

# Rethinking Temporal Dependencies in Multiple Time Series: A Use Case in Financial Data

Patrick Asante Owusu\*, Etienne Tajeuna\*, Jean-Marc Patenaude†, Armelle Brun‡, and Shengrui Wang\*

\*Department of Computer Science, Université de Sherbrooke, Sherbrooke, Québec, Canada

†Laplace Insights, Sherbrooke, Québec, Canada

‡LORIA, Université de Lorraine, Nancy, France

Email: {owup2301, taje2102, shengrui.wang}@usherbrooke.ca, jeanmarc@laplaceinsights.com, armelle.brun@loria.fr

**Abstract**—These days, complex systems yield copious time series data, necessitating understanding co-generation, often assessed through pairwise comparisons. However, this method lacks scalability and temporal dynamics handling. In this paper, we advocate using a temporal graph to capture contiguous effects among multiple time series efficiently. Our two-step approach identifies patterns and temporal influences with low execution time, showcasing its potential in financial system incident prediction.

**Index Terms**—Temporal graph, Multiple time series, Cross-sectional patterns, Series trajectory, Financial data

## I. INTRODUCTION

Complex systems generate huge amounts of temporal data. Examples include climate systems and financial systems that generate or record data of multiple, often similar or correlated, variables sequentially to form datasets of multiple time series.

From a data perspective, this represents co-evolving data based on their time dependency. For example, data consisting of different weather stations’ recordings of the rainfall volume are multiple (co-evolving) time series. Identifying dependencies in time series is crucial in areas such as financial markets [1], climate science [2], etc.

Dependency in time series data refers to the relationship(s) between variables or factors in time. To identify these dependencies a pairwise comparison such as the Granger causality test can be performed to measure a series of past values relationships and determine their subsequent directionality. It can also merely be done by looking at how series are correlated [3] in order to capture dependency strength.

To motivate this research, we present a case study involving a two-time series obtained from the Forex market, illustrated in Fig. 1. When examining the initial figure (the up), it is obvious that the series displays a clear downward trend that sharply declines for EURUSD right before the vertical black boundary. Conversely, CADEUR maintains a consistent horizontal fluctuation until the boundary. Starting from the boundary, CADEUR is on a declining trend while EURUSD remains relatively stable with some minor fluctuations. The question we ask is whether one time series has an effect on another beyond this vertical boundary. The four plots below with further explanations in the next paragraph, illustrate how the two-time series can influence each other. This showcases the complexities and interconnections that can arise in time series analysis. More on this in the subsequent sections.

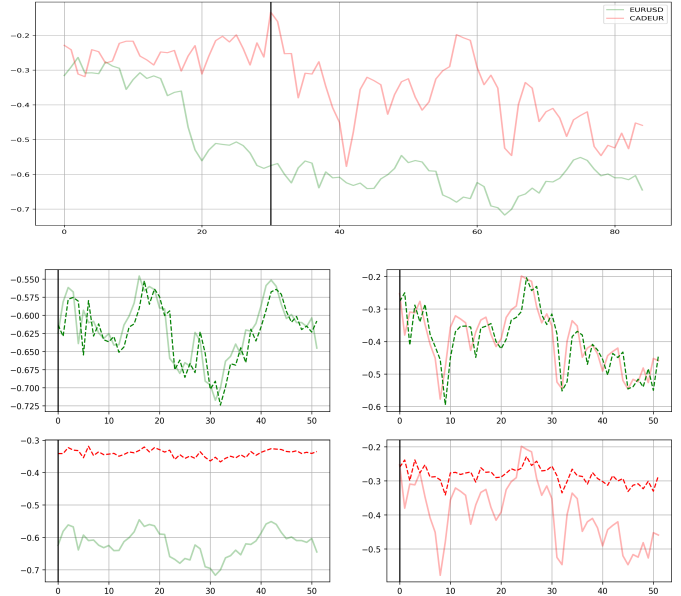


Fig. 1: Example of two-time series from the Cross-rate Forex market. The  $y$  - axes displays scaled values, and the  $x$  - axes shows time points on the graph. We analyze the impact of one series on another (four plots down), focusing on the time after the black line in the first plot up. Solid lines indicate the ground-truth values, and dashed lines represent the generated values.

Despite some advances in identifying dependencies, progress in analyzing multiple time series data is rare. Existing methods mainly concentrate on autocorrelation, failing to capture the propagation of dependencies over time. Moreover, these approaches assume stationarity, limiting their ability to detect long-duration dependencies. In this paper, we address these challenges by learning directional dependencies in the co-evolution behaviour of multiple time series data. We also simplify our model by assuming that these time series can be regenerated using a finite number of linear models. We use a sequence of bipartite graphs to mimic dependencies we have at contiguous time intervals.

Our major contributions are as follows:

- We suggest an approach that detects latent dependencies in multiple time series data by approximating them using finite linear models based on identified patterns.
- We illustrate temporal dependencies using a bipartite

causal graph to model changes across multiple time series by linking repetitive and new dependencies at varying time durations in sub-series.

