MCNS: Mining Causal Natural Structures Inside Time Series via A Novel Internal Causality Scheme

Zihan Jiang[1,2], **Dehui Du**[1,2], **Yuanhao Liu**[1,2]

[1]Software Engineering Institute
[2]East China Normal University, Shanghai, China
dhdu@sei.ecnu.edu.cn

Abstract

Causal inference permits us to discover covert relationships of various variables in time series. However, in most existing works, the variables mentioned above are the dimensions. The causality between dimensions could be cursory, which hinders the comprehension of the internal relationship and the benefit of the causal graph to the neural networks (NNs). In this paper, we find that causality exists not only *outside* but also *inside* the time series because it implies the succession of events in the real world. It inspires us to seek the relationship between internal subsequences. However, the challenges are the hardship of discovering causality from subsequences and utilizing the causal natural structures to improve Neural Networks. To address these challenges, we propose a novel framework called Mining Causal Natural Structure (*MCNS*), which is *automatic* and *domain-agnostic* and helps to find the causal natural structures inside time series via the internal causality scheme. We evaluate the *MCNS* framework and integrate NN with *MCNS* on time series classification tasks. Experimental results illustrate that our impregnation, by refining attention, shape selection classification, and pruning datasets, drives NN, even the data itself preferable accuracy and interpretability. Besides, *MCNS* provides an in-depth, solid summary of the time series and datasets.

Introduction

Time series data, such as medical electrocardiograms and financial data, have played an essential role in society. Furthermore, the possibility of making causal inferences [Mastakouri and Schölkopf 2020; Li et al. 2022] in time series data greatly appeals to social and behavioural scientists and has been widely used in a plethora of applications. However, classical causal discovery [Granger 1969] approaches in time series usually treat the time series as a whole, and problematic to find causal relationships inside time series.

A rich body of research has been proposed to seek causal relations in structured multivariate time series data. For example, most works suggested leveraging the concept of *Granger* causality [Huang et al. 2019; Schamberg and Coleman 2019; Mastakouri, Schölkopf, and Janzing 2021], and some other works proposed to rely on the idea of *Pearl* causality in i.i.d multivariate time series data [Gerhardus and Runge 2020; Bica, Alaa, and Van Der Schaar 2020]. In particular, Liu et al. [Liu et al. 2024] introduced the Causal-Temporal Attention Network (CTAttn) to integrate causal relationships into multivariate time series forecasting. Their approach combines causal discovery and attention mechanisms to improve forecasting accuracy and interpretability by modeling inter-variable causal dependencies. But those works focused on relations between dimensions, and we find that the causal relationships need to be more profound and in-depth. There is causality not only outside (as a whole) but also inside the time series (in the subsequence). The causal natural structure inside the time series is crucial for causal inference.

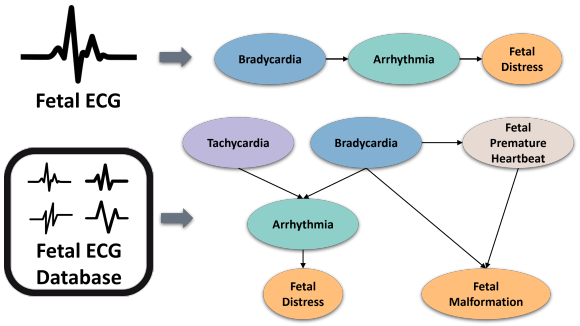

Figure 1: An example of a causal natural structure obtained from *MCNS* for the fetal ECG. The above is from a specific fetal ECG, and the below is from the whole dataset.

Actually, discovering causal relationships inside time series is also valuable or vital for making decisions. For instance, when a medical AI system assists doctors in dealing with the classification of diseases in the fetal electrocardiogram (ECG), causal inference could help to figure out the exact distinguishable subsequences (symptoms) crucial for accurate and explainable diagnosis. As shown in Figure 1, if the system can spot two crucial points, (1) the cause chain of the disease from a given specific fetal ECG, and (2) obtaining causal natural structures from the fetal ECG database, then the prediction disease can be more convincing and helpful, also straightforward to locate errors in the AI, rather than a label from a black box. In practice, we ex-

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

pect a medical AI system to provide human-readable and sound explanations to support doctors in making the right decisions. It is worthwhile, especially for underdeveloped areas, where such techniques could help the doctors of rural areas with more reliable references from previous cases. Furthermore, our approach is domain agnostic, which would be applied to other domains, such as autonomous driving, the financial field, etc. Therefore, an intuitive idea we want to explore is, can we discover causality, not from the relation between dimensions, but from inside specific time series of one dimension? However, there are two challenges: (1) How to discover causality inside time series? (2) If there is a causal relationship inside the time series, how to leverage it to benefit neural networks?

