Neural Computing and Applications (2023) 35:18519–18531
https://doi.org/10.1007/s00521-023-08674-y

ORIGINAL ARTICLE

Predicting the state of synchronization of financial time series using cross recurrence plots

Mostafa Shabani[1] · Martin Magris[1] · George Tzagkarakis[2,3] · Juho Kanniainen[4] · Alexandros Iosifidis[1]

Received: 25 November 2022 / Accepted: 10 May 2023 / Published online: 8 June 2023

Abstract

Cross-correlation analysis is a powerful tool for understanding the mutual dynamics of time series. This study introduces a new method for predicting the future state of synchronization of the dynamics of two financial time series. To this end, we use the cross recurrence plot analysis as a nonlinear method for quantifying the multidimensional coupling in the time domain of two time series and for determining their state of synchronization. We adopt a deep learning framework for methodologically addressing the prediction of the synchronization state based on features extracted from dynamically subsampled cross recurrence plots. We provide extensive experiments on several stocks, major constituents of the S &P100 index, to empirically validate our approach. We find that the task of predicting the state of synchronization of two time series is in general rather difficult, but for certain pairs of stocks attainable with very satisfactory performance (84% F1-score, on average).

Keywords Cross recurrence plot · Synchronization · Kernel convolutional neural network · Financial time series

1 Introduction

Time series prediction and classification in finance is significantly challenging due to the complexity, multivariate nature, and non-stationary nature of time series in this domain [1]. Security trading and price dynamics in financial markets are particularly complex due to the interacting nature and inter-connectedness of their underlying driving forces and determinants leading to significant co-movements in stocks' prices. The characterization and modeling of multivariate time series dynamics have long been discussed in the financial literature, where the prevailing approach is that based on classical econometric theory. Among the multivariate linear models, the most widespread ones are vector autoregressive (VAR) models [2, 3], vector moving averages, ARMA (autoregressive moving average) models [4], and cointegrated VAR models [5]. Widespread is the use of multivariate conditional heteroskedasticity GARCH-type, see, e.g., [6] for a review, multivariate stochastic volatility models [7], and more methods based on the realized volatility [8].

Among the nonlinear models, the threshold autoregressive model [9], smooth transition autoregressive models [10], and Markov switching models [11] are nowadays standard approaches. Alternatives include nonparametric methods, functional coefficient [12] and nonlinear additive AR models [13], recurrence analysis, and neural networks. The complexity of modern financial markets running over the so-called limit-order book mechanism is, however, characterized by typical nonlinear, noisy, and often non-

✉ Martin Magris
magris@ece.au.dk

Mostafa Shabani
mshabani@ece.au.dk

George Tzagkarakis
gtzag@ics.forth.gr

Juho Kanniainen
juho.kanniainen@tuni.fi

Alexandros Iosifidis
ai@ece.au.dk

[1] Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark

[2] Institute of Computer Science, Foundation for Research and Technology - Hellas, Iraklio, Greece

[3] IRGO (EA 4190), University of Bordeaux, Bordeaux, France

[4] Department of Computing Sciences, Tampere University, Tampere, Finland

stationary dynamics. In addition, the high-dimensional nature of the limit-order book flow and the complexity of the interactions within it constitute severe limits in the applicability of classic econometric methods for its modeling and forecasting. Besides a very limited number of analytical and tractable models for the order flow and price dynamics in limit-order books [14–16], machine learning methods have received much attention [17, 18], as they are naturally appealing in this context.

By considering the stock market as a complex system, it is natural to apply such methods for addressing those prediction problems where the application setting and assumptions beneath standard econometric techniques are stringent or inadequate. Indeed, it has extensively been shown that, in financial applications, deep learning (DL) models are often capable of outperforming traditional approaches due to their ability to learn complex data representations based on end-to-end data-driven training, see, e.g., [19–24]. DL models have been adopted for a variety of problems ranging from price prediction [25–29], limit-order book-based mid-price prediction [20, 21, 23, 30, 31], and volatility prediction [32–34].

Whereas the target of the above literature is generally the analysis and prediction of single time series, this paper focuses on an analysis of stock pair co-movements. Several trading strategies can be put into play to take advantage of co-movements and exploit statistical arbitrages, including pair-trading, portfolio management, or relative and convergence trading strategies applied at an intraday level, e.g., see [35] for an overview. While DL provides a basis for prediction given a set of descriptive features, the issue of how to detect and quantify co-movements remains to be addressed. This paper suggests the use of recurrence analysis based on cross recurrence plots (CRP) for detecting and extracting features indicative of stocks' shared dynamics or co-movements, along with a deep learning framework for predicting whether certain pairs of stocks will exhibit a shared dynamics in the future (in the sense specified in Sect. 2). Not only in the view of extending the ML and applied econometrics literature in this direction, but the possibility of forecasting epochs of time series synchronization is likewise relevant for practitioners.

For detecting and quantifying co-movements or, more generally, shared dynamical features in time series, the standard econometric approach is that of cross-correlation analysis [11, Ch. 8]. This intuitive linear approach, based on the estimation and perhaps forecasting of cross-correlation matrices, appears to be an element of a much wider theory and methodological approach that has been explored and developed in the last years within a broader generic non-financial setting. Simple cross-correlation analysis has been remarkably extended and generalized toward methods that help explore co-movements between time series within nonlinear, noisy, and non-stationary systems of very complex dynamics, either financial [36–38] or not [39–41].

Recurrent analysis [42] explores the reconstruction of a phase-space using time-delay embedding for quantifying characteristics of nonlinear patterns in a time series over time [43]. This is done by calculating the so-called Recurrence Plot [44], the core concept of which is to identify all points in time that the phase-space trajectory of a single time series visits roughly the same area in the phase-space. Recurrence plot analysis has no assumptions or limitations on dimensionality, distribution, stationarity, and size of data [42]. These characteristics make it suitable for multidimensional and non-stationary financial time series data analysis. The CRP [45] is an extension of the recurrence plot, introduced to analyze the co-movement and synchronization of two different time series. The CRP indicates points in time that a time series visits the state of another time series, with possibly different lengths in the same phase-space. These concepts are discussed in further detail in Sect. 2.

In this paper, we propose a method for predicting the state of synchronization over time of two multidimensional[1] financial time series based on their CRP. In particular, we use the CRP to quantify the co-movements and extract the binary representation of its diagonal elements as the prediction targets for a DL model. For predicting the state of synchronization at the next epoch, we employ a Convolutional Neural Network (CNN) that uses as inputs CRPs independently calculated from data-crops obtained by applying fixed-size sliding windows on the time series. Our extensive experiments on 12 stocks of the S &P100 index selected from different sectors show that the proposed method can predict the synchronization of stock pairs with satisfactory performance.

The remainder of the paper is organized as follows. Section 2 introduces in detail the concepts and theory behind the CRP, with an outlook on its applications in financial and economic problems. Our proposed approach for predicting time instances of time series' synchronization is presented in Sect. 3. Empirical results on real market data are provided in Sect. 4, while Sect. 5 provides conclusions.

[1] Throughout the paper, with uni- or multi- *variate* we refer to the nature of the analyses (RP as opposed to CRP), and with one- or multi- *dimensional* we refer to the nature of the time series. That is, the RP (as presented in equation (2)) provides a univariate analysis of a single one-dimensional time series, while the CRP (as presented in latter equation (3)) a multivariate analysis of two one-dimensional or multidimensional time series.