- To validate our approach, we experiment with real financial market datasets. We show the perks of the proposed method by predicting the subsequent dependencies occurring at time frames ahead.

This paper includes the following sections: related work, background and definitions, methodology, experiments, and conclusion.

## II. RELATED WORK

This study builds on recent progress in comprehending temporal data dependencies and causality in finance. Traditional correlation analysis [4] underpins dependency detection, with financial research exploring the dynamic nature of market price correlations, especially during crises [5]. Furthermore, the GARCH framework [6] is used to gauge interdependence in financial market movements, highlighting potential drawbacks in relying solely on conditional correlation coefficients. In [7], copula functions were employed to model temporal dependencies in daily stock market returns, uncovering heightened interconnections among European markets in response to shared directional shifts, leading to market crashes or booms.

Previous research on identifying dependencies through representation, particularly in the context of causality, has been explored in works by Eichler [8]. The study demonstrates how to transform temporal data into graphs using Granger causality, leveraging time order to establish causal relationships among variables, albeit under certain assumptions, including non-instantaneous dependencies. Tian et al. [9] introduced a method for identifying causal dependencies by detecting structural changes based on local, spontaneous alterations in the underlying data-generating model.

In this context, an important issue is how dependency between multiple time series data in finance can be identified, represented and predicted over time.

## III. PRELIMINARIES

We rely on several concepts and definitions for learning the parameters and identifying representative patterns/behaviours to represent temporal dependencies and predict causal dependencies. In what follows, we provide descriptions of important concepts used throughout this work.

**Multiple time series:** Given a time series  $S^i$ , we call multiple-time series a set of time series synchronously generated. The notation  $\mathcal{MS} = \{S^i \mid i = 1, 2, \dots, n\}$  is employed to denote a set of  $n$  series.

**Time interval:** A time interval refers to a set of limited time points. For a series of time points  $(t_1, t_2, \dots, t_{100})$ , we use the notation  $T_1$  to mean a time interval of a duration  $|T_1| = 100$ . In the same way, any time interval  $T_j$  of duration  $d$ ,  $d \in \mathbb{N}^*$  (i.e.,  $|T_j| = d$ ) will corresponds to the series  $(t_j, t_{j+1}, \dots, t_{j+d-1})$ .

**Sub-time-series data:** A sub-time-series data is a section of temporal data observed within a time interval. The notation

$S_j^i$  denotes the portion of the time series  $S^i$  observed within the time interval  $T_j$ .

In identifying dependencies among multiple time series data, accounting for dynamic and temporal aspects is essential, as traditional lagged approaches may prove ineffective [10]. Our approach emphasizes the significance of varying intervals in establishing time-dependent dependencies, as elaborated below.

**Definition III.1 (Pattern).** We call a pattern, the values of the set of parameters  $\Theta^i = \{e_0^i, e_1^i, \dots, e_p^i\}$ ,  $p \in \mathbb{N}^*$  of an autoregression model enabling the regeneration of a sub-time series  $S^i$ . That is, for each time point  $t \in T_j$  the observed data  $y(t) \in S_j^i$  can be approximated as,

$$y(t) \approx \underbrace{[e_0^i, e_1^i, e_2^i, \dots, e_p^i]}_{\Theta^i} \cdot [1, y(t-1), \dots, y(t-p)]^T \quad (1)$$

For the purpose of simplification, in the rest of the paper, we will use  $f(T_j | \Theta^i, S^i)$  to denote the above autoregression function enabling the regeneration of the sub-time-series  $S_j^i$  via the pattern  $\Theta^i$ .

From the Definition III.1, at each time interval  $T_j$ , for each sub-time-series  $S_j^i$ , there could be a corresponding pattern  $\Theta^i$  enabling its regeneration. This means that for  $m$  time intervals, there can be a total of  $m \times n$  distinct patterns. However, with distinct patterns, no dependency between the multiple time series can be captured. We assume that if there are dependencies in the evolving process of multiple time series, then there should be a much smaller number of common patterns  $K \ll m \times n$  allowing approximately regenerate the sub-time-series.

**Definition III.2 (Co-evolving Time Series).** A multiple time series data set  $\mathcal{MS} = \{S^i \mid i = 1, \dots, n\}$  is said to co-evolve across time intervals  $T_1, \dots, T_m$ , if there exists a small set of patterns  $\Theta = \{\Theta^\kappa \mid \kappa = 1, \dots, K\}$ , with  $K \ll m \times n$ , such that,

$$\begin{cases} \bullet \forall S^i \in \mathcal{MS}, S^i \approx \odot_{j=1}^m (f(T_j | \Theta^\kappa, S^i)), \\ \bullet \text{where } \Theta^\kappa = \underset{\Theta^\kappa \in \Theta}{\operatorname{argmin}} \|S_j^i - f(T_j | \Theta^\kappa, S^i)\|, \forall T_j. \end{cases} \quad (2)$$

with  $\odot$  the concatenation operator.

