes, which are mainly represented by edges ($msl \leftrightarrow sea\ ice$) and ($wind\ speed \leftrightarrow sea\ ice$) in the figure. A dipole pattern in the mean sea level

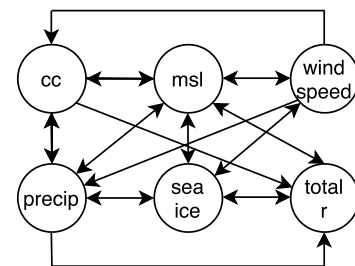


Fig. 8. Domain knowledge graph for real-world dataset experiments based on Earth Science literature with variables as cloud cover (cc), mean sea level pressure (msl), wind speed, precipitation (precip), sea ice extent (sea ice) and total surface radiation (total r).

pressure trend in the Arctic was found to drive more sea ice out of the Arctic dynamically and enhance sea ice melt by promoting transport of heat and moisture [43–46]. In the meantime, the sea ice retreat and increase in open water area can directly modify the large-scale circulation patterns, including mean sea level pressure and wind speed [47]. The mean sea level pressure and wind speed also actively interact with each other, shown as edge (msl

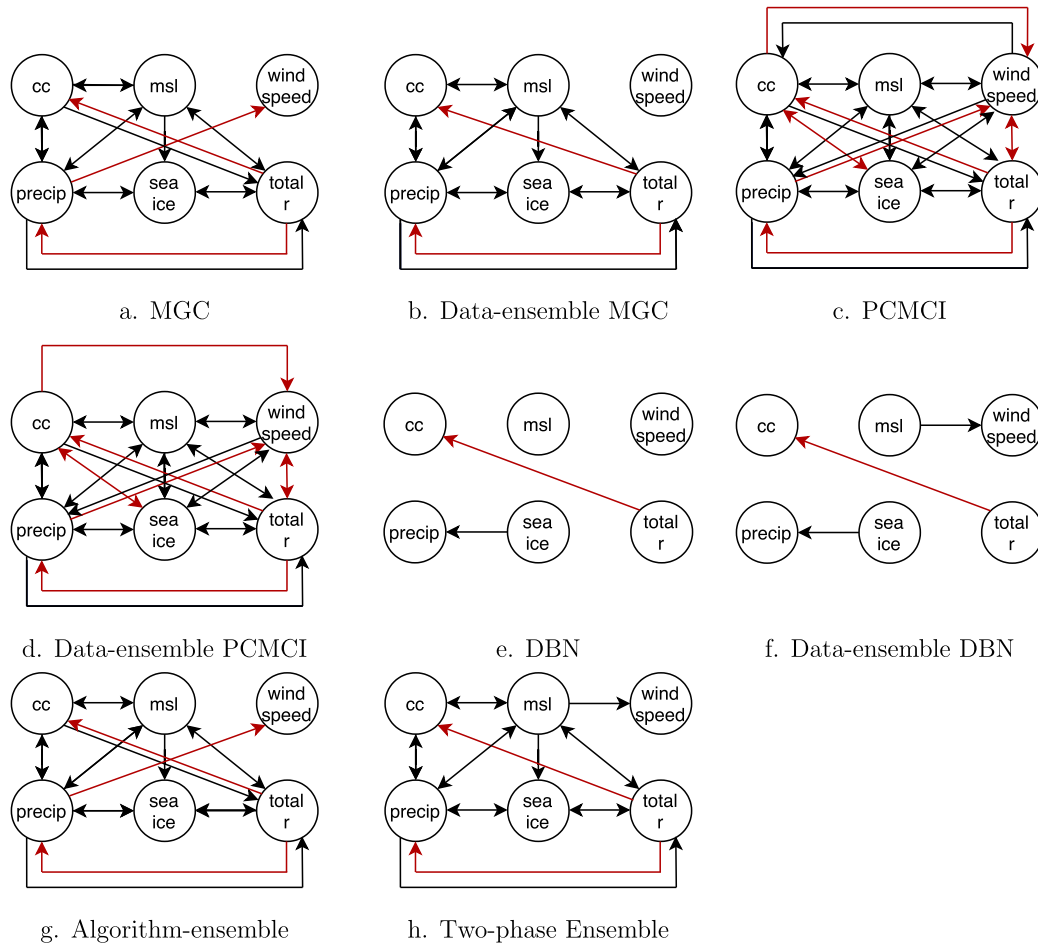


Fig. 9. Causality discovery results for real-world dataset experiments with variables as cloud coverage (cc), mean sea level pressure (msl), wind speed, precipitation (precip), sea ice extent (sea ice) and total radiation (total r). Red arrows mean wrong edges based on domain knowledge. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