To deal with these issues, we propose a novel framework called Mining Causal Natural Structures (*MCNS*). We discover representative subsequences called snippets from the time series and utilize snippets to encode the initial time series into a binary sequence for discretizing a continuous time series. Then, we use a Greedy Fast Causal Inference (GFCI) algorithm to seek causal relations between snippets and construct an inside causal graph. It is worth mentioning that, unlike most related work that requires domain knowledge, *MCNS* is *domain-agnostic*, which greatly enhanced the generalization of our approach.

Based on the above explorations, we do not follow the existing causal graph construction approach that requires pruning or constructing by domain experts, which is a non-automated causal discovery. We impose two restrictions on the GFCI algorithm so that it can *automatically* prune causal graphs. After that, we determine the final causal natural structure using the Bayesian Information Criterion (BIC) and calculate causal strength on edges using Propensity Score Matching (PSM) [Caliendo and Kopeinig 2008] and Average Treatment Effect (ATE) [Holland 1986].

For the second challenge, we impregnate Deep Neural Networks (DNNs) with causal graphs generated by *MCNS*. The first usage is inspired by [Jain and Wallace 2019; Serrano and Smith 2019], which confirmed that attention could not correctly communicate the relative importance of inputs. Hence, we employ causal strength to refine attention to be more precise. Secondly, we leverage the *MCNS* to select shapes to classify time series, similar to the shapelets-based classification method but more explainable and accurate. Additionally, we prune the dataset with the portion containing causality, which leaves the most critical part and results in more accuracy and efficiency.

Our evaluation based on the PyTorch framework with the UCR dataset demonstrates that our *MCNS* can successfully inject the extracted causal knowledge into deep neural networks and improve NN's performance extensively, especially accuracy and interpretability.

In summary, our main contributions are as follows:

- We propose a novel framework for mining causal natural structures (*MCNS*) inside time series, which is both domain-agnostic and automatic.
- We investigate training popular neural network models with our causal natural structures obtained from *MCNS*. It boosts neural network models to realize causal knowledge emanating from *MCNS*.
- Experimental results illustrate that our *MCNS* can effectively enhance NN models for better performance in time series of various domains and scales. It can also help improve the interpretability of neural networks and the time series itself.

Approach

Our mining causal natural structures framework has three components: finding critical data in time series, constructing inside causal graph, and calculating causal strength. Figure 2 shows the overall architecture of our approach.

Problem Definition

MCNS is used to find a causal natural structure $\mathbb{S}$ in subsequences $T_{i,m}$ from given time series T as follows:

- A time series T is a sequence of real-valued numbers $t_i : T = t_1, t_2, \ldots, t_n$, (with an optional label l_T for classification tasks), where n is the length of T.
- A subsequence $T_{i,m}$ of a time series T is a continuous subset of the values from T of length m starting from position i. Formally, $T_{i,m} = t_i, t_{i+1}, \ldots, t_{i+m-1}$, where $1 \leq i \leq n - m + 1$.
- A causal natural structure $\mathbb{S}$ inside time series T is a 4-tuple $< S_{sub}, l_T, \psi, C >$, which is composed of subsequences set S_{sub}, optional label l_T, causal relations ψ, and causal strength C.

Finding Critical Data in Time Series

First, we should find critical data representing the entire time series to discover the event in the real world behind it.

To begin with, we need to determine how to set the subsequence length l. Since the subsequence length corresponds to the time span of events occurring, and often similar real-world events have periodicity, it is desirable that l is equal to the length of the intrinsic period of the time series T. For example, concerning fetal ECG data shown in Figure 1, l should be around the duration of a single fetal heartbeat. We adopt the popular Fast Fourier Transform (FFT) [Cooley and Tukey 1965] as a solution. Time series T is converted into the frequency domain, extracting the dominant frequency f. The subsequence length l is guided by $1/f$. In section 5, we will explore the effectiveness of our approach.

Additionally, to determine the same subsequence length of the complete dataset, we calculate the subsequence length l_T for each time series T in the dataset using FFT, and employ the maximum value in l_T as the unified subsequence length.

To extract representative subsequences, we discover k snippets s_T from each time series T using the time series snippets algorithm [Imani et al. 2018], which is domain agnostic to guarantee *MCNS* can be applied to datasets in any domain.