The identification of the optimal number patterns is in-depth given in Section V-B. As suggested in Definition III.2, if we have multiple time series that co-evolve, then there should be similar patterns (if not equivalent patterns) enabling the regeneration of these time series at different time intervals. This, therefore, yields the following definition.

**Definition III.3 (Pattern Similarity).** Two patterns  $\Theta^1$  and  $\Theta^2$  extracted from a pair of co-evolving multiple time series  $\mathcal{MS}$  are said to be similar ( $\Theta^1 \cong \Theta^2$ ) if there exist a sub-time-series  $S_j^i$  that could be well generated by all of these patterns.

$$\exists T_j, S^i \in \mathcal{MS} / f(T_j | \Theta^1, S^i) \approx S_j^i \wedge f(T_j | \Theta^2, S^i) \approx S_j^i \quad (3)$$

## IV. PROBLEM STATEMENT

Given a set of multiple time series  $\mathcal{MS}$ ,

(1) We search for a (small) number of behaviours  $B = \{B^\kappa, \kappa = 1, \dots, K\}$  in terms of regenerative autoregression

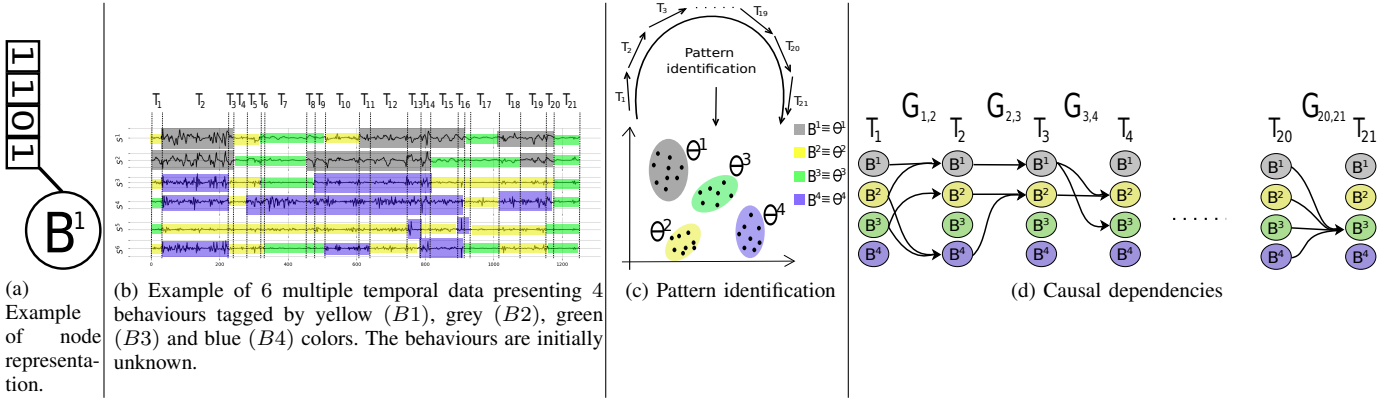


Fig. 2: Overview of the proposed framework.

from multiple time series. In other words, we aim to learn  $K$  autoregressive models  $f(\cdot|\Theta^\kappa)$ ,  $\kappa = 1, \dots, K$ , representing patterns from which the multiple time series values can be regenerated.

(2) Based on the identified behaviours  $B$ , we want to learn the transition mechanism that captures the temporal dependency among the set of series that time-evolve together.

## V. METHODOLOGY

### A. Overview of the proposed approach

Graphs are commonly used to uncover co-dependencies in temporal data with nodes representing series and edges denoting relationships. However, for a large number of co-evolving series, this approach can lead to either dense or sparse graphs. A denser graph limits its utility in understanding inter-series effects, and we aim to avoid pairwise comparisons.

To streamline comparisons and handle multiple time series efficiently, we create a pattern graph (see Section III), where each node denotes a potential behaviour exhibited by a series. Each node is associated with an  $n$ -dimensional attribute variable, taking values in  $\{0, 1\}^n$ . Specifically, the attribute's value at position  $i$  is 1 (0) if series  $S^i$  exhibits (does not exhibit) the represented behaviour. For instance, suppose we have four series:  $S^1, S^2, S^4$  displaying behaviour  $B^1$ , while  $S^3$  does not. Instead of using four separate nodes, we employ a single node to represent behaviour  $B^1$  and its corresponding attribute to indicate which sequences exhibit the behaviour. See Fig. 2(a) for a visual representation.

To get an overview of the proposed approach, consider Fig. 2(b) – (d). In Fig. 2(b), we have a temporal multiple series with 6 examples. Our approach aims to identify 4 main behaviours ( $B^1 - B^4$ ) represented by grey, yellow, green, and blue colours over time. To achieve this, we start by setting a time interval  $T_1$  and extracting sub-co-evolving sequences ( $S_1^1 - S_1^6$ ). We then estimate optimal parameters for linear models to regenerate these sub-series, using a clustering strategy. Initially, we obtain parameters for three linear models associated with yellow, grey, and green behaviours.