\leftrightarrow wind speed). In addition to dynamical processes, the thermodynamical processes are also found to affect Arctic sea ice variations. The downward longwave radiation at the surface dominates surface warming and therefore enhances sea ice melt in winter and spring, while the shortwave radiation becomes more important for sea ice variations in summer [48,49]. Sea ice melt/growth can also modulate the surface radiation by changing the surface albedo, emissivity and temperature [50]. These processes are represented by edge (sea ice \leftrightarrow total r). Moreover, edge (precip \rightarrow sea ice) indicates that the magnitude and phase of the precipitation change the surface albedo and thus modulate the rate of sea ice melt and growth [51,52]. The precipitation in turn could be increased by enhanced local evaporation due to sea ice melt [53], shown as edge (sea ice \rightarrow precip). In the meantime, the atmospheric variables are connected with each other through various processes. The edge (cc \leftrightarrow precip) represents multiple cloud microphysical processes [54,55] for the conversion between rain drops and cloud water droplets/ice crystals, such as autoconversion and accretion. The large-scale circulation such as mean sea level pressure and wind speed would help to form or dissipate clouds; the clouds in turn modulate the dynamical or thermodynamical structure of the troposphere [56,54]. These processes are indicated by edges (msl \leftrightarrow cc) and (wind speed \rightarrow cc). As for the precipitation, the large-scale circulation could either be a source or sink, which are represented by edges (msl \leftrightarrow precip) and (wind speed \rightarrow precip) [57,56]. Here, we only consider the immediate effects of cloud and precipitation on mean sea level pressure, but their impacts on wind fields over a large domain remain uncertain. Therefore, only

one direction arrows are plotted in edges (wind speed \rightarrow precip) and (wind speed \rightarrow cc). Furthermore, the mean sea level pressure, precipitation and cloud cover exert a large influence on surface radiation through the changes in temperature, emissivity, scattering and absorption, etc. [58], shown as edges (msl \rightarrow total r), (precip \rightarrow total r) and (cc \rightarrow total r). Here, we also consider the direct impacts of surface radiation of changes in mean sea level pressure (total r \rightarrow msl), while the indirect impacts of cloud cover and precipitation on surface radiation are neglected.

The results of real-world dataset experiments are shown as in Fig. 9. The red edges in the figure are considered unrealistic based on Earth Science knowledge. To compare the results in Fig. 9 and the domain knowledge graph in Fig. 8, we calculate SHD scores between every result of each approach and the domain knowledge graph. We also calculate the precision, recall and F1 score for each approach to compare their learning accuracy. As shown in Table 13, our two-phase ensemble approach ranks 2nd (next to PCMCI) in terms of SHD, and ranks 1st of precision. For recall and F1 score, our two-phase ensemble approach gets 3rd ranking because PCMCI and data-ensemble PCMCI find many more edges. It shows our approach also performs very well in discovering causality from our real-world dataset.

From Table 3 and Table 13, one interesting finding is that no individual method is always the best for the five datasets we experimented with. For instance, accuracies of MGC and PCMCI are much better than DBN in linear 1M, 10M and real-world dataset, but they performed much worse than DBN for nonlinear 1M, 10M synthetic data. By doing ensemble from these divergent algorithms,

Table 13

Accuracy comparison of single causality discovery approach, baseline ensemble approach, and our proposed two-phase hybrid causality approach on real-world dataset (best results are in bold).