Constructing Inside Causal Graph

In order to construct the causal graph, we should determine the factors, assemble the edges between factors, impose the constraints on edges, and obtain the final causal graph.

Determine the Factors To merge similar subsequences, we cluster the subsequences obtained in the previous step into n classes using k-shape clusters algorithm [Paparrizos and Gravano 2015]. These classes represent events mentioned above, as reflected by this dataset. n classes factors and (optional) l_T labels constitute the factors of the causal graph. Each time series T can be expressed as a binary sequence. If T contains the corresponding factors, the value of this factor is 1, and not vice versa [Liu et al. 2021]. This binary sequence represents the time series by events (e.g., Bradycardia, Arrhythmia, Fetal Distress in Figure 1).

Assemble the Edges Between Factors The following is to establish the edges, which denote the causal relationship between factors. We choose the GFCI algorithm [Ogarrio, Spirtes, and Ramsey 2016] to detect causal relations and infer without causal sufficiency. GFCI permits us to make causal inferences when having confounding variables and output a Partial Ancestral Graph (PAG). It offers us a preliminary inside causality between factors.

Impose the Constraints on Edges Additionally, we put two kinds of constraints on the graph to refine the causal graph. The first is banning edge from label factors to other factors because classification labels are not involved in actual events. Moreover, the second is an effect that does not precede its cause [Black 1956]. What happens earlier in the time series leads to what happens later. For most time series, provided that factor X appears after Y, we should ban the edge from X to Y.

Obtain the Final Causal Graph The PAG obtained in the last part contains four edge types [Zhang 2008], which are $\rightarrow, \leftrightarrow, \circ\!\!\rightarrow, \circ\!\!-\!\!\circ$. Among them, $X \rightarrow Y$ denotes X causes Y, and $X \leftrightarrow Y$ denotes that there is an unobserved confounder of X and Y. So $\rightarrow$ edges are retained and $\leftrightarrow$ edges are removed. For the remaining two cases, $X \circ\!\!\rightarrow Y$ denotes either $X \rightarrow Y$ or $X \leftrightarrow Y$, and $X \circ\!\!-\!\!\circ Y$ denotes either $X \rightarrow Y$, $Y \rightarrow X$ or $X \leftrightarrow Y$. There's no way to get a true probability, so we operate bootstrapping algorithm to determine the final causal graph. Each case is given the same probability for the two uncertainties mentioned above. We employ Bayesian Information Criterion (BIC) [Schwarz 1978] to estimate the quality of each graph G_n is measured by its fitness with time series T.

Calculating Causal Strength

Even after we have gone through the above steps, the resulting inside causal graph is still noisy. We calculate causal strength on edges to further refine the causal graph. High strength is allocated to edges with a high causal effect. Similarly, meager strength is allocated to edges with low causal effects.

We utilize propensity score matching to measure average treatment effect $\phi_{T,Y}$, which denotes the causal strength of $T \rightarrow Y$. In this paper, it represents the effect of changing the 1 in our binary sequence to 0 through the do-calculus on the classification result, that is to say, the effect of subsequence on classification:

$$\phi_{T,Y} = E[Y \mid do(T=1)] - E[Y \mid do(T=0)] \quad (1)$$

$$= \left[\sum_{t_i=1} \Delta_{i,j} - \sum_{t_i=0} \Delta_{i,j} \right] /N \quad (2)$$

where the do-calculus $do(T=1)$ shows intervention on T and altering the value of T to 1. j represents the most similar instance in a different set than i, and $\Delta_{i,j}$ means the difference between the outcome value of instance i and j.

Impregnation DNN with *MCNS*

Recently, some effort has been made to exploit the DNNs, especially recurrent neural networks (RNNs), for different sizes of time series prediction and classification. However, the application of causal graphs to benefit deep neural networks in time series has been limited. We proposed three methods to impregnate DNN with *MCNS* as shown in Figure 3.