Next, we move forward using a fixed stride, checking if the existing models can still regenerate the sub-series. If they can't, we identify new linear models for regeneration. In our

example, this process continues until we reach time interval  $T_2$ , where we discover a new behaviour tagged with the colour blue. This process repeats until the latest observable time point of the co-evolving time series, as shown in Fig. 2(c).

Based on the identified models, we build a bipartite graph relating the effects series may have on each other at each contiguous time interval. Fig. 2(d) depicts the series of bipartite graphs. From  $T_1$  we can note that nodes tagged with behaviours  $B^2$  and  $B^3$  are connected to nodes tagged with behaviours  $B^1, B^2$  and  $B^4$  at time  $T_2$ . For example, the relation  $B^1 \rightarrow B^1$  is observed because series  $S^2$  presents the same behaviour during the two-time intervals. In the same way, the relation  $B^2 \rightarrow B^1$  is observed because series  $S^1$  switches its behaviour from  $B^2$  at  $T_1$  to  $B^1$  at  $T_2$ . It is good to recall that, each node has attributes.

In brief, we suggest a two-step method to detect and model evolving effects in multiple time series. First, we identify patterns to recreate series variation over various time intervals. Second, using these patterns, we reconstruct the ensemble of series variation across different time intervals to analyze their mutual effects. We use a temporal bipartite graph to depict these incidents chronologically. Now, let's delve deeper into each step of our approach.

### B. Step-1: Pattern identification

Pattern identification is a core step of our approach. In Algorithm 1 we have steps illustrating how we identify the patterns from a given multiple time series.

Given  $\mathcal{MS} = \{S^i | i = 1, 2, \dots, n\}$ , at a given first time interval  $T_1$  of duration  $d$ , through a linear autoregressive model  $f(\cdot)$ , we initially identify the set  $\hat{\Theta} = \{\hat{\Theta}^i | i = 1, \dots, n\}$  of parameters (patterns) such that  $f(T_1|\hat{\Theta}^i, S^i) \approx S_1^i$ ,  $i = 1, \dots, n$ . The set  $\hat{\Theta}$  is later clustered into an ensemble of homogeneous subsets each represented by one pattern  $\Theta^\kappa \in \hat{\Theta}$ . It is good noting that, patterns all falling within the same cluster enable the regeneration of the same time series within a given time interval.

Moving across the series, we repeat the process at each time interval  $T_j$  ahead if and only if there exists some sub-time series for which none of the discovered patterns can regenerate their values at that time interval. At the end

---

**ALGORITHM 1: Pattern identification**


---

**Data:**  $\mathcal{MS} = \{S^i \mid i = 1, 2, \dots, n\}, f(), T_1, \tau$  /\* temporal data, an autoregression model, first-time interval of duration  $d$  and moving step. \*/

**Result:**  $\Theta = \{\Theta^\kappa, \kappa = 1, \dots, K\}$ ; /\* Set of patterns. \*/

**begin**

$\hat{\Theta} \leftarrow \emptyset, \Theta \leftarrow \emptyset, \overline{\mathcal{MS}} \leftarrow \{S^i \mid i = 1, 2, \dots, n\}$ ;

**1- Get new patterns:**

**begin**

**for**  $S_1^\mu \in \overline{\mathcal{MS}}$  **do**

$\hat{\Theta}^\mu \leftarrow \underset{\Theta^\mu}{\operatorname{argmin}} \|f(T_1 | \Theta^\mu, S^\mu) - S_1^\mu\|$  /\* fitting. \*/

$\hat{\Theta} \leftarrow \hat{\Theta} \cup \{\hat{\Theta}^\mu\}$ ;

$\Theta \leftarrow \Theta \cup$  a partition of the set  $\hat{\Theta}$  /\* The partition could be done using any clustering algorithm. Each subset is represented by a single pattern. \*/

$\overline{\mathcal{MS}} \leftarrow \overline{\mathcal{MS}} \setminus \hat{\Theta}$ ;

**2- Update set of patterns:**

**begin**

**while**  $\tau$  never reached series length **do**

$T_j \leftarrow$  move previous time interval of a step  $\tau$  and get next time interval;

**for**  $i = 1, \dots, n$  **do**

**if**  $\forall \Theta^\kappa \in \Theta, \|f(T_j | \Theta^\kappa, S^i) - S_j^i\| >$

$\|f(T_1 | \Theta^\kappa, S^i) - S_j^i\|$  **then**

$\overline{\mathcal{MS}} \leftarrow \overline{\mathcal{MS}} \cup \{S_j^i\}$ ;

**if**  $\overline{\mathcal{MS}} \neq \emptyset$  **then**

Run step 1-;

---

of the series coverage, we obtain a final set of patterns  $\Theta = \{\Theta^\kappa \mid \kappa = 1, \dots, K\}$  from which we can regenerate the overall set of co-evolving time series.