	MGC	PCMC1	DBN	Data-level Ensemble MGC	Data-level Ensemble PCMC1	Data-level Ensemble DBN	Algorithm- level Ensemble	Two-Phase Hybrid Ensemble
SHD	10	8	22	10	9	21	10	9
Precision	0.833	0.733	0.500	0.875	0.724	0.666	0.833	0.882
Recall	0.681	1.000	0.045	0.636	0.954	0.090	0.681	0.681
F1	0.750	0.846	0.083	0.737	0.824	0.160	0.750	0.770

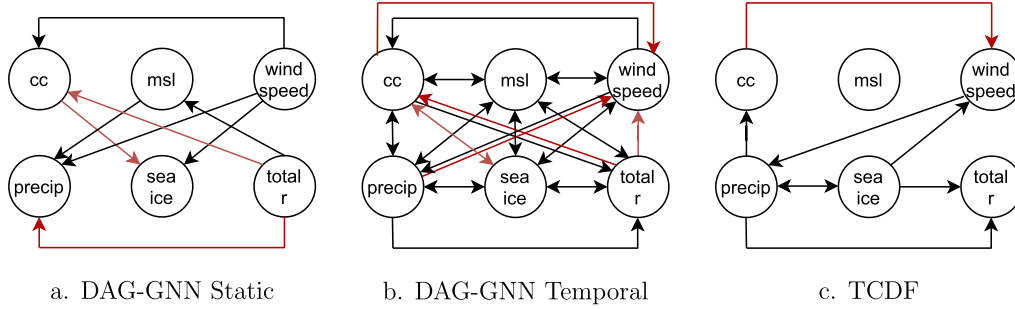


Fig. 10. Causality discovery results with deep learning methods for real-world dataset experiments with variables as cloud coverage (cc), mean sea level pressure (msl), wind speed, precipitation (precip), sea ice extent (sea ice) and total radiation (total r). Red arrows mean wrong edges based on domain knowledge.

our two-phase ensemble approaches show very good generalization and outperform almost all individual methods in all experiments.

6.3.4. Comparison with other causality discovery methods

Similar to the comparison in Section 6.2.4, we conduct experiments on real-world sea-ice dataset using DAG-GNN and TCDF methods, with results showing in Fig. 10 and Table 14. Besides, in the experiments, different hyperparameter configurations have been fed into the model as in Section 6.2.4 and we only present the best performance results here. For DAG-GNN approach, among $(t = 0.3, \tau = 0)$, $(t = 0.3, \tau = 1e-07)$, $(t = 0.2, \tau = 0)$ and $(t = 0.2, \tau = 1e-07)$, the best result was achieved with hyperparameters $t = 0.3$ and $\tau = 0$. For TCDF approach, since the maxlag is 14, kernel size = 15 is fixed. Different values of hidden layer, namely 0, 1, 2 and 3, are tested. The best result comes from hidden layer = 2 and kernel size = 15. Also, static and temporal results are generated using DAG-GNN method. We still use the domain knowledge graph (Fig. 8) as the ground truth to calculate the evaluation metrics.

As shown in Fig. 10, the DAG-GNN Temporal finds the largest numbers of edges among three methods, which are comparable with our proposed ensemble methods. It seems that DAG-GNN Temporal tends to detect the two-way interactions between all variables. In the meantime, it also detects some unrealistic edges, for example, total r \rightarrow wind speed, cc \rightarrow wind speed, and cc \leftrightarrow sea ice. In comparison, DAG-GNN Static is less efficient to detect those connections, as it produces much fewer edges than DAG-GNN Temporal and also generates three unrealistic edges. The TCDF only finds one unrealistic edge, however, it generates the smallest number of true positive edges compared to other two deep learning methods and all ensemble methods.

From the results in Table 14, our two-phase hybrid ensemble has highest precision, and ranks 2nd in SHD, recall and F1. DAG-GNN Temporal has the lowest SHD, highest recall, and highest F1 score. The recall of DAG-GNN Temporal is high because it finds all causal edges in the domain knowledge causal graph (Fig. 8). Besides, TCDF ranks 2nd in precision, which means it is also good at finding true positive edges. However, its recall is relatively low because it only produces very few edges. DAG-GNN Static performs

worst in every metric in real-world experiments, and exposes the shortcoming of directly apply time-series data as iid data.

6.3.5. Scalability evaluation

Execution Time. The execution time of our proposed two approaches on real-world dataset under 4, 6 and 8 worker nodes is shown in Table 15. Since we are using daily data, the testing maximum lag is set to be 30, meaning a month. Usually the real-world data are nonlinear, so we applied our nonlinear two-phase approaches on this dataset. The results show that our approaches also have good scalability on causality discovery with large lags.