2 Financial time series recurrence analysis

Recurrence in the analysis of time series, seen as a nonlinear dynamic system, is the repetition of a pattern over time. The visualization of recurrences in the dynamics of a time series can be expressed via a RP or recurrence matrix [42]. In other words, the RP represents the recurrence of the phase-space trajectory to a state. The phase-space of a d-dimensional time series $\mathcal{N}$ with T observations $\mathcal{N} = \{\mathbf{n}_1^\top, \mathbf{n}_2^\top, \ldots, \mathbf{n}_T^\top\}^\top$, with $\mathbf{n}_i$ being the row vector representing a generic observation at time i, $i = 1, \ldots, T$ is calculated using the time-delay embedding method. State N_i in the phase-space is obtained by

$$N_i = [\mathbf{n}_i, \mathbf{n}_{i+\tau}, \ldots, \mathbf{n}_{i+(k-1)\tau}], \ i = 1, \ldots, T', \tag{1}$$

where τ denotes the delay and k is the embedding dimension, $T' = T - \tau(k-1)$, τ and k can, respectively, be determined with the Average Mutual Information Function method [46] and the False Nearest Neighbors method of [47]. For a uni-dimensional time series, N_i is a row vector of size $(1 \times k)$, and for a d-dimensional times series N_i is a row vector of size $(1 \times kd)$. The recurrence state matrix of the reconstructed phase-space, known as Recurrence Plot (RP), at times i and j, is defined as

$$R_{i,j}(\varepsilon) = H(\varepsilon - \|N_i - N_j\|), \ i,j = 1, \ldots, T', \tag{2}$$

where ε is a threshold distance value, $H(\cdot)$ is the Heaviside function, and $\|\cdot\|$ is the euclidean distance. Due to the underlying embedding (1), $R_{i,j}$ is defined for i $i = 1$ up to $T' = T - \tau(k-1)$. If two states N_i and N_j are in an ε-neighborhood the value of $R_{i,j}$ is equal to 1, otherwise is 0. The value of ε highly affects the output of RP. When ε is too small or too large, the RP cannot identify the true recurrence of states. There are different approaches for finding the best value for ε in the literature [42]. We follow the guidelines provided in [48] for selecting ε. The values on the diagonal line of RP are equal to one (i.e., $R_{i,i} = 1$) because in that case, the two states introduced to $H(\cdot)$ are identical. The diagonal line of RP is called the Line Of Identity (LOI). Recurrence quantification measures derived from RPs, such as recurrence rate (RR), percent determinism, and maximum line length in the diagonal direction [42], give insights into the dynamic behavior of time series. These measures have been used in financial research to analyze the behavior of financial data, e.g., [49–51]. The RP of a time series can be used as a data transformation method for time series prediction. A method employing the RP of seven financial time series to train a deep neural network for predicting the market movement is proposed by [52]. Several authors have suggested an RP forecasting approach via DL. A feature extraction method exploiting the RP for parsing a DL algorithm is proposed by [53]. RP can be treated as images enabling the use of computer vision techniques for the forecasting task, such as autoencoders [54] or CNNs [55].

The CRP of two multidimensional time series [56] corresponds to an extension of RP which explores the co-movement of two time series and allows the study of the nonlinear dependencies between them. Let us denote by N_i, $i = 1, \ldots, T$ and M_j, $j = 1, \ldots, S$ the phase-space states of the time series $\mathcal{N}$ and $\mathcal{M}$ of length T and S, respectively. The cross recurrence (CR) states of the reconstructed state-space a time i and j are

$$CR_{i,j}(\varepsilon) = H(\varepsilon - \|N_i - M_j\|) \tag{3}$$

with $i = 1, \ldots, T' = T - \tau(k-1)$ and $j = 1, \ldots, S' = S - \tau(k-1)$. Here N_i and M_i are defined as in (1). $\mathrm{CR}_{i,j}$ defines the concept of synchronization and the way synchronization between two financial time series measured: an ε-neighborhood of the embeddings N_i, M_j at epochs i, j. We denote the full cross recurrence matrix, known as cross recurrence plot (CRP) extracted for $\mathcal{N}$ and $\mathcal{M}$ as $\mathrm{CRP}_{(\mathcal{N},\mathcal{M})}$, obtained through

$$\mathrm{CRP}_{(\mathcal{N},\mathcal{M})} := \{\mathrm{CR}_{i,j}(\varepsilon)\}_{i=1,\ldots,T',j=1,\ldots,S'}. \tag{4}$$

The CRP corresponds to a matrix of dimension $T' \times S'$, which may not be square, as the time series $\mathcal{N}$ and $\mathcal{M}$ may have different lengths, i.e., $T \neq S$.

For $\mathcal{N}$, $\mathcal{M}$ of equal length T, the $CRP_{(\mathcal{N},\mathcal{M})}$ is a square $T' \times T'$ matrix. Opposed to the (univariate) recurrence analysis of one time series (with itself, RP in equation (2)), the diagonal entries of the CRP are either 1 or 0, as the two time series may or may not be synchronized at (i, j), $i = j, i = 1, \ldots, T'$, see, e.g., Figure 2. In the CRP of two time series, the LOI is replaced by a distorted diagonal, called the Line Of Synchronization (LOS). The LOS reveals the relationship between the two time series in the time domain. In particular, it provides a nonparametric function containing information about the time-rescaling of the two time series, which further allows their re-synchronization [57].

As the time series we consider in our application are multidimensional ($d > 1$), we point out that the CRP is indeed a multidimensional cross recurrence plot (MdCRP) [56] where $\mathbf{n}_i$, $\mathbf{m}_j$ are row vectors rather than scalars, and N_i, M_j are dk-dimensional row vectors rather than k-dimensional row vectors, as opposed to the conventional CRP based on one-dimensional time series. Yet the above discussion is general and applies to both cases, and $CR_{i,j}$ is in any case a scalar equal to either 0 or 1. For multidimensional time series, the entries of each of the two time series require normalization in each dimension before estimating the MdCRP [56], e.g., with the z-score.

Financial time series co-movement analysis using the CRP and LOS is studied in [58, 59]. The work in [60] analyses the inter-dependencies of the stock market index and its associated volatility index, further proposing a method for the LOS estimation based on a corrupted CRP. Furthermore, it is important to notice that financial time series are often represented as multivariate instances. Indeed, even the most basic source of financial data generally provides information about volumes along with prices. Furthermore, a number of closely related variables (e.g., volatility measures) are simple to extract. Despite the use of multidimensional inputs being effective and commonly found across several applications [42], existing CRP applications on market data are broadly limited to the use of one-dimensional series only (e.g., prices or volatilities) [42, 49, 61].

3 Proposed method

We exploit the CRP to quantify the co-movement of two multidimensional time series $\mathcal{N}$ and $\mathcal{M}$ observed continuously over a common period of length V. Our goal is to predict whether at a certain epoch (e.g., a certain calendar day), $\mathcal{N}$ and $\mathcal{M}$ are synchronized in the state-space embedding ε-neighborhood given by (3).

Assume two generic time series $\mathcal{N}$ and $\mathcal{M}$ are observed over the non-overlapping time-domains $D_{\mathcal{N}} = \{t_1, \ldots, t_T\}$ and $D_{\mathcal{M}} = \{s_1, \ldots, s_S\}$ of respective length T and S, their CRP corresponds to a $T' = (T - \tau(k-1)) \times (S - \tau(k-1)) = S'$ matrix where the (i, j) element expresses the state of synchronization at the i-th time instance of the first time series (t_i) and at the j-th time instance of the second (s_j), in terms of the ε-neighborhood of the states N_i and M_j, as expressed by (3). With the domains being non-overlapping, there are no epochs t_i and s_j such that $t_i = s_j$ in (the same) calendar time, and the state of synchronization at a same calendar time cannot be determined. Indeed, for a fixed t_i, the column-vector $\mathrm{CRP}_{i,\cdot}$ reports for the epochs $s_j, j = 1, \ldots, S'$ (past or future with respect to t_i) whether state-space embedding M_j is in the same ε-neighborhood of N_i.