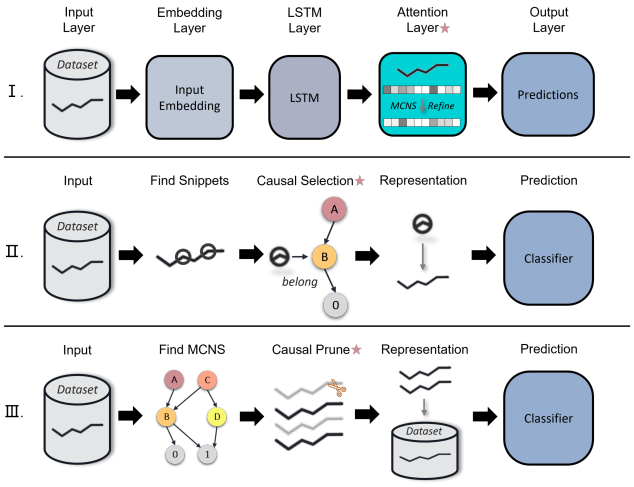

Figure 2: Three usages of impregnating neural networks with causal inference. *Refine Attention with Causal Strength* (Above I), *Shape Causal Selection Based Classification* (Middle II), and *Dataset Prune with Causality*(Below III). The step with an asterisk is the core step of impregnation DNN with *MCNS*.

Refine Attention with Causal Strength Attention has become an effective mechanism for superior results, as demonstrated in time series prediction and classification. However, some prior work substantiates that there is some distance away from attention and the relative importance of inputs [Jain and Wallace 2019; Serrano and Smith 2019]. Attention can not wholly explain the relative importance of inputs.

Causal strength can be exploited to improve it. We utilize a Long Short-Term Memory (LSTM) with attention model [Wang et al. 2016]. The input vector

$X_{t-n} \cdots X_{t-3}, X_{t-2}, X_{t-1}, X_t$ is the n multi-dimensional feature vectors up to the time to be predicted. The hidden layer processes the input vector in the LSTM in some intermediate states. The attention coefficient is obtained from the hidden layer of the last moment of another LSTM network in the decoder. Finally, vector C passes the fully connected layer to calculate the predicted result vector ϱ.

$$e_{ij} = \nu \tanh\left(W \cdot h_j + U \cdot h'_{i-1} + b\right) \tag{3}$$

$$a_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=t-n}^{t} \exp\left(e_{ik}\right)} \tag{4}$$

$$C = \sum_{j=t-n}^{t} a_{ij} h_j \tag{5}$$

where e_{ij} is the relation score between h'_{i-1} and h_j. a_{ij} is the attention coefficient corresponding to e_{ij}. After that, the obtained attention coefficient is assigned to different middle layer states h_j and summed to obtain vector C input to the decoder.

We refine attention using additional loss function H_{cau}. The extra loss function guides attention to the causal strength from *MCNS*. Specifically, $\sum_{q=1}^{Q} \mathrm{BIC}\left(G_q, \mathbf{X}\right) \times \phi_{T,Y}$ is the causal strength corresponding to each factors, and ζ_i is the normalized strength over the whole time series. For the initial LSTM model, $H(p,q)$ means cross-entropy loss on ϱ. Hence, H_{cau} and $H(p,q)$ can be denoted as follows:

$$H(p,q) = -\sum_{i=1}^{n} p\left(x_i\right) \log\left(q\left(x_i\right)\right) \tag{6}$$

$$H_{cau} = \sum_{i=1}^{n} |a_{ij} - \zeta_i| \tag{7}$$

To sum up, we set the updated loss function of the model as follows:

$$L = \alpha H(p,q) + \beta H_{cau} \tag{8}$$

where $\alpha + \beta = 1$. What is worth mentioning is that each factor represents a subsequence. The strength of all the time steps in this subsequence is treated as the causal strength of the factors.

Shape Causal Selection Based Classification Causal inference explores how changes in variable X affect another variable Y. When we set variable X as the shapes inside the time series and variable Y as the classification label, we can recognize that shapes affect the classification results. Hence, our causal natural structure from *MCNS* depicts the classification process of time series.

In other words, *MCNS* may contain crucial information for time series classification. Hence, we operate the factors and causal relations in *MCNS* to guide the classification process in the neural network. Inspired by [Hills et al. 2014], we leverage causal relations and snippets [Imani et al. 2018] to classify time series.

For input time series T, we discover k snippets s_{iT}, and concat snippets as a representation r_T of a time series like shapelets:

$$r_T = \mathrm{concat}\left(s_{1T}, s_{2T}, \ldots, s_{kT}\right) \tag{9}$$

Furthermore, we draw on the causal graph to select snippets. If any snippets of time series belong to causal graph factors that affect the label, we choose them to represent the real content related to classification. If not, we utilize the initial r_T:

$$r_T = \begin{cases} \mathrm{concat}\left(s_{jT}\right) & \text{if } s_{jT} \in S_{sub}, j \geq 1 \\ r_T & \text{otherwise} \end{cases} \tag{10}$$

We mask the parts less than the maximum length and use them as input to the LSTM or other classifiers like k nearest neighbor as the experimental setting. Hence, the neural network or traditional classifier can comprehend the nature of the input.