Speedup. The algorithm ensemble baseline program was executed for real-world data, and the wall time is around 3.5 hours. So the speedup of two-phase algorithm-data ensemble is done by comparing to the baseline as in Fig. 11. The shape of the chart is similar to that in Fig. 7. From the figure, the speedup with 8 worker nodes achieved near seven times speedup compared to the baseline. The speedup is not as significant as that of nonlinear synthetic experiment since the partition number is four, which is less than that of 48 in the nonlinear synthetic experiment, thus the level of parallelization is relatively less.

7. Related work

There have been many studies on ensemble learning and scalable/parallel machine learning. But we believe our work is the first study dealing with both algorithm variety and data volume for causality discovery. We also did not find many studies directly on ensemble learning for causality. Because causality graph can be categorized as a type of probabilistic graphic model, we first discuss and compare with related work on ensemble learning for probabilistic graphic models in the first subsection. In the second subsection, we further discuss and compare additional big data parallel ensemble learning work beyond probabilistic graphic models.

7.1. Ensemble learning for probabilistic graphic models

To achieve probabilistic graphical model ensemble, using the three categories explained in Section 3, existing ensemble learning approaches can also be categorized into 1) algorithm ensemble

Table 14

Accuracy comparison of DAG-GNN (Static, Temporal), TCDF and our proposed two-phase hybrid causality approach on real-world dataset (best results are in bold).

	DAG-GNN Static	DAG-GNN Temporal	TCDF	Two-Phase Hybrid Ensemble
SHD	20	6	16	9
Precision	0.625	0.786	0.875	0.882
Recall	0.227	1.000	0.318	0.681
F1	0.333	0.880	0.467	0.770

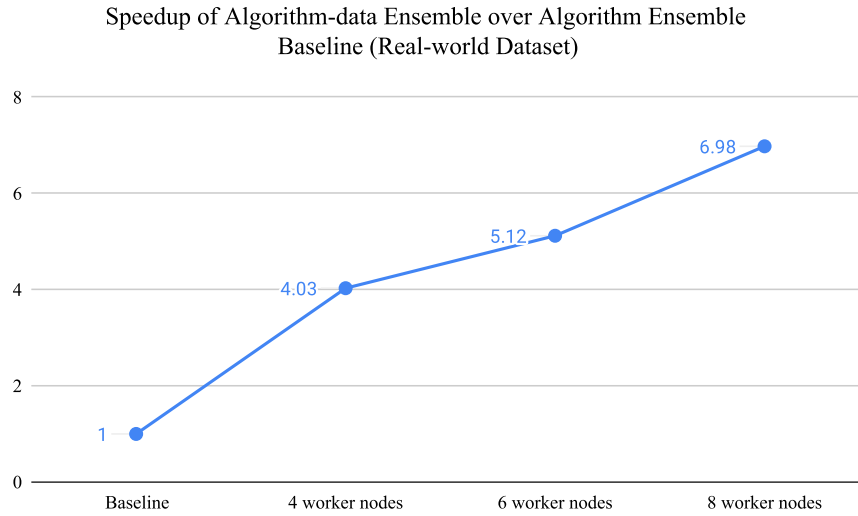


Fig. 11. Speedup of algorithm-data ensemble compared to algorithm ensemble baseline for real-world dataset. With 8 worker nodes and 4 partitions, the speed up is nearly seven times.

Table 15

Execution time table for parallel experiments on real-world data (H:MM:SS).

Real-world Dataset	Two-phase Ensemble Algorithm-Data	Two-phase Ensemble Data-Algorithm
4 Worker Nodes	0:52:52	0:54:12
6 Worker Nodes	0:41:17	0:43:21
8 Worker Nodes	0:30:43	0:31:55

for work at [59], 2) data ensemble work at [60–63], and 3) hybrid ensemble for both data and algorithm at [64].