### C. Step-2: Causal dependencies

Based on the set of patterns, we build a temporal graph  $\mathcal{SG} = \{\mathbf{G}_{1,2}, \mathbf{G}_{2,3}, \dots\}$  where each  $\mathbf{G}_{j,j+1} = (B, E_{j,j+1}, A_j, A_{j,j+1})$  is bipartite, attributed and directed graph with

$$E_{j,j+1} = \left\{ (B^\mu, B^\nu), B^\mu, B^\nu \in B \mid \exists S^i \in \mathcal{MS}, \right. \\ \left. f(T_j | \Theta^\mu, S^i) \approx S_j^i \wedge f(T_{j+1} | \Theta^\nu, S^i) \approx S_{j+1}^i \right\}, \quad (4)$$

the set of oriented edges relating the behaviour transition a series may adopt in consecutive time intervals and

$$\begin{cases} A_j = \{v_j^\mu \in \{0, 1\}^n \mid \forall \nu, (B^\mu, B^\nu) \in E_{j,j+1}\}, \\ A_{j+1} = \{v_{j+1}^\nu \in \{0, 1\}^n \mid \forall \mu, (B^\mu, B^\nu) \in E_{j,j+1}\}, \end{cases} \quad (5)$$

are the sets of outgoing and incoming node attributes.

In the rest of the paper, when going to mathematical calculation, sets  $A_j$  (resp.  $A_{j+1}$ ) and  $E_{j,j+1}$  should be viewed as the attribute matrix in  $\{0, 1\}^{K \times n}$  and adjacency matrix in  $\{0, 1\}^{K \times K}$  respectively.

## VI. PREDICTING TEMPORAL DEPENDENCIES

Knowing how to identify dependencies from temporal data, we now want to capture the hidden logic explaining the behavioural changes that time series undergo from one time interval to another.

### A. Learning process

We leverage the temporal bipartite graph to learn link dynamics, using a graph neural network for encoding and a reconstructor neural network for decoding. This constitutes a graph autoencoder, explained further in the following sections.

1) *Encoder:* To learn the causal dependencies, we exploit a graph-based neural network for encoding the previous behaviour transitions into a latent space. Due to the fact that snapshots  $\mathbf{G}_{j,(j+1)} \in \mathcal{SG}$  of our temporal graph are distinctly built, for an order  $p \in \mathbb{N}^+$ , we distinctly and respectively encode  $p+1$  consecutive graphs  $\mathbf{G}_{(j-p-1),(j-p)}, \dots, \mathbf{G}_{(j-1),j}$  into latent data  $Y_{(j-p-1),(j-p)} \in \mathbb{R}^{2K \times q}, \dots, Y_{(j-1),j} \in \mathbb{R}^{2K \times q}$ ,  $q \ll n$ . Each latent data  $Y_{(j-p-1+l),(j-p+l)}$  is here calculated using a  $l^{\text{th}}$  ( $0 \leq l \leq p$ ) topology adaptive graph convolutional neural network ( $TAGCN_l()$ ) as,

$$Y_{(j-p-1+l),(j-p+l)} = TAGCN_l(\mathbf{G}_{(j-p-1+l),(j-p+l)}) \cdot \quad (6)$$

Above the fact that each snapshot of the temporal graph is generated using a TAGCN, through a perceptron, we aggregate the distinct  $p$  first encoded graph as

$$\hat{Y}_{(j-1),j} = MLP\left(\odot_{l=0}^{p-1} Y_{(j-p-1+l),(j-p+l)}\right) \cdot \quad (7)$$

Using equations (6) and (7) we thus define our encoder as,

$$\begin{aligned} \operatorname{Enc}(\mathbf{G}_{(j-p-1+l),(j-p+l)}, 0 \leq l \leq p \mid \mathbf{W}^{\operatorname{enc}}) = \\ \begin{cases} \bullet Y_{(j-p-1+l),(j-p+l)}, & 0 \leq l \leq p \\ \bullet \hat{Y}_{(j-1),j} \end{cases} \quad (8) \end{aligned}$$

where  $\mathbf{W}^{\operatorname{enc}}$  are the encoder parameters.

2) *Decoder:* In contrast of the encoder that ingest  $p+1$  graphs, the decoder takes as input the  $(p+1)^{\text{th}}$  embeddings and attempts to reconstruct the  $p+1$  corresponding graph, that is the adjacency matrix and the attribute matrices. Formally, we define our decoder as,

$$\begin{aligned} \mathbf{G}_{(j-1),j} = \operatorname{Dec}(Y_{(j-1),j} \mid \mathbf{W}^{\operatorname{attr}}, \mathbf{W}^{\operatorname{edge}}) \quad (9) \\ = \begin{cases} A_j \odot A_{j+1} = \operatorname{Softmax}\left(MLP[Y_{(j-1),j} \mid \mathbf{W}^{\operatorname{attr}}]\right), \\ E_{j,(j+1)} = \operatorname{Softmax}\left(MLP[Y_{(j-1),j} \cdot Y_{(j-1),j}^* \mid \mathbf{W}^{\operatorname{edge}}]\right), \end{cases} \end{aligned}$$

where  $W_{\operatorname{attr}}^r$  and  $W_{\operatorname{edge}}^r$  are the decoder neural network parameters.