In this light, if the domains of $\mathcal{N}$ and $\mathcal{M}$ only partially overlap over a region $D := D_{\mathcal{N}} \cap D_{\mathcal{M}}$ of length V, for our forecasting purpose, their non-overlapping regions $D_{\mathcal{N}} \backslash D$ and $D_{\mathcal{M}} \backslash D$ are irrelevant and can be discarded. Their V overlapping time instances $t_i, \ldots, t_{i+V}$ and $s_j, \ldots, s_{j+V}$ correspond to the same calendar epochs, i.e., $t_{i+h} = s_{j+h}, \forall h = 1, \ldots, V$, and are of actual relevance. Over the common domain D, the CRP corresponds to a square $V' \times V'$ matrix ($V' = V - \tau(k-1)$) with a well-defined diagonal expressing the state of synchronization at $t_i = s_j$, e.g., answering whether at the (same) calendar day $t_i = s_j$, $\mathcal{N}$ and $\mathcal{M}$ are synchronized or not.

This justifies the required form for the input data, corresponding to two (multidimensional) time series $\mathcal{N}$ and $\mathcal{M}$ observed over a common period D of equal length V, with $D = \{v_1 = \max(t_1, s_1), \ldots, v_V = \min(t_T, s_S)\}$. Since the essence of time series forecasting is that of predicting the future from the past, the data from the past needs to be representative of the h-step ahead forecast. This implicitly requires D to be a continuous set of times for the given sampling frequency. That is, there should be no gaps between days or epochs, namely, $v_V \equiv v_1 + (V - 1)$. Furthermore, in order to calculate (3), we require the time series to be non-corrupted over D in all its multivariate entries, i.e., without missing values. The above requirements are generally met for the financial time series of our interest. The only constraints are that of using data for stocks traded at the same exchange (same trading days and observed festivities) and that of selecting stocks not subject to delisting in the period of interest.

Aligned with the general rationale of time series forecasting, we aim at predicting the one-step-ahead synchronization status between $\mathcal{N}$ and $\mathcal{M}$ at epoch $i + 1$, based on some lagged historical records available up to time i, that is based on some suitable set of feature observed or extracted over w past epochs. For $i = w, \ldots, V' - 1$, let us denote by $\mathcal{N}_i^w$ and $\mathcal{M}_i^w$ the sub-sample of $\mathcal{N}$ and $\mathcal{M}$ of the w most recent observations up to and including epoch i, that is

$$\mathcal{N}_i^w = [\mathbf{n}_{i-w+1}, \mathbf{n}_{i-w+2}, \ldots, \mathbf{n}_i],$$
$$\mathcal{M}_i^w = [\mathbf{m}_{i-w+1}, \mathbf{m}_{i-w+2}, \ldots, \mathbf{m}_i].$$

Let us denote by $\mathrm{CRP}_{(\mathcal{N}_i^w, \mathcal{M}_i^w)}$ the $w' \times w'$ CRP computed from $\mathcal{N}_i^w$ and $\mathcal{M}_i^w$ (with embedding dimension k, lag τ and $w' = w - \tau(k-1)$). At epoch i, $\mathrm{CRP}_{(\mathcal{N}_i^w, \mathcal{M}_i^w)}$ is used as the input of the neural network for predicting the state of synchronization at $i + 1$. Within this framework, there are $V' - w$ input-target pairs. The first pair corresponds to the input $\mathrm{CRP}_{(\mathcal{N}_w^w, \mathcal{M}_w^w)}$ and target $(\mathrm{CRP}_{(\mathcal{N}, \mathcal{M})})_{w+1, w+1}$, the last to the input $\mathrm{CRP}_{(\mathcal{N}_{V'-1}^w, \mathcal{M}_{V'-1}^w)}$ and target $(\mathrm{CRP}_{(\mathcal{N}, \mathcal{M})})_{V', V'}$. The prediction target at epoch i corresponds to the state of synchronization at $i + 1$, provided by the (diagonal) entry $\left(\mathrm{CRP}_{(\mathcal{N}, \mathcal{M})}\right)_{i+1, i+1}$ of the CRP computed for the entire times series $\mathcal{N}$, $\mathcal{M}$. In particular, the state of synchronization at any epoch $i = 1, \ldots, V'$ is provided by the diagonal of $\mathrm{CRP}_{(\mathcal{N}, \mathcal{M})}$, i.e.,

$$\underset{i=w,\ldots,V'-1}{\mathbf{diag}}(\mathrm{CRP}_{(\mathcal{N}, \mathcal{M})})_i = \begin{cases} 1, & \text{if } \mathcal{N} \text{ and } \mathcal{M} \\ & \text{are synchronized at time } i, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

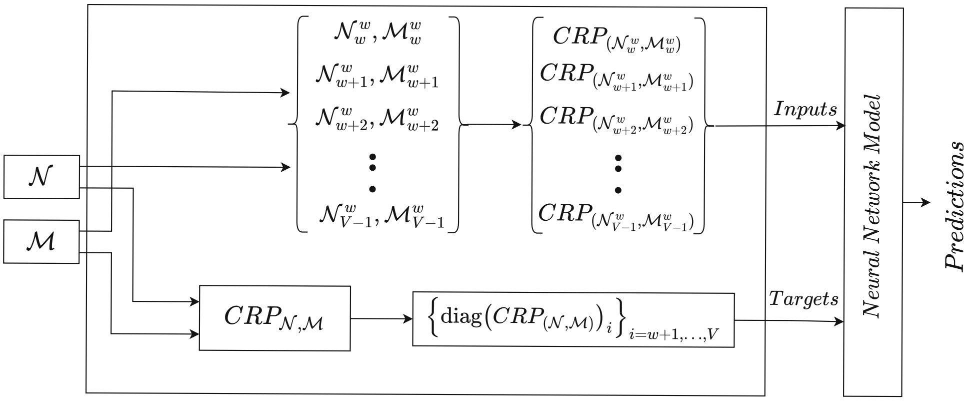
$\mathcal{N}$
$\mathcal{M}$
$\mathcal{N}^w_w, \mathcal{M}^w_w$
$\mathcal{N}^w_{w+1}, \mathcal{M}^w_{w+1}$
$\mathcal{N}^w_{w+2}, \mathcal{M}^w_{w+2}$
$\mathcal{N}^w_{V-1}, \mathcal{M}^w_{V-1}$
$CRP_{(\mathcal{N}^w_w,\mathcal{M}^w_w)}$
$CRP_{(\mathcal{N}^w_{w+1},\mathcal{M}^w_{w+1})}$
$CRP_{(\mathcal{N}^w_{w+2},\mathcal{M}^w_{w+2})}$
$CRP_{(\mathcal{N}^w_{V-1},\mathcal{M}^w_{V-1})}$
$CRP_{\mathcal{N},\mathcal{M}}$
$\left\{\mathrm{diag}\left(CRP_{(\mathcal{N},\mathcal{M})}\right)_i\right\}_{i=w+1,\ldots,V}$
Inputs
Targets
Neural Network Model
Predictions