In algorithm ensemble category, [59] supports parallel ensemble learning of multiple classifiers on the same data. Both horizontal and vertical parallelization are implemented in the paralleled PC algorithm. As a data ensemble approach, [60] first splits the training data, then trains Bayesian sub-networks in parallel, finally does boosting as ensemble method on the trained sub-networks to get the learning result. [61] is also a data ensemble approach for Bayesian network learning from big datasets to achieve better scalability and accuracy. [62] studies how to conduct data ensemble learning based linear causal model discovery and mainly uses bagging ensemble [24] to get different data samples for a specific causal discovery algorithm and later used different voting mechanism to combine the results. [63] extends [62] by considering additional ensemble strategies including Adaboost [65] and GASEN [66]. Similar to our findings, their work also shows ensemble learning could achieve more accurate causality discovery results. Because the dataset used in [62] and [63] is rather small, they did not discuss partitioning based data ensemble. As a hybrid ensemble approach, [64] conducts two-phase (algorithm ensemble for each data partition and data ensemble for multiple data partitions) Bayesian network ensemble learning. The main differences of the work in this paper and [64] are 1) our algorithm-level ensemble belongs to heterogeneous ensemble because each learning algorithm uses its own causality discovery models, while [64] belongs to heterogeneous ensemble with different learning algo-

gorithms of the same Bayesian network model; 2) this paper targets causality discovery instead of Bayesian network learning; 3) this paper supports two types of two-phase ensemble, namely both data-algorithm hybrid ensemble and algorithm-data hybrid ensemble, while [64] only supports data-algorithm ensemble.

7.2. Parallel ensemble learning in big data systems

Besides the probabilistic graphic model related ensemble studies in the previous subsection, most other big data parallel ensemble learning algorithms are tree based where different trees can be trained in parallel with a data subset, then results from multiple trees are ensembled via majority voting (e.g., [67]) or tree boosting (e.g., XGBoost [22]). There are two main approaches of data partitioning: horizontal data partitioning based on rows and vertical data partitioning based on columns.

[67] contains horizontal data partitioning and parallel learning among the data partitions. Input data is first partitioned vertically to divide training data features to independent subsets. Then each task loads the data from one feature subset to train an independent tree and multiple trees can be trained in parallel. For XGBoost, PLANET [68] and COMET [69], parallel training is done via horizontally partitioned data and they differ in how different trees are ensembled.

As a comparison, parallelization in our hybrid ensemble approaches is done via horizontal data partitioning because all features are needed for each training and our data has time dependency. Further, multiple learning algorithms are employed for each data partition in our algorithm-data ensemble while the above related work only employs the same learning algorithm for different data partitions.

8. Conclusions

Causality discovery is a fundamental research topic in many disciplines and discovered cause-effect relationships can help explain

why a system has certain behavior or state. Nowadays, data-driven causality discovery faces two challenges: 1) the large volume of datasets to be learned from and 2) the variety of causality discovery algorithms. To deal with these two challenges, this paper proposes a flexible two-phase ensemble causality discovery framework and two approaches for scalable and hybrid ensemble learning. Experiments show our algorithms outperform baseline ones in terms of both accuracy and execute time in most cases.

For future work, we will focus on the following aspects. First, we will extend the work to further enable ensemble of time lag and probability of causal edges. Second, we will study how to better select and merge results from many available individual causality learning algorithms, i.e. measuring individual learner diversity and weighted majority voting, for better ensemble result accuracy. Further, we plan to investigate whether and how other ensemble approaches, such as boosting ensemble [25,70] could help better causality discovery.

Contribution

Pei Guo: Writing - Original Draft, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Visualization.

Yiyi Huang: Validation, Data Curation.

Jianwu Wang: Conceptualization, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by grant CyberTraining: DSE: Cross-Training of Researchers in Computing, Applied Mathematics and Atmospheric Sciences using Advanced Cyberinfrastructure Resources (OAC-1730250), and grant CAREER: Big Data Climate Causality Analytics (OAC-1942714) from the National Science Foundation. The execution environment is provided through the High Performance Computing Facility at UMBC.