Dataset Pruning with Causality Every time series in the dataset only sometimes help NN to learn their features. Some data may be redundant or harmful [Angelova, Abu-Mostafam, and Perona 2005]. However, causality can reveal the time series that matters for classification. We employ causality to prune the dataset.

To prune a time series dataset $\varsigma = \{T_1, T_2, \ldots, T_m\}$, we first discover *MCNS* $S_\varsigma =< S_{sub}, l_T, \psi, C >$ on the whole dataset and $S_{T_i} =< S_i, l_{T_i}, \psi_i, C_i >$ on each time series T_i.

Each time series T_i in the dataset is treated as follows:

$$T_i = \begin{cases} T_i & \text{if } < a_{ij}, l_T > \in \psi, a_{ij} \in S_i, i \leq m \\ None & \text{otherwise} \end{cases} \tag{11}$$

The equation (11) means that some time series are abandoned because their causal factors do not affect the classification label. The dataset is already pruned by the operations above.

Afterward, we got the essence data and input it to the LSTM or other neural networks. Therefore, the neural network can comprehend the entire dataset precisely through the essence data.

Experiment

Datasets

To illustrate that *MCNS* can be applied to datasets of different scales and multiple domains, we explore several experiments on six benchmark datasets introduced in the UCR time-series [Dau et al. 2019], which come from the electric power, biology, behavior, food spectrograph, and automotive subsystem domains.

Experimental Setup

Our Models. In this paper, we evaluate our *MCNS* framework as described in Section 3.2-3.4 and three models impregnate NN with *MCNS* (*LSTM+Att+MCNS*, *MCNS*, and *CausalPrune*) as described in Section 3.5.

Parameter Settings. We employ LSTM as the main body, and 2 hidden layers, 128 neurons in each layer, and one fully connected layer connected to the output function are uniformly set. Moreover, we find 5 snippets for each time series and the length determined by the FFT-based method.

Comparison Models. Because no previous work has found causality in univariate time series, we compare