Fig. 1 Proposed method. For two input time series $\mathcal{M}$, $\mathcal{N}$, we construct their CRP and take its one-period-ahead diagonal entries as targets for model prediction (bottom flow). On the other hand, we construct a sequence of CRPs based on a fixed-size sliding window ending at the current period, whose elements constitute the inputs for the Neural Network (upper flow)

so that $\left\{\mathrm{diag}\left(\mathrm{CRP}_{(\mathcal{N},\mathcal{M})}\right)_i\right\}_{i=w+1,\ldots,V'}$ corresponds to the targets for the inputs $\left\{\mathrm{CRP}_{(\mathcal{N}_i^w,\mathcal{M}_i^w)}\right\}_{i=w,\ldots,V'-1}$. The above corresponds to a framework where inputs are created dynamically by using CRPs computed over sub-sampled time series obtained by applying sliding windows of a fixed size. The above construction is illustrated in Fig. 1. Note that $\mathrm{CRP}_{(\mathcal{N}_i^w,\mathcal{M}_i^w)}$ is not analogous to the sub-matrix P obtained from $\mathrm{CRP}_{(\mathcal{N},\mathcal{M})}$ by considering its rows and columns from $i-w+1$ to i. In $\mathrm{CRP}_{(\mathcal{N},\mathcal{M})}$ the entire data in $\mathcal{N}$ and $\mathcal{M}$ accounts for the time series normalization and furthermore tunes the parameter ε. $\mathrm{CRP}_{(\mathcal{N}_i^w,\mathcal{M}_i^w)}$ is thus truly dependent on the cropped times series data $\mathcal{N}_i^w$, $\mathcal{M}_i^w$, while P is not. In a forecasting context, our approach is feasible and unbiased as it does not use any future information following the one available at i. Note that, in general, nothing prevents from choosing the embedding size and lag parameter differently for the CRP computation of the targets and for the CRP computations of the inputs.

A simple example with two one-dimensional time series clarifies how we extract the input features and prediction targets. Consider the two time series $\mathcal{A}$ and $\mathcal{B}$ of 10 observations:

$$\mathcal{A} = \{A,B,A,A,C,D,D,B,C,C\}^\top,$$
$$\mathcal{B} = \{A,C,C,C,D,B,D,B,C,C\}^\top.$$

Fig. 2 depicts their CRP, i.e., $\mathrm{CRP}_{(\mathcal{A},\mathcal{B})}$ (for simplicity computed with $k=\tau=1$, and $V=V'=10$). The diagonal line of the CRP is highlighted and includes the values of the recurrence states. The diagonal line shows that the behavior of $\mathcal{A}$ and $\mathcal{B}$ at timestamps between 1 and 7 to 10 is synchronized, therefore at these timestamps the prediction targets are set to 1 (the actual value of the Heaviside function in (3)). By, for instance, setting $w=3$, we aim at predicting $V-w=7$ states of synchronization. The first prediction concerns the synchronization at epoch $w+1=4$, based on the $\mathrm{CRP}_{(\mathcal{N}_3^3,\mathcal{M}_3^3)}$, that is, the CRP calculated from the first three observations of $\mathcal{A}$ and $\mathcal{B}$. The prediction of the synchronization at epoch 5 is based on the CRP calculated on observations 2 to 4, i.e., on $\mathrm{CRP}_{(\mathcal{N}_4^3,\mathcal{M}_4^3)}$. The procedure is repeated up to epoch $V-1=9$, where $\mathrm{CRP}_{(\mathcal{A}_9^3,\mathcal{B}_9^3)}$, calculated from the observations 7 to 9, is used for predicting $\mathrm{diag}(\mathrm{CRP}_{(\mathcal{A},\mathcal{B})})_{10}$.

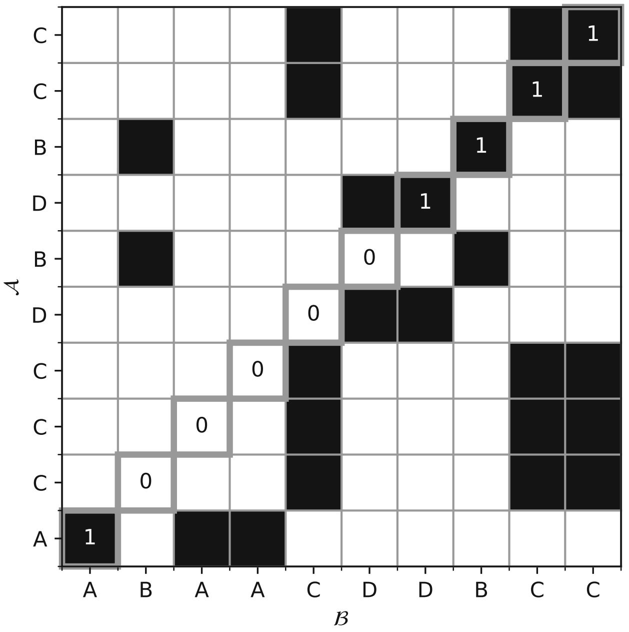

Fig. 2 The CRP of two time series with the diagonal line highlighted

Appendix B includes a visualization that displays the results of recurrence analysis on real-world stock data. This visualization provides empirical evidence of the complexity of the patterns underneath the RP and CRP.

To practically implement the underlying DL model that maps each $\mathrm{CRP}_{(\mathcal{N}_i^w,\mathcal{M}_i^w)}$ input to its corresponding $\mathrm{diag}\left(\mathrm{CRP}_{(\mathcal{N},\mathcal{M})}\right)_{i+1}$ output, consider that each input consists of a matrix of zeros and ones that can be considered analogous to an image. Therefore we can rely on well-established classification models. In particular, we employ a Convolutional Neural Network (CNN). Such a neural network is well-suited for capturing the spatial relationships between the features in their input, which in our case correspond to the 0-1 features encoded in the entries of $\mathrm{CRP}_{(\mathcal{N}_i^w,\mathcal{M}_i^w)}$. Note that in the CRP calculation $\mathcal{N}_i^w$ and $\mathcal{M}_i^w$ are z-score normalized before computing (3).

We use a CNN architecture formed by two convolutional and one fully connected layer, as illustrated in Fig. 3. The neural network involves the typical blocks of the CNN architecture. The convolutional layers adaptively learn the spatial relationships of inputs, the Rectified Linear Unit (ReLU) activation introduces nonlinearity to the model, and the max-pooling layer provides down-sampling operations reducing the size of the feature map by extracting the maximum value in each patch from the input feature map. The current CNN is chosen based on a grid search over different network architectures, layers' types, and sizes, aimed at maximizing the F1-score and showing the feasibility of our CRP-based DL approach. Importantly, the temporal connections in the input data are handled within the CNN. In fact, the construction of the input data as outlined above relates all the instances in CRP_{SW_i} to the target y_i, where the entries in CRP_{SW_i} furthermore constitute aggregate values (CRP entries) capturing the similarity of the input processes at different lags of time, within the windows W. The output of the stacked convolutional layers is introduced to a fully connected layer, leading to the network's output by applying a softmax function.

4 Experiments

4.1 Data

Our analyses rely on daily adjusted closing prices and daily number of traded shares (volumes) for 12 representative constituents of the S &P100 index in the period from December 31st, 2014 to November 29th, 2021 ($V = 1,741$ trading days). The data is retrieved from Yahoo Finance. These 12 stocks are selected based on their market capitalization and their market sector. For each sector, we select the first two stocks of highest-but-comparable capitalization, a practice well-supported by financial theory [62]. Market sectors provide a natural grouping for securities: analyses conducted at a sector level are a common practice for granting comparability and robustness of the results, as across market sectors the dynamics of economic variables are well-known to be asymmetric. Table 1 lists our stock selection. Each stock is expressed as a trivariate time series consisting of daily prices, volumes, and returns. CRPs express temporal similarities in joint terms of the price level, traded volume, and daily return, providing a generalized definition of similarity in time series dynamics at a multivariate level.