References

- [1] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed., Cambridge University Press, New York, NY, USA, 2009.
- [2] R. Guo, L. Cheng, J. Li, P.R. Hahn, H. Liu, A survey of learning causality with data: problems and methods, *ACM Comput. Surv.* 53 (2020) 1–37.
- [3] C.W. Granger, Investigating causal relations by econometric models and cross-spectral methods, *Econometrica* 37 (1969) 424–438.
- [4] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, D. Sejdinovic, Detecting causal associations in large nonlinear time series datasets, <https://arxiv.org/abs/1702.07007v2>, 2018. (Accessed 28June2018).
- [5] K.P. Murphy, S. Russell, *Dynamic Bayesian Networks: Representation, Inference and Learning*, 2002.
- [6] H. Ye, E.R. Deyle, L.J. Gilarranz, G. Sugihara, Distinguishing time-delayed causal interactions using convergent cross mapping, *Sci. Rep.* 5 (2015) 14750.
- [7] H. Song, J. Tian, J. Huang, P. Guo, Z. Zhang, J. Wang, Hybrid causality analysis of ENSO's global impacts on climate variables based on data-driven analytics and climate model simulation, *Front. Earth Sci.* 7 (2019) 233, <https://doi.org/10.3389/feart.2019.00233>, <https://www.frontiersin.org/article/10.3389/feart.2019.00233>.
- [8] N.J. Holbrook, J.N. Brown, J. Davidson, M. Feng, A.J. Hobday, J.M. Lough, S. McGregor, S.B. Power, J.S. Riseby, El Niño–Southern Oscillation, 2012.
- [9] C. Zou, K.J. Denby, J. Feng, Granger causality vs. dynamic Bayesian network inference: a comparative study, *BMC Bioinform.* 10 (2009) 122.
- [10] S. Hussung, S. Mahmud, A. Sampath, M. Wu, P. Guo, J. Wang, Evaluation of Data-driven Causality Discovery Approaches among Dominant Climate Modes, Technical Report HPCF-2019-12, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2019.
- [11] J.T. Overpeck, G.A. Meehl, S. Bony, D.R. Easterling, Climate data challenges in the 21st century, *Science* 331 (2011) 700–702.
- [12] A. Arnold, Y. Liu, N. Abe, Temporal causal modeling with graphical granger methods, in: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07*, ACM, New York, NY, USA, 2007, pp. 66–75.
- [13] L. Rokach, Ensemble-based classifiers, *Artif. Intell. Rev.* 33 (1–2) (2010) 1–39.
- [14] Homepage — Apache Spark Project, <http://spark.apache.org>, 2020. (Accessed 30May2020).
- [15] Homepage — Apache Flink Project, <http://flink.apache.org>, 2020. (Accessed 1March2020).
- [16] Scalable Ensemble Learning for Causality Discovery, https://github.com/big-data-lab-umbc/ensemble_causality_learning, 2020. (Accessed 28May2020).
- [17] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, *Commun. ACM* 51 (2008) 107–113.
- [18] P. Guo, A. Ofonedu, J. Wang, Scalable and hybrid ensemble-based causality discovery, in: *Proceedings of the IEEE International Conference on Smart Data Services, SDDS 2020*, IEEE, 2020.
- [19] Homepage — SIGKDD Cup Achieve, <https://www.kdd.org/kdd-cup>, 2019. (Accessed 1March2020).
- [20] R.E. Schapire, The strength of weak learnability, *Mach. Learn.* 5 (1990) 197–227.
- [21] A. Liaw, M. Wiener, et al., Classification and regression by randomForest, *R News* 2 (2002) 18–22.
- [22] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 785–794.
- [23] D.H. Wolpert, Stacked generalization, *Neural Netw.* 5 (1992) 241–259.
- [24] L. Breiman, Bagging predictors, *Mach. Learn.* 24 (1996) 123–140.