Models	Ratio	PC	EFD	FA	FB	Sb	SKA
LSTM	1%	61.67%/55.06%	–	49.16%/43.77%	50.32%/42.89%	57.30%/67.08%	32.80%/16.46%
	5%	73.67%/62.84%	–	52.27%/46.18%	52.47%/37.93%	62.97%/68.65%	33.06%/16.56%
	10%	85.56%/85.41%	49.36%/33.04%	55.83%/55.04%	57.03%/59.72%	57.03%/51.96%	49.60%/40.83%
	30%	82.78%/82.58%	51.45%/37.30%	51.59%/34.03%	62.96%/60.64%	67.03%/78.89%	38.40%/28.22%
	50%	91.11%/91.09%	60.16%/52.68%	59.69%/66.16%	59.38%/58.93%	68.38%/76.17%	53.06%/51.50%
	80%	83.33%/83.08%	80.60%/80.16%	64.24%/66.71%	61.39%/61.72%	64.32%/78.29%	53.60%/51.84%
	100%	92.77%/92.77%	82.81%/82.75%	60.23%/64.21%	65.80%/58.41%	72.97%/77.97%	57.33%/50.62%
	Prune	**97.44**%/**97.44**%	**88.39**%/**88.22**%	**86.06**%/**86.04**%	**81.36**%/**82.26**%	**87.56**%/**90.25**%	**74.93**%/**75.04**%
LSTM+Att	1%	50.00%/33.33%	–	49.77%/35.96%	53.82%/51.04%	61.89%/74.03%	33.06%/16.59%
	5%	50.00%/33.33%	–	51.59%/34.03%	56.67%/52.21%	64.32%/39.14%	50.40%/40.24%
	10%	88.33%/88.32%	49.71%/33.20%	68.26%/67.75%	57.65%/61.59%	70.54%/70.13%	60.53%/60.27%
	30%	87.22%/87.16%	57.72%/57.40%	**68.71**%/**68.68**%	56.79%/61.87%	65.94%/65.93%	62.93%/58.79%
	50%	92.22%/92.21%	74.09%/72.99%	67.12%/72.36%	68.20%/63.09%	75.67%/75.32%	66.13%/65.34%
	80%	93.89%/93.88%	78.04%/77.94%	71.21%/74.34%	69.75%/61.45%	76.75%/74.92%	40.80%/29.29%
	100%	91.11%/91.11%	84.32%/84.08%	71.74%/75.51%	69.50%/65.69%	76.48%/76.27%	39.73%/33.55%
LSTM+Att +MCNS	1%	68.33%/66.42%	–	49.84%/48.01%	**54.32**%/58.52%	**64.32**%/39.14%	43.20%/34.86%
	5%	**90.00**%/**89.99**%	–	54.84%/52.49%	**59.75**%/54.17%	**64.86**%/41.44%	**60.00**%/48.00%
	10%	**91.67**%/**91.66**%	**63.47**%/**60.30**%	**81.97**%/**81.48**%	**58.47**%/63.24%	**77.56**%/76.85%	**64.00**%/**63.41**%
	30%	**91.67**%/**91.66**%	**68.29**%/**64.95**%	65.34%/61.28%	**66.41**%/**69.91**%	**76.48**%/75.59%	**65.06**%/**62.62**%
	50%	**93.89**%/**93.88**%	**75.49**%/**75.45**%	**69.84**%/**68.72**%	**68.89**%/**70.91**%	**80.81**%/79.26%	**67.73**%/65.76%
	80%	**94.44**%/**94.44**%	**83.51**%/**83.48**%	**74.62**%/**74.94**%	**69.74**%/**69.87**%	**82.43**%/80.55%	63.20%/56.26%
	100%	**96.67**%/**96.66**%	**89.54**%/**89.34**%	**75.00**%/**76.16**%	**72.67**%/**73.19**%	**87.02**%/**85.76**%	57.33%/47.56%
MCNS	1%	**73.13**%/**69.32**%	–	**55.30**%/**52.34**%	52.22%/61.03%	60.00%/**70.98**%	**43.78**%/**38.62**%
	5%	75.56%/76.59%	–	**57.65**%/**60.38**%	52.29%/42.61%	62.60%/65.35%	53.85%/**49.43**%
	10%	74.56%/71.74%	50.29%/66.92%	57.72%/58.17%	56.17%/55.79%	67.30%/74.20%	58.67%/55.49%
	30%	75.56%/69.86%	53.42%/65.93%	57.12%/61.34%	55.67%/52.19%	74.59%/**79.20**%	62.67%/61.50%
	50%	73.88%/68.45%	60.28%/63.76%	58.03%/61.47%	57.63%/53.67%	77.29%/**81.25**%	66.67%/**66.14**%
	80%	76.67%/73.42%	64.23%/69.86%	60.23%/62.34%	59.50%/52.26%	79.72%/**83.59**%	**70.40**%/**66.93**%
	100%	78.89%/76.25%	64.69%/64.57%	60.76%/64.28%	58.89%/56.36%	81.89%/85.65%	**69.86**%/**69.38**%
Shapelets	1%	39.68%/45.56%	–	48.41%/00.00%	49.50%/**66.23**%	35.67%/00.00%	32.78%/11.47%
	5%	43.89%/56.28%	–	48.41%/00.00%	49.50%/**66.23**%	64.32%/**78.29**%	36.80%/31.65%
	10%	49.44%/48.59%	48.89%/55.73%	48.41%/00.00%	49.50%/**66.23**%	64.32%/**78.29**%	33.87%/17.78%
	30%	52.22%/51.51%	59.95%/60.48%	51.59%/68.07%	49.50%/66.23%	64.32%/78.29%	38.40%/27.84%
	50%	51.11%/54.34%	49.71%/00.00%	51.59%/68.07%	49.50%/66.23%	64.32%/78.29%	34.13%/18.36%
	80%	50.00%/66.67%	49.71%/00.00%	51.59%/68.07%	49.50%/66.23%	64.32%/78.29%	44.00%/34.98%
	100%	63.33%/59.76%	49.71%/00.00%	51.59%/68.07%	49.50%/66.23%	64.32%/78.29%	42.67%/42.67%

Table 1: Performance on time series classification task. The first number is Acc, and the second number is F1. The highest results under each ratio are marked with bold. The *Prune* line shows the performance of LSTM using the CausalPrune train set. The – result indicates that the number of samples under this ratio is too small to reach the number of categories in the classification.

three *MCNS*-based models with NN baselines and shape-based methods, including *LSTM*, *LSTM+Att*, and *Shapelets* [Abelson, Sussman, and Sussman 1985]. *LSTM+Att* is a standard model of processing time series. Since the prior

knowledge may result in unfair comparison, we do not add expert knowledge to keep our *MCNS* without domain experts involvement.