For our bivariate analysis on two time series, we have $(12^2 - 12)/2 = 66$ pairs of stocks. For each stock pair, we use the first 70% of the data for training ($V_{\mathrm{train}} = 1,218$ days) and the last 30% for testing ($V_{\mathrm{test}} = 523$ days). As the future input instances should not affect the training process, the order of the input data during the training is fixed. The input and targets of the train data and the test data are, respectively,

$$\text{Inputs: } \left\{\mathrm{CRP}_{(\mathcal{N}_i^w,\mathcal{M}_i^w)}\right\}_{i\in I},$$

$$\text{Targets: } \left\{\mathrm{diag}\left(\mathrm{CRP}_{(\mathcal{N},\mathcal{M})}\right)_i\right\}_{i\in T},$$

where $I = w,\ldots,V_{\mathrm{train}}$ and $T = w+1,\ldots,V_{\mathrm{train}}+1$ for the training set and $I = V_{\mathrm{train}}+1,\ldots,V-1$ and $T = V_{\mathrm{train}}+2,\ldots,V$ for the test set. We train the neural network once over the data for all the picks of the stock pairs. This pooled approach is a common practice in closely related Machine Learning literature, e.g., [21], and supported by the empirical findings of [63], suggesting the existence of an universal price formation mechanism (model), and thus price dynamic, not specific for individual assets. In practice, the input and output data is the concatenation of the individual pairs' inputs-targets. For example, for a set window size w, for the train set the input-target data consists of $(V'_{\mathrm{train}} - w) \times 66$ examples, that is $(V'_{\mathrm{train}} - w) \times 66$ pairs of cross recurrent matrices and (scalar) targets, where $V'_{\mathrm{train}} = (V_{\mathrm{train}} - \tau(k-1))$. In the training phase, the training data is used to estimate the optimal weights of the CNN. The test data is then parsed to the estimated CNN and the quality of the network outputs

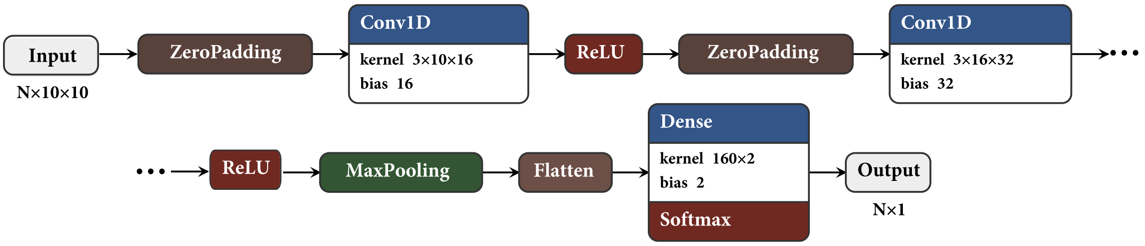

Fig. 3 Architecture of the proposed convolutional neural network

Table 1 List of selected stocks

Sector	Ticker	Stock name
Electronic technology (ET)	INTC	Intel corporation
Electronic technology	QCOM	Qualcomm inc
Energy minerals (EM)	XOM	Exxon mobil corporation
Energy minerals	CVX	Chevron corporation
Finance (F)	JPM	JP Morgan Chase & co
Finance	V	Visa Inc
Health technology (HT)	JNJ	Johnson & Johnson
Health technology	PFE	Pfizer, inc
Retail trade (RT)	HD	Home depot, inc. (The)
Retail trade	WMT	Walmart inc
Technology services (TS)	MSFT	Microsoft corporation
Technology services	GOOG	Alphabet inc

is evaluated against the actual targets. Details are provided in the following two subsections.

For the training of the CNN we adopt the ADAM optimizer with the following hyperparameters: learning rate 0.01 (reduced by a factor of 5 every 40 epochs), momentum parameters 0.9 and 0.999, batch size 128 and epoch size 300. Across the epochs, we keep track of the F1-score on the validation set, which is set to the last 15% portion of the training set. For our classification task, we adopt the binary cross-entropy loss. As the target classes are unbalanced, the loss is weighted for the targets' class proportion. Details on the filter sizes, kernel sizes and the max pooling size are provided in Fig. 3.

With respect to the CRP computations, throughout our analyses the embedding dimension k is set to 2 or 3 (estimated via FNN method) based on input type, and the delay parameter τ is set to 1. Values 0.45, 0.55, 0.65, and 0.75 are used for the threshold ε. These hyperparameters are selected according to the guidelines and discussion in [48] and [56]. The same values are applied for both the computation for the CRP related to the targets and the CRPs related to the inputs.

In our experiments, we consider two different choices for the window-size hyperparameter, namely $w = \{10, 30, 50, 60, 80\}$ days. With the above settings, $V = V' = 1,741$ days, $V_{\text{train}} = V'_{\text{train}} = 1,218$, and $V_{\text{test}} = V'_{\text{test}} = 523$ days. For $i = w, \ldots, V-1$, $\text{CRP}_{(\mathcal{N}_i^w, \mathcal{M}_i^w)}$ are square matrices of size $w' = w$ and $CRP_{(\mathcal{N}, \mathcal{M})}$ a square matrix of size V on whose diagonal are found the relevant targets, i.e., $\text{diag}\left(CRP_{(\mathcal{N}, \mathcal{M})}\right)_i$, $i = w+1, \ldots, V$.

Table 2 Performance measures on the test set using (adjusted) price and volume as input variables

w	ε	Accuracy	Precision	Recall	F1-score
10	0.45	0.960	0.886	0.818	**0.848**
10	0.55	0.981	0.842	0.762	0.796
10	0.65	0.992	0.836	0.684	0.737
10	0.75	0.997	0.999	0.647	0.727
30	0.45	0.957	0.877	0.804	0.836
30	0.55	0.979	0.821	0.752	0.782
30	0.65	0.991	0.816	0.684	0.732
30	0.75	0.996	0.998	0.539	0.571
50	0.45	0.956	0.861	0.808	0.832
50	0.55	0.982	0.907	0.719	0.784
50	0.65	0.993	0.907	0.668	0.737
50	0.75	0.997	0.935	0.665	0.739
60	0.45	0.954	0.850	0.814	0.831
60	0.55	0.976	0.775	0.730	0.751
60	0.65	0.992	0.937	0.633	0.703
60	0.75	0.997	0.816	0.689	0.737
80	0.45	0.950	0.827	0.809	0.818
80	0.55	0.979	0.839	0.725	0.769
80	0.65	0.990	0.776	0.666	0.707
80	0.75	0.997	0.820	0.709	0.753

4.2 Experimental results

Stock pairs from the same sector or two different sectors with different co-movement behaviors can provide comprehensive experimental data to show the ability of the proposed method to predict the state of synchronization. To evaluate the performance of our proposed method, all pairs of stocks are used as the input of the method. We collect all pairs of stocks, and for each pair we follow the proposed steps (ref. Figure 1) to create the inputs and targets. We stack the input-target pair-specific data to create a single train and test set for all pairs.

Tables 2 and 3 show the performance of our proposed approach for all pairs of stocks using two types of input: (price, volume) and (price, return, volume), respectively. Given that the target classes are generally imbalanced, the preferred reference performance metric is the F1-score. Yet, we also include accuracy, precision, and recall[2] to have a clearer overview of the classification performance. For robustness, we run our experiments over a range of

[2] For a binary classification task involving N samples, let TP, TN, FP, and FN, respectively, denote true positive, true negative, false positive, and false negative: the performance measures are defined as follows: $\text{Accuracy} = (\text{TP+TN})/N$, $\text{Precision} = \text{TP}/(\text{TP+FP})$, $\text{Recall} = \text{TP/(TP+FN)}$, $\text{F1} = 2(\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$.