- [25] Y. Freund, R.E. Schapire, et al., Experiments with a new boosting algorithm, in: *ICML*, vol. 96, Citeseer, 1996, pp. 148–156.
- [26] R. Polikar, Ensemble learning, in: *Ensemble Machine Learning*, Springer, 2012, pp. 1–34.
- [27] H. Luetkepohl, The New Introduction to Multiple Time Series Analysis, <https://doi.org/10.1007/978-3-540-27752-1>, 2005.
- [28] Gábor J. Székely, Maria L. Rizzo, Nail K. Bakirov, Measuring and testing dependence by correlation of distances, *Ann. Stat.* 35 (2007).
- [29] I. Ben-Gal, Bayesian Networks, *Encyclopedia of Statistics in Quality and Reliability*, John Wiley & Sons, 2007.
- [30] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, *Mach. Learn.* 9 (1992) 309–347.
- [31] M. Wang, W. Fu, X. He, S. Hao, X. Wu, A survey on large-scale machine learning, *IEEE Trans. Knowl. Data Eng.* (2020).
- [32] L. Zhou, S. Pan, J. Wang, A.V. Vasilakos, Machine learning on big data: opportunities and challenges, *Neurocomputing* 237 (2017) 350–361.
- [33] The UMBC High Performance Computing Facility (HPCF), <https://hpcf.umbc.edu/>, 2020. (Accessed 1March2020).
- [34] M.A. Jette, A.B. Yoo, M. Grondona, Slurm: simple Linux utility for resource management, in: *Proceedings of Job Scheduling Strategies for Parallel Processing, JSSPP 2003*, in: *Lecture Notes in Computer Science*, Springer-Verlag, 2002, pp. 44–60.
- [35] I. Tsamardinos, L.E. Brown, C.F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, *Mach. Learn.* 65 (1) (2006) 31–78.
- [36] Y. Yu, J. Chen, T. Gao, M. Yu, DAG-GNN: DAG structure learning with graph neural networks, in: *Proceedings of the 36th International Conference on Machine Learning*, 2019, Code available on, <https://github.com/fishmoon1234/DAG-GNN>.
- [37] M. Nauta, D. Bucur, C. Seifert, Causal discovery with attention-based convolutional neural networks, *Mach. Learn. Knowl. Extr.* 19 (2019), Code available on, <https://github.com/M-Nauta/TCDF>.
- [38] J. Peters, D. Janzing, B. Schölkopf, Causal inference on time series using restricted structural equation models, in: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems*, vol. 26, Curran Associates, Inc., 2013, <https://proceedings.neurips.cc/paper/2013/file/47d1e990583c9c67424d369f3414728e-Paper.pdf>.
- [39] C.C.C.S. C3S, ERA5: fifth generation of ECMWF atmospheric reanalyses of the global climate, in: *Technical Report, Copernicus Climate Change Service Climate Data Store (CDS)*, 2017, <https://cds.climate.copernicus.eu/cdsapp#!/home>.
- [40] D.P. Dee, S.M. Uppala, A.J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M.A. Balsamed, G. Balsamo, P. Bauer, P. Bechtold, A.C.M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A.J. Geer, L. Haimberger, S.B. Healy, H. Hersbach, E.V. Hölm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A.P. McNally, B.M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavalato, J.-N. Thépaut, F. Vitart, The era-interim reanalysis: configuration and performance of the data assimilation system, *Q. J. R. Meteorol. Soc.* 137 (2011) 553–597, <https://doi.org/10.1002/qj.828>.
- [41] ERA5 hourly data on single levels from 1979 to present, <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels>, 2020. (Accessed 31May2020).
- [42] National Snow and Ice Data Center, <https://nsidc.org/>, 2020. (Accessed 5October2020).