3) *Learning:* Given  $\mathbf{W}^{\operatorname{enc}}$  the parameters of the encoder, we train it in such way that the calculated latent data  $\hat{Y}_{(j-1),j}$  values via Eq.(7) should be close to the value naturally generated by  $TAGCN_{p+1}()$  the  $(p+1)^{\text{th}}$  TAGCN network as follows,

$$\mathcal{L}^{\operatorname{enc}} = \min_{\mathbf{W}^{\operatorname{enc}}} \sum_{j=2}^m \left[ \|\hat{Y}_{(j-1),j} - Y_{(j-1),j}\| \mid \mathbf{W}^{\operatorname{enc}} \right]. \quad (10)$$

For the decoder, we calculate the loss in reconstructing the attribute matrices as well as the adjacency matrix as,

$$\mathcal{L}_{\operatorname{dec}}^{\operatorname{attr}} = \min_{\mathbf{W}^{\operatorname{attr}}} \sum_{j=2}^m \left[ \|\hat{A}_j \odot \hat{A}_{j+1} - A_j \odot A_{j+1}\| \mid \mathbf{W}^{\operatorname{attr}} \right] \quad (11)$$

$$\mathcal{L}_{\operatorname{dec}}^{\operatorname{edge}} = \min_{\mathbf{W}^{\operatorname{edge}}} \sum_{j=2}^m \left[ \|\hat{E}_{j,(j+1)} - E_{j,(j+1)}\| \mid \mathbf{W}^{\operatorname{edge}} \right] \quad (12)$$

Using equations (10), (11) and (12) we then calculate the whole loss of our autoencoder as,

$$\mathcal{L} = \mathcal{L}^{\operatorname{enc}} + \mathcal{L}_{\operatorname{dec}}^{\operatorname{attr}} + \mathcal{L}_{\operatorname{dec}}^{\operatorname{edge}}. \quad (13)$$

## B. Prediction

Given the temporal dependencies  $\mathcal{SG} = \{\mathbf{G}_{1,2}, \dots, \mathbf{G}_{(m-1),m}\}$  covering time intervals ranging from  $T_1$  to  $T_m$ . We want to predict the plausible dependencies we may observe at the next contiguous time intervals  $T_m$  and  $T_{m+1}$  (i.e., the graph  $\mathbf{G}_{m,(m+1)}$ ). For this, we will exploit our trained graph autoencoder. Here, we particularly make use of the MLP component of the encoder and the whole decoder component.

Recall that for a settled lag  $p$ , the encoder will ingest  $p+1$  consecutive snapshots of the temporal graph and reconstruct the  $(p+1)^{th}$  graph. Knowing the latest  $p+1$  snapshots we generate the corresponding embeddings using Eq. (8). To predict next embeddings at the contiguous time intervals  $T_m$  and  $T_{m+1}$ , we use the relation Eq. (7) where we pass the novel list of  $p$  embeddings given as,  $Y_{(m-p),(m-p+1)}, \dots, \hat{Y}_{(m-1),m}$ . From the aggregated embeddings, we thus use the decoder given in Eq. (9) to reconstruct the node attributes as well as the adjacency matrix. Formally, to predict the next dependencies  $\mathbf{G}_{m,(m+1)}$  we only need to estimate the attribute  $\hat{A}_{m+1}$  and the adjacency  $\hat{E}_{m,(m+1)}$  matrices as follows,

$$\hat{\mathbf{G}}_{m,(m+1)} = \begin{cases} \bullet \hat{Y}_{m,(m+1)} = MLP(\odot\{ \\ Y_{(m-p),(m-p+1)} \dots, \hat{Y}_{(m-1),m} \}) \\ \bullet Dec(\hat{Y}_{m,(m+1)} | \mathbf{W}^{attr}, \mathbf{W}^{edge}) \end{cases} \quad (14)$$

## VII. EXPERIMENTS

In this section, we evaluate the proposed approach:

- 1) For identifying causal dependencies in multiple time series using a linear regression model. We also present how patterns are repeatedly exhibited over time across the input multiple data.
- 2) For learning the underlying causal dependencies' mechanism explaining the co-evolutionary aspect of the input multiple data present.
- 3) For predicting causal dependencies in subsequent remaining series at times ahead.

We utilize the FINCH clustering algorithm [11] to determine initial clusters in a scalable manner, eliminating the need for predefined cluster numbers. For addressing time series data representation challenges, we turn to Topology Adaptive Graph Convolutional Networks (TAGCNs) [12], renowned for their interpretable node representations, scalability, and performance in time series analysis, particularly in forecasting. We leverage TAGCN to perform convolutional operations, enabling us to aggregate local neighbourhood information effectively.