Other Settings. The experiments are conducted on Windows 10, coming with an Intel Xeon Silver 4210R CPU and a NVIDIA Tesla T4 GPU.

Main Results

In this section, we investigate the classification accuracy and f1-score of our applications. Each set of experiments was repeated five times for *MCNS*, which is randomized. The main experimental results are shown in Table 2.

Attention vs. No Attention. We can see that *LSTM+Att* outperforms *LSTM* by around 3-4% on average Acc and F1. However, sometimes the addition of attention will make the model less effective. That is because attention can only sometimes enhance the features that affect the results. This above suggests that attention is helpful for neural network models to capture part of crucial information in the time series, but sometimes something else is needed.

Attention vs. Attention + *MCNS*. Furthermore, we can find out that *LSTM+Att+MCNS* transcends *LSTM+Att* by around 6-8% on average Acc and F1. The performance gap is related to the size of the dataset. The above illustrates that causal strength is helpful for attention-based models to discover core content in the time series. What is not explained in the table is that we find that *LSTM+Att+MCNS* converges much faster than *LSTM+Att*, which may be because attention has received the correct guidance.

Causal Inference vs. Neural Networks. Comparing *MCNS* with NN baselines *LSTM* and *LSTM+Att*, we observe in few-shot settings $(1\%, 5\%)$, *MCNS* outperforms NNs by about 6-7% on average Acc and 12-18% on average F1 since NNs tend to underfit in few-shot settings. However, with the increase in training data, the performance gap becomes narrower, and consequently, NNs outperform *MCNS* in several cases. Compared with *MCNS*, NNs have the advantage of learning from large amounts of data.

***MCNS* vs. *Shapelets*.** Similarly, *MCNS* and *Shapelets* are both discriminative subsequences for time series classification. Comparing *MCNS* with baselines *Shapelets* in the case where the other settings are the same except for the shape selection, we observe that *MCNS* outperforms *Shapelets* by about 14% on average Acc and 17.31% on average F1 since our *MCNS* is better than *Shapelets* at capturing subsequences' affection of classification results.

***CausalPrune* vs. No Prune.** The pruned size of the train set is shown in Table 1. We observe under different scales dataset settings by comparing *CausalPrune* with non-*CausalPrune*. Datasets are cropped in different proportions, which is related to the size of the dataset. Furthermore, after the prune, the Acc and F1 on the *LSTM* have increased about 13-15% on average Acc and F1, which illustrates that our *CausalPrune* method can discard harmful and redundant data.

***MCNS* as Presentations.** Additionally, our *MCNS* can represent time series datasets or specific time series. As shown in Figure 4, one significant use of *MCNS* is to replace standard folder icons with *MCNS* graphs that show critical data and relations reflecting the dataset's content. For labeled time series datasets, we can see why different time series are categorized into different classes and essential features. For unlabeled time series datasets, we can see representative subsequences and causality among them, allowing an analyst to spot patterns and anomalies at a glance. Furthermore, by discarding some factors that do not exist in the specific time series, we have similar representations as in Figure 5.

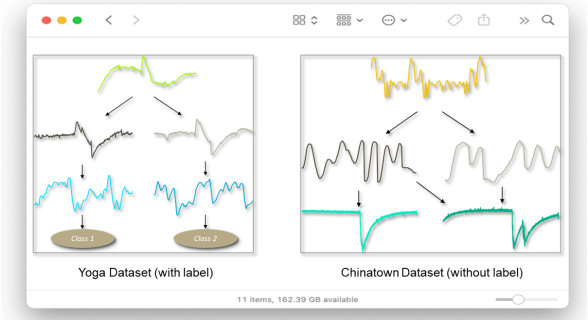

Figure 3: *MCNS* representation of labeled time series datasets (left) and unlabelled time series datasets (right), which allows researchers to discover the features and relationships of datasets at a glance.

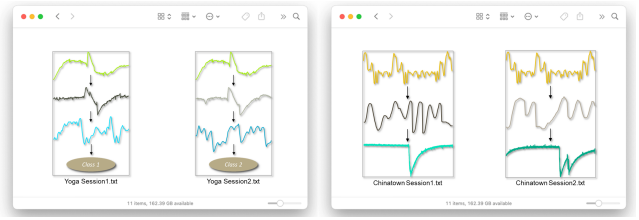

Figure 4: Two examples of specific labeled (left) and unlabeled (right) data represent the simple representation of our causal natural structures from *MCNS* on each time series.