- [43] J. Wang, J. Zhang, E. Watanabe, M. Ikeda, K. Mizobata, J.E. Walsh, X. Bai, B. Wu, Is the dipole anomaly a major driver to record lows in Arctic summer sea ice extent?, *Geophys. Res. Lett.* 36 (2009).
- [44] M.-L. Kapsch, N. Skific, R.G. Graversen, M. Tjernström, J.A. Francis, Summers with low Arctic sea ice linked to persistence of spring atmospheric circulation patterns, *Clim. Dyn.* 52 (2019) 2497–2512.
- [45] Y. Huang, X. Dong, B. Xi, Y. Deng, A survey of the atmospheric physical processes key to the onset of Arctic sea ice melt in spring, *Clim. Dyn.* 52 (2019) 4907–4922.
- [46] E. Watanabe, J. Wang, A. Sumi, H. Hasumi, Arctic dipole anomaly and its contribution to sea ice export from the Arctic Ocean in the 20th century, *Geophys. Res. Lett.* 33 (2006).
- [47] J.E. Overland, M. Wang, Large-scale atmospheric circulation changes are associated with the recent loss of Arctic sea ice, *Tellus, Ser. A Dyn. Meteorol. Oceanogr.* 62 (2010) 1–9.
- [48] Y. Huang, X. Dong, B. Xi, E.K. Dolinar, R.E. Stanfield, The footprints of 16 year trends of Arctic springtime cloud and radiation properties on September sea ice retreat, *J. Geophys. Res., Atmos.* 122 (2017) 2179–2193.
- [49] M.-L. Kapsch, R.G. Graversen, M. Tjernström, Springtime atmospheric energy transport and the control of Arctic summer sea-ice extent, *Nat. Clim. Change* 3 (2013) 744–748.
- [50] J.E. Kay, A. Gettelman, Cloud influence on and response to seasonal Arctic sea ice loss, *J. Geophys. Res., Atmos.* 114 (2009).
- [51] M. Sturm, J. Holmgren, D.K. Perovich, Winter snow cover on the sea ice of the Arctic Ocean at the Surface Heat Budget of the Arctic Ocean (SHEBA): temporal evolution and spatial variability, *J. Geophys. Res., Oceans* 107 (2002), SHE–23.
- [52] D. Perovich, T. Grenfell, B. Light, P. Hobbs, Seasonal evolution of the albedo of multiyear Arctic sea ice, *J. Geophys. Res., Oceans* 107 (2002), SHE–20.
- [53] R. Bintanja, F. Selten, Future increases in Arctic precipitation linked to local evaporation and sea-ice retreat, *Nature* 509 (2014).
- [54] M.K. Yau, R.R. Rogers, *A Short Course in Cloud Physics*, Elsevier, 1996.
- [55] H.R. Pruppacher, J.D. Klett, Microphysics of clouds and precipitation, *Nature* 284 (1980) 88.
- [56] J.M. Wallace, P.V. Hobbs, *Atmospheric Science: An Introductory Survey*, Vol. 92, Elsevier, 2006.
- [57] J. Holton, G. Hakim, *An Introduction to Dynamic Meteorology*, 5th edn., Academic Press, New York, 2013.
- [58] K.-N. Liou, *An Introduction to Atmospheric Radiation*, Elsevier, 2002.
- [59] A.L. Madsen, F. Jensen, A. Salmerón, H. Langseth, T.D. Nielsen, A parallel algorithm for Bayesian network structure learning from large data sets, *Knowl.-Based Syst.* 117 (2017) 46–55, <https://doi.org/10.1016/j.knosys.2016.07.031>.
- [60] J. Hu, G. Wu, P. Sun, Q. Xiong, A parallel Bayesian network learning algorithm for classification, in: 2016 7th IEEE International Conference on Software Engineering and Service Science, ICSESS, IEEE, 2016, pp. 259–263.
- [61] J. Wang, Y. Tang, M. Nguyen, I. Altintas, A scalable data science workflow approach for big data Bayesian network learning, in: Proceedings of the 2014 IEEE/ACM International Symposium on Big Data Computing, BDC 2014, 2014, pp. 16–25.
- [62] H. Dai, G. Li, Z.-H. Zhou, Ensembling MML causal discovery, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2004, pp. 260–271.
- [63] G. Li, H. Dai, Study of ensemble strategies in discovering linear causal models, in: International Conference on Fuzzy Systems and Knowledge Discovery, Springer, 2005, pp. 368–377.
- [64] Y. Tang, J. Wang, M. Nguyen, I. Altintas, Penbayes: a multi-layered ensemble approach for learning Bayesian network structure from big data, *Sensors* 19 (2019), <https://doi.org/10.3390/s19204400>, <https://www.mdpi.com/1424-8220/19/20/4400>.
- [65] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *J. Comput. Syst. Sci.* 55 (1997) 119–139.
- [66] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, *Artif. Intell.* 137 (2002) 239–263.
- [67] J. Chen, K. Li, Z. Tang, K. Bilal, S. Yu, C. Weng, K. Li, A parallel random forest algorithm for big data in a spark cloud computing environment, *IEEE Trans. Parallel Distrib. Syst.* 28 (2017) 919–933.
- [68] B. Panda, J.S. Herbach, S. Basu, R.J. Bayardo, Planet: massively parallel learning of tree ensembles with MapReduce, in: Proceedings of the 35th International Conference on Very Large Data Bases, VLDB-2009, 2009, <http://www.bayardo.org/ps/vldb2009.pdf>.
- [69] J.D. Basilico, M.A. Munson, T.G. Kolda, K.R. Dixon, W.P. Kegelmeyer, Comet: a recipe for learning and using large ensembles on massive data, in: 2011 IEEE 11th International Conference on Data Mining, 2011.
- [70] P. Guo, C. Liu, Y. Tang, J. Wang, Parallel gradient boosting based granger causality learning, in: 2019 IEEE International Conference on Big Data, Big Data, IEEE, 2019, pp. 2845–2854.