Table 3 Performance measures on the test set using (adjusted) price, volume and returns as input variables

w	ε	Accuracy	Precision	Recall	F1-score
10	0.45	0.946	0.858	0.802	**0.827**
10	0.55	0.973	0.810	0.733	0.765
10	0.65	0.988	0.776	0.694	0.728
10	0.75	0.995	0.795	0.664	0.711
30	0.45	0.943	0.845	0.800	0.820
30	0.55	0.971	0.789	0.742	0.763
30	0.65	0.987	0.760	0.678	0.711
30	0.75	0.996	0.998	0.601	0.667
50	0.45	0.938	0.826	0.785	0.804
50	0.55	0.971	0.795	0.731	0.758
50	0.65	0.987	0.758	0.667	0.702
50	0.75	0.995	0.998	0.503	0.505
60	0.45	0.938	0.823	0.788	0.804
60	0.55	0.967	0.752	0.735	0.743
60	0.65	0.987	0.756	0.654	0.692
60	0.75	0.996	0.955	0.595	0.657
80	0.45	0.933	0.805	0.788	0.796
80	0.55	0.967	0.755	0.730	0.742
80	0.65	0.989	0.893	0.616	0.677
80	0.75	0.995	0.998	0.548	0.586

values for the window-size *W* and threshold ε hyperparameters, a setup that further clarifies the effect of these hyperparameters on prediction performance. Additional results for all the 66 pairs of stocks are provided in Appendix A.

Results for the (price, volume) time series input are provided in Table 2, results for the (price, return, volume) input in Table 3. In general, our results show that the task of predicting the state of synchronization is not only feasible but, under our setup, quite satisfactory. Indeed our preferred performance F1 metric is as high as 84%. As expected, the results appear to be sensible to the choice of the window size and threshold parameter. In particular, the performance metrics decrease in their values as the threshold parameter and the window size increase. This means that stricter ε-neighborhoods are easier to predict and that the relevant information for the prediction of the synchronization state is found in the most recent instances of the CRP. This suggests the existence of patterns in the data that are strongly indicative of close ε-neighborhoods. I.e., the CNN detects clear patterns indicative of the fact that the day-ahead synchronization is likely to be very strong (the ε-neighborhood is tight), indeed, as ε increases, the performance metric decreases, indicating that the model indeed detects strong evidence of "strong" day-ahead synchronization. Regarding the window size, long-lagged CRP information appears to introduce noise in the system without providing any predictive gains, aligned with the intuition that further-in-time information is less and less related to the current state of the system and of little use for prediction.

Suspecting that the joint use of prices and returns might be redundant since they are closely related to each other, we also run a second experiment involving volumes and returns only. It is interesting to note that the inclusion of the returns does not seem to provide any advantage with respect to the (price, volume) input time series, but rather the opposite effect. It is expected that the inclusion of further input variables complicates the patterns in the CRP chessboard so that under the same network architecture, the performance metrics decrease. Furthermore, and aligned with the above, in additional experiments here not reported, we included squared returns (as a gross measure of daily volatility), finding that they also appear detrimental to the performance metrics and prediction task. This perhaps suggests that the network architecture needs to scale up with the dimensionality of the input data, reasonably inducing more complex patterns in the CRP.

An alternative method for assessing the prediction method's performance is to examine its performance on periods of high and low volatility for stock pairs. To do this, for each stock, we construct estimates of daily volatility by applying the Exponentially Weighted Moving Average (EWMA) filter [64]:

$$\sigma_t = \sqrt{\lambda \cdot r_{t-1}^2 + (1-\lambda) \cdot \sigma_{t-1}^2} \tag{6}$$

where σ is the volatility, r is the return, t is an index denoting the day, and λ a decay factor. When calculating the volatility, the decay factor λ determines the weight given to older returns. We set $\lambda = 0.94$, which is commonly used for daily returns series [64].

For each stock in every pair, we intersect the days of highest 30% and lowest 30% volatility. From such an intersection, we identify dates corresponding to two volatility regimes (high volatility and low volatility). For every pair, on average, 14% (12%) of the dates in the test set correspond to high (low) -volatility days. In this way, we derive high-volatility and low-volatility subsets from the test set.

Table 4 displays the out-of-sample performance of the best model ($w = 10$, $\varepsilon = 0.45$, price-volume data) for the above two subsets. The results show that the prediction of the synchronization state is very satisfactory on both high-volatility and low-volatility test sets, with higher accuracy observed for the low-volatility set. Notably, the high-volatility test set has a higher percentage of non-synchronized time instances (12% in Class 0) than the low-volatility test set (5% in Class 0), which can be interpreted

Table 4 Performance measures on the high-volatility and low-volatility test sets

Volatility regime	Accuracy	Precision	Recall	F1-score
High	0.922	0.833	0.779	0.803
Low	0.982	0.933	0.853	0.889

as evidence of higher non-synchronization or disentangled dynamics on high-volatility days with respect to low-volatility ones. Interestingly, these percentages indicate that even on high-volatility days, stocks tend to be by far very synchronized (88%), though not as much as on low-volatility days (where the fraction of non-synchronized days is 7% smaller). The relatively low number of non-synchronized days for the high-volatility regime also suggests that stocks are similarly perturbed/exposed to the market risk factors causing volatility and that their response to volatility outbursts is similar. Indeed, in 88% of high-volatility test days, the data is detected in the same price-volume embedding, or, in simpler terms, two series share similar dynamics.

5 Conclusion

Predicting the co-movement of two multidimensional time series is a relevant task for the financial industry that supports potential trading strategies based on their inter-relationships. This paper contributes to the literature by providing (i) a method relying upon the CRP to quantify the time series coupling over time, (ii) a DL model for predicting the time series synchronization state, (iii) the use of a multidimensional time series representation of the inputs involving prices, volumes, and returns. We conducted extensive analyses on real stock market data from different sectors: our results show that the proposed setup can effectively predict the one-day-ahead synchronization of two time series. An interesting future research direction would be to investigate the applicability of such an approach to a high-frequency domain where the high-dimensional nature of the raw data may provide valuable information for analyzing and predicting the coupling in settings that are known to be characterized by high levels of noise.

Appendix A: Additional results

See Table 5.

Table 5 Performance on the test set for all the stock pairs in Table 1

Stock Pair	Sector	F1-score	Accuracy	Precision	Recall	Pct. Class 1	$w^\star$
CVX - HD	EM - RT	0.86	0.91	0.86	0.86	0.79	20
WMT - GOOG	TS - EM	0.85	0.86	0.85	0.86	0.40	20
CVX - MSFT	EM - RT	0.84	0.88	0.83	0.87	0.75	20
XOM - JPM	TS - RT	0.84	0.85	0.83	0.85	0.35	20
CVX - GOOG	EM	0.84	0.87	0.83	0.85	0.74	20
QCOM - CVX	ET - EM	0.82	0.83	0.83	0.82	0.56	20
JPM - WMT	RT - TS	0.82	0.82	0.87	0.82	0.53	20
V - GOOG	TS - EM	0.81	0.81	0.82	0.84	0.37	20
INTC - GOOG	RT - EM	0.81	0.81	0.81	0.82	0.41	20
XOM - WMT	TS	0.80	0.88	0.76	0.91	0.14	20
INTC - WMT	RT - TS	0.80	0.80	0.80	0.80	0.55	20
QCOM - GOOG	ET - EM	0.79	0.83	0.81	0.78	0.69	20
INTC - HD	RT	0.79	0.79	0.85	0.80	0.49	20
INTC - QCOM	RT - ET	0.77	0.78	0.78	0.78	0.45	10
JNJ - WMT	RT - TS	0.77	0.87	0.82	0.74	0.81	20
XOM - V	TS	0.76	0.85	0.72	0.86	0.15	20
INTC - MSFT	RT	0.76	0.76	0.76	0.75	0.56	20
QCOM - PFE	ET - RT	0.75	0.76	0.81	0.77	0.47	20
JPM - GOOG	RT - EM	0.75	0.81	0.86	0.73	0.65	20
CVX - V	EM - TS	0.75	0.75	0.75	0.75	0.47	20
XOM - HD	TS - RT	0.74	0.74	0.74	0.74	0.46	10
QCOM - V	ET - TS	0.73	0.74	0.73	0.75	0.35	20