Conclusion

Mining causal natural structures inside time series is a challenging problem. To find out the causal natural structures inside time series data, We propose a novel framework called *MCNS*. It benefits neural networks by refining attention, shape causal selection based classification, and dataset pruning. Extensive experimental results on six real-world datasets from various domains and scales have demonstrated the feasibility and generalization of our approach. The future work will apply *MCNS* to multidimensional time series and integrate *MCNS* into diverse NNs. Furthermore, our *MCNS* can naturally benefit other fields, such as reinforcement learning, adversarial attack, etc.

References

Abelson, H.; Sussman, G. J.; and Sussman, J. 1985. *Structure and Interpretation of Computer Programs*. Cambridge, Massachusetts: MIT Press.

Angelova, A.; Abu-Mostafam, Y.; and Perona, P. 2005. Pruning training sets for learning of object categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, 494–501. IEEE.

Bica, I.; Alaa, A.; and Van Der Schaar, M. 2020. Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. In *International Conference on Machine Learning*, 884–895. PMLR.

Black, M. 1956. Why cannot an effect precede its cause? *Analysis*, 16(3): 49–58.

Caliendo, M.; and Kopeinig, S. 2008. Some practical guidance for the implementation of propensity score matching. *Journal of economic surveys*, 22(1): 31–72.

Cooley, J. W.; and Tukey, J. W. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90): 297–301.

Dau, H. A.; Bagnall, A.; Kamgar, K.; Yeh, C.-C. M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C. A.; and Keogh, E. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6): 1293–1305.

Gerhardus, A.; and Runge, J. 2020. High-recall causal discovery for autocorrelated time series with latent confounders. *Advances in Neural Information Processing Systems*, 33: 12615–12625.

Granger, C. W. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, 424–438.

Hills, J.; Lines, J.; Baranauskas, E.; Mapp, J.; and Bagnall, A. 2014. Classification of time series by shapelet transformation. *Data mining and knowledge discovery*, 28(4): 851–881.

Holland, P. W. 1986. Statistics and causal inference. *Journal of the American statistical Association*, 81(396): 945–960.

Huang, B.; Zhang, K.; Gong, M.; and Glymour, C. 2019. Causal discovery and forecasting in nonstationary environments with state-space models. In *International conference on machine learning*, 2901–2910. PMLR.

Imani, S.; Madrid, F.; Ding, W.; Crouter, S.; and Keogh, E. 2018. Matrix profile xiii: Time series snippets: a new primitive for time series data mining. In *2018 IEEE international conference on big knowledge (ICBK)*, 382–389. IEEE.

Jain, S.; and Wallace, B. C. 2019. Attention is not Explanation. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 3543–3556. Association for Computational Linguistics.

Li, A.; Jiang, S.; Sun, Y.; and Pearl, J. 2022. Learning probabilities of causation from finite population data. *arXiv preprint arXiv:2210.08453*.

Liu, W.; He, Y.; Guan, J.; and Zhou, S. 2024. Multivariate Time Series Forecasting with Causal-Temporal Attention Network. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5735–5739. IEEE.

Liu, X.; Yin, D.; Feng, Y.; Wu, Y.; and Zhao, D. 2021. Everything Has a Cause: Leveraging Causal Inference in Legal Text Analysis. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tür, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 1928–1941. Association for Computational Linguistics.

Mastakouri, A.; and Schölkopf, B. 2020. Causal analysis of Covid-19 spread in Germany. *Advances in Neural Information Processing Systems*, 33: 3153–3163.

Mastakouri, A. A.; Schölkopf, B.; and Janzing, D. 2021. Necessary and sufficient conditions for causal feature selection in time series with latent common causes. In *International Conference on Machine Learning*, 7502–7511. PMLR.

Ogarrio, J. M.; Spirtes, P.; and Ramsey, J. 2016. A hybrid causal search algorithm for latent variable models. In *Conference on probabilistic graphical models*, 368–379. PMLR.

Paparrizos, J.; and Gravano, L. 2015. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 1855–1870.

Schamberg, G.; and Coleman, T. P. 2019. Measuring sample path causal influences with relative entropy. *IEEE Transactions on Information Theory*, 66(5): 2777–2798.

Schwarz, G. 1978. Estimating the dimension of a model. *The annals of statistics*, 461–464.

Serrano, S.; and Smith, N. A. 2019. Is Attention Interpretable? In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2931–2951. Association for Computational Linguistics.

Wang, Y.; Huang, M.; Zhu, X.; and Zhao, L. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, 606–615.

Zhang, J. 2008. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9: 1437–1474.