### A. Dataset description

Our model was tested with a diverse range of real-world financial market data, including stocks, currencies, investment funds, and major indices from Yahoo Finance. This data offered unique insights into market behaviours. For stocks, we focused on 74 out of 78 large-cap stocks in the Global Industry Classification Standard - Information Technology sector index as of March 2022, spanning from 2017-01-01 to 2022-01-01. Additionally, we utilized Exchange-Traded Funds (ETFs) data, representing 15 series of the most liquid and

TABLE I: Summary of Data Used

Financial Instruments	Data	#Series ( $n$ )	#Data points ( $m$ )
Large-cap Market	S&P 500	74	1,259
Investment Fund	ETFs	15	3,893
Stock Exchange	World Indices	35	1,318
Currency	Forex Cross-rates	25	4,175

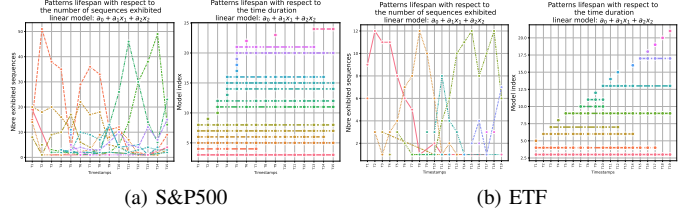


Fig. 3: Example of patterns identified using a linear autoregression model with lag 2. Figures illustrate how the identified patterns persist over time.

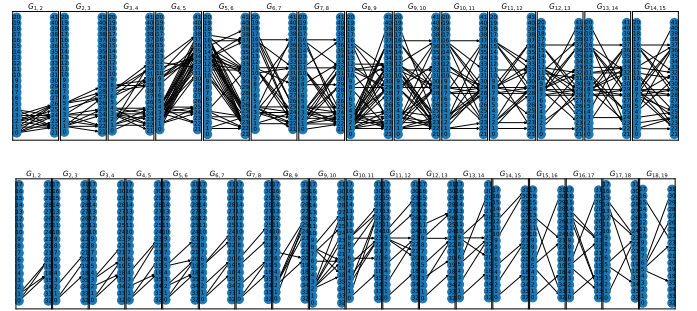


Fig. 4: Temporal dependencies identified in each dataset. From top to down are S&P500, ETF, Indices and Forex respectively. Initial settings (i.e.,  $|T_1|$  and stride  $\tau$ ) are identical as in Fig. 3.

traded ETFs, suitable for global macro investment strategies. This dataset encompassed US and global equities, emerging markets, US Treasuries, bonds, commodities, real assets, and the US dollar index. For major indices, we sourced data from Yahoo Finance, covering a series range from 2009-01-01 to 2014-03-28, including 35 out of 36 series available as of June 2023. Lastly, we analyzed forex cross-currency rates, incorporating all major currencies and others, utilizing 26 out of 29 cross-currency series as of June 2023 from Yahoo Finance, between 2006-01-01 and 2021-12-31. We excluded 3 cross-currency series, 4 large-cap stocks, and 1 world index due to null values or data unavailability.

Table I summarizes the datasets of multiple series of returns. Each set exhibits observable temporal dependencies, as shown in Fig. 1.

### B. Causal dependencies

To begin, let's examine the potential patterns that can arise from using linear autoregressive models of the form  $y(t) = a_0 + a_1y(t-1) + a_2y(t-2)$  (i.e., lag 2). Fig. 3 depicts the persistence of identified patterns in each of the datasets. Each data set comprises two components; the left side is the total number of series generated by the identified model and, the right side is the lifespan (duration) where these models are

TABLE II: Compared Execution time of models.

Data	Granger Causality ( <i>sec</i> )	PCMCI ( <i>sec</i> )	Proposed ( <i>sec</i> )
S&P 500	505	3273	<b>21</b>
ETFs	19	137	<b>9</b>
World Indices	228	1288	<b>20</b>
Forex Cross-rates	57	280	<b>14</b>

TABLE III: Performance of the approach in predicting causal dependencies in our dataset

	Proposed approach, train performances			
	S&P500	ETFs	Indices	Forex
Accuracy	91.4%	97.1%	80.6%	88.5%
F-1	83.4%	92.3%	61.9%	72.4%
Precision	90.7%	96.6%	86.17%	89.1%
Recall	89.2%	95.4%	63.3%	78.4%

used to generate the ground-truth series. For these plottings, we make use of first-time interval  $T_1$  values whose duration are  $|T_1| = 300, 300, 50, 200$  for S&P500, ETF, Indices and Cross-rate Forex respectively. As stride values, we used  $\tau = 50, 20, 20, 200$ .