Table 5 (continued)

Stock Pair	Sector	F1-score	Accuracy	Precision	Recall	Pct. Class 1	$w^\star$
PFE - GOOG	RT - EM	0.73	0.73	0.77	0.75	0.43	10
JPM - MSFT	RT	0.73	0.79	0.77	0.71	0.69	20
CVX - PFE	EM - RT	0.72	0.77	0.81	0.71	0.60	20
INTC - PFE	RT	0.72	0.75	0.73	0.83	0.21	20
XOM - JNJ	TS - RT	0.72	0.81	0.69	0.88	0.13	20
JPM - V	RT - TS	0.69	0.75	0.80	0.69	0.62	20
JPM - HD	RT	0.69	0.73	0.74	0.69	0.59	20
QCOM - XOM	ET - TS	0.69	0.69	0.69	0.69	0.50	20
QCOM - WMT	ET - TS	0.68	0.73	0.68	0.68	0.31	10
QCOM - JPM	ET - RT	0.68	0.68	0.73	0.72	0.41	20
XOM - MSFT	TS - RT	0.68	0.71	0.68	0.68	0.37	20
CVX - JPM	EM - RT	0.68	0.75	0.68	0.68	0.74	20
INTC - CVX	RT - EM	0.68	0.69	0.71	0.69	0.50	10
XOM - GOOG	TS - EM	0.67	0.69	0.68	0.67	0.39	20
JPM - PFE	RT	0.67	0.69	0.72	0.79	0.24	20
QCOM - JNJ	ET - RT	0.67	0.68	0.73	0.69	0.49	20
INTC - JPM	RT	0.67	0.69	0.71	0.67	0.55	20
INTC - JNJ	RT	0.67	0.73	0.74	0.66	0.62	20
CVX - WMT	EM - TS	0.67	0.68	0.66	0.67	0.37	20
V - MSFT	TS - RT	0.66	0.72	0.66	0.72	0.79	20
QCOM - MSFT	ET - RT	0.65	0.72	0.76	0.65	0.62	20
WMT - MSFT	TS - RT	0.65	0.65	0.65	0.65	0.55	20
INTC - V	RT - TS	0.64	0.74	0.64	0.65	0.77	20
JNJ - GOOG	RT - EM	0.64	0.66	0.67	0.64	0.54	10
CVX - JNJ	EM - RT	0.63	0.63	0.69	0.69	0.35	20
INTC - XOM	RT - TS	0.62	0.68	0.63	0.71	0.20	20
PFE - MSFT	RT	0.62	0.64	0.66	0.71	0.25	10
PFE - WMT	RT - TS	0.62	0.87	0.59	0.75	0.05	10
V - HD	TS - RT	0.60	0.61	0.60	0.60	0.58	20
JNJ - HD	RT	0.60	0.64	0.60	0.60	0.67	10
V - PFE	TS - RT	0.60	0.78	0.60	0.88	0.06	20
HD - WMT	RT - TS	0.59	0.59	0.59	0.59	0.42	20
XOM - PFE	TS - RT	0.57	0.63	0.58	0.57	0.63	10
JPM - JNJ	RT	0.56	0.63	0.59	0.57	0.62	10
V - WMT	TS	0.56	0.87	0.58	0.55	0.91	20
QCOM - HD	ET - RT	0.56	0.59	0.61	0.58	0.51	10
JNJ - MSFT	RT	0.54	0.81	0.54	0.55	0.90	20
HD - GOOG	RT - EM	0.53	0.63	0.56	0.54	0.65	10
HD - MSFT	RT	0.52	0.85	0.52	0.53	0.92	10
JNJ - PFE	RT	0.51	0.61	0.58	0.75	0.10	20
V - JNJ	TS - RT	0.50	0.86	0.50	0.50	0.94	10
XOM - CVX	TS - EM	0.49	0.52	0.54	0.56	0.22	10
MSFT - GOOG	RT - EM	0.48	0.94	0.47	0.50	0.94	20
PFE - HD	RT	0.44	0.44	0.61	0.62	0.22	20

“Pct. class 1” refers to the fraction of targets in class 1 (time series synchronization), “$w^\star$” is the value of the hyperparameter w under which the best F1-score is observed. Results refer to the 3-dimensional times series consisting of prices, volumes, and returns

Appendix B: RP and CRP Visualization

See Fig. 4.

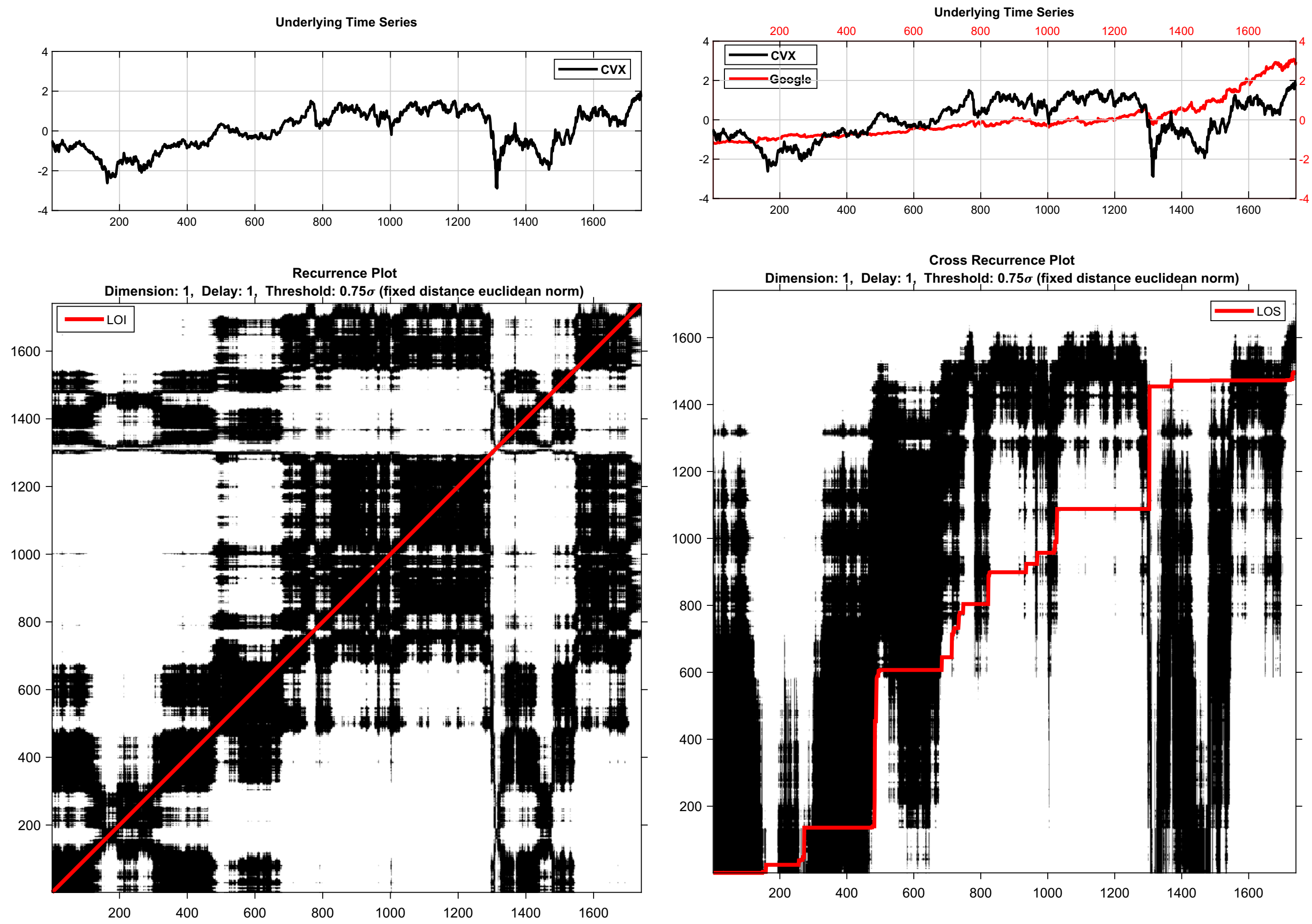

Fig. 4 Left column: visualization for the Chevron Corporation stock from January 1, 2015 to November 30, 2021. The Top Plot shows the normalized adjusted closing price, and the bottom plot shows the corresponding recurrence plot and the LOI. Right column: visualization of Chevron Corporation and Google stocks' daily adjusted closing prices from January 1 2015 to November 30, 2021. The top plot displays the normalized adjusted closing price of both stocks, while the bottom plot shows the CRP for this duration. The distorted red line on the CRP corresponds to the LOS. For generating the figure, we used the CRP toolbox [65]

Funding Open access funding provided by Royal Danish Library, Aarhus University Library. The research received funding from the Independent Research Fund Denmark project DISPA (project No. 9041-00004), and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie project BNNmetrics (grant agreement No. 890690).

Data Availability Statement The data supporting the findings of this study is freely available for.download at Yahoo finance at the address finance.yahoo.com/quote/XXX/history, where XXX corresponds to the ticker of each of the stocks listed in Table 1. Query the relevant period December 31st, 2014 to November 29th, 2021. The data consists of the daily closing prices and volumes on which the analyses are available.

Declarations

Conflict of Interests The authors have no relevant financial or non-financial interests to disclose that are relevant to the content of this article.

References

1. Murphy JJ (1999) Technical analysis of the financial markets: a comprehensive guide to trading methods and applications. Penguin, London
2. Lütkepohl H (1999) Vector autoregressions. Unpublished manuscript, Institut füur Statistik und Ökonometrie, Humboldt-Universität zu Berlin
3. Lütkepohl H (2013) Introduction to multiple time series analysis. Springer, New York
4. Reinsel GC (1993) 2. Vector ARMA time series models and forecasting. Springer, New York, NY, pp 21–51
5. Juselius K (2006) The cointegrated VAR model: methodology and applications. Oxford University Press, Oxford
6. Bauwens L, Laurent S, Rombouts JV (2006) Multivariate garch models: a survey. J Appl Economet 21(1):79–109
7. Harvey A, Ruiz E, Shephard N (1994) Multivariate stochastic variance models. Rev Econ Stud 61(2):247–264
8. Chiriac R, Voev V (2011) Modelling and forecasting multivariate realized volatility. J Appl Economet 26(6):922–947
9. Tong H (1978) On a threshold model in pattern recognition and signal processing. Sijhoff & Noordhoff, Amsterdam
10. Dv Dijk, Teräsvirta T, Franses PH (2002) Smooth transition autoregressive models-a survey of recent developments. Economet Rev 21(1):1–47
11. Tsay RS (2005) Analysis of financial time series. John Wiley & Sons, Hoboken, New Jersey, United States
12. Chen R, Tsay RS (1993) Functional-coefficient autoregressive models. J Am Stat Assoc 88(421):298–308
13. Chen R, Tsay RS (1993) Nonlinear additive arx models. J Am Stat Assoc 88(423):955–967
14. Cont R, Stoikov S, Talreja R (2010) A stochastic model for order book dynamics. Oper Res 58(3):549–563
15. Huang H, Kercheval AN (2012) A generalized birth-death stochastic model for high-frequency order book dynamics. Quant Financ 12(4):547–557
16. Hawkes AG (2018) Hawkes processes and their applications to finance: a review. Quant Financ 18(2):193–198
17. Heaton JB, Polson NG, Witte JH (2017) Deep learning for finance: deep portfolios. Appl Stoch Models Bus Ind 33(1):3–12
18. Dixon MF, Halperin I, Bilokon P (2020) Machine learning in finance, vol 1406. Springer, Switzerland
19. Sezer Berat O, Ozbayoglu Murat A, Dogdu Erdogan (2017) An artificial neural network-based stock trading system using technical analysis and big data framework. In: Southeast Conference, pp. 223–226
20. Zhang Z, Zohren S, Roberts S (2019) Deeplob: deep convolutional neural networks for limit order books. IEEE Trans Signal Process 67:3001–3012
21. Tran DT, Iosifidis A, Kanniainen J, Gabbouj M (2019) Temporal attention-augmented bilinear network for financial time-series data analysis. IEEE Trans Neural Netw Learn Syst 30(5):1407–1418
22. Passalis N, Tefas A, Kanniainen J, Gabbouj M, Iosifidis A (2020) Temporal bag-of-features learning for predicting mid price movements using high frequency limit order book data. IEEE Trans Emerg Topics Comput Intell 4(6):774–785
23. Shabani M, Tran DT, Kanniainen J, Iosifidis A (2023) Augmented bilinear network for incremental multi-stock time-series classification. Pattern Recogn 141:109604. https://doi.org/10.1016/j.patcog.2023.109604
24. Haselbeck F, Killinger J, Menrad K, Hannus T, Grimm DG (2022) Machine learning outperforms classical forecasting on horticultural sales predictions. Machine Learn Appl 7:100239
25. Khare K, Darekar O, Gupta P, Attar VZ (2017) Short term stock price prediction using deep learning. In: IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology, pp. 482–486
26. Fons E, Dawson P, Zeng X-j, Keane J, Iosifidis A (2021) Augmenting transferred representations for stock classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3915–3919
27. Bhandari HN, Rimal B, Pokhrel NR, Rimal R, Dahal KR, Khatri RKC (2022) Predicting stock market index using lstm. Machine Learn Appl 9:100320
28. Basher SA, Sadorsky P (2022) Forecasting bitcoin price direction with random forests: How important are interest rates, inflation, and market volatility? Machine Learn Appl 9:100355
29. Xu X, Zhang Y (2021) Network analysis of corn cash price comovements. Machine Learn Appl 6:100140
30. Shabani M, Iosifidis A (2020) Low-rank temporal attention-augmented bilinear network for financial time-series forecasting. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 2156–2161. IEEE
31. Shabani M, Tran DT, Magris M, Kanniainen J, Iosifidis A (2022) Multi-head temporal attention-augmented bilinear network for financial time series prediction. In: 2022 30th European Signal Processing Conference (EUSIPCO), IEEE, Belgrade, Serbia, pp 1487–1491. https://doi.org/10.23919/EUSIPCO55093.2022.9909957
32. Kyoung-Sook M, Hongjoong K (2019) Performance of deep learning in prediction of stock market volatility. Econ Comput Econ Cybernet Stud Res 53(2):77–92
33. Liu Y (2019) Novel volatility forecasting using deep learning-long short term memory recurrent neural networks. Expert Syst Appl 132:99–109
34. Christensen K, Siggaard M, Veliyev B (2021) A machine learning approach to volatility forecasting. J Financ Econ. https://doi.org/10.1093/jjfinec/nbac020
35. Guo X