As shown in Fig. 3, some patterns persist throughout the entire time intervals, indicating inter-series influences through generative models. This observation aligns with the findings in Fig. 4, detailed in Section V-C, where the causal graph represents patterns as nodes. Notably, each sub-graph maintains a constant number of nodes, predetermined by the total pattern count, yet the scalability of nodes and edges depends on the initial  $T_1$  size. These co-evolving dependencies underscore the existence of inter-series dependencies, with a significant impact from the initial settings at  $T_1$  and the chosen stride value. Table II shows the execution time taken to identify dependencies in our approach, the Granger causality test and PCMCI model [10]. We maintained the same data settings on all models.

### C. Predicting causal dependencies

In this section, we present how could the temporal graph be exploitable for predicting subsequent dependencies in multiple time series having a co-evolving aspect. For this, we first define the following settings,

(1) For each data, we take all the historical temporal graphs except the two latest time intervals as knowledge-based information. The two latest time intervals are assumed to be unknown information.

(2) From the knowledge-based information, we then train our graph autoencoder.

(3) Having the neural network trained. We then predict the subsequent dependencies at the subsequent time intervals.

**Training phase:** We employ a consistent neural network architecture across all our datasets, consisting of three graph neural networks (GNNs) with two hidden layers of TAGCN type. These hidden layers are configured with 18 and 25 neurons, respectively. We incorporate a custom binary activation function, which transforms all values below a predefined threshold of 0.5 to 1 and sets the rest to 0 within the encoder.

**Prediction phase:** Predicting subsequent dependencies can also be likened to link prediction. We compare our approach

TABLE IV: Accuracy of models.

Predicting one time interval ahead				
Data	Proposed	RGNN	EvGCN	TGCN
S&P500	81.1%	65.9%	65.4%	<b>93.1%</b>
ETF	<b>90.8%</b>	68.4%	69.6%	84.5%
Indices	<b>95.8%</b>	66.6%	84.1%	63.8%
Forex	<b>86.7%</b>	78.6%	79.5%	74.4%

to three recent deep-based models for temporal graphs: RGNN [13], DCGR [14], and TGCN [15], using the same architecture settings as our proposed approach. Table IV displays accuracy results, with bold indicating the best values for each dataset.

## VIII. CONCLUSION

In summary, our novel two-tier dependency model surpasses conventional techniques like correlation and autoregression, enhancing interpretability by uncovering dependencies within sub-time series through parameter-based pattern exploration. It consistently outperforms traditional and recent models in detecting temporal dependencies, as demonstrated in Table II. Future works will expand the dataset to include more system data.

## ACKNOWLEDGMENT

We would like to thank NSERC and Laplace Insights for their financial support and input throughout the project.

## REFERENCES

- [1] H. Cheng and al, "Inferring causal interactions in financial markets using conditional granger causality based on quantile regression," *Computational Economics*, pp. 1–30, 2021.
- [2] C. Monteleoni and al, "Climate informatics," tech. rep., CRC Press, 2013.
- [3] E. G. Tajeuna and al, "Modeling regime shifts in multiple time series," *ACM TKDD*, vol. 17, no. 8, pp. 1–31, 2023.
- [4] N. J. Gogtay and U. M. Thatte, "Principles of correlation analysis," *Journal of the Association of Physicians of India*, vol. 65, no. 3, pp. 78–81, 2017.
- [5] M. F. Öztek and al, "Financial crises and the nature of correlation between commodity and stock markets," *International Review of Economics & Finance*, vol. 48, pp. 56–68, 2017.
- [6] K. J. Forbes and al, "No contagion, only interdependence: measuring stock market comovements," *The journal of Finance*, vol. 57, no. 5, pp. 2223–2261, 2002.
- [7] E. Jondeau and al, "The copula-garch model of conditional dependencies: An international stock market application," *Journal of international money and finance*, vol. 25, no. 5, pp. 827–853, 2006.
- [8] M. Eichler and al, "Causal reasoning in graphical time series models," *arXiv preprint arXiv:1206.5246*, 2012.
- [9] J. Tian and al, "Causal discovery from changes," *arXiv preprint arXiv:1301.2312*, 2013.
- [10] J. Runge and al, "Detecting and quantifying causal associations in large nonlinear time series datasets," *Science advances*, vol. 5, no. 11, p. eaau4996, 2019.
- [11] S. Sarfraz and al, "Efficient parameter-free clustering using first neighbor relations," in *Proceedings of the IEEE/CVF*, pp. 8934–8943, 2019.
- [12] Y. Chen and al, "Multi-scale temporal features extraction based graph convolutional network with attention for multivariate time series prediction," *Expert Systems with Applications*, vol. 200, p. 117011, 2022.
- [13] Y. Seo and al, "Structured sequence modeling with graph convolutional recurrent networks," in *ICONIP*, pp. 362–373, Springer, 2018.
- [14] Y. Li and al, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.
- [15] L. Zhao and al, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE TITS*, vol. 21, no. 9, pp. 3848–3858, 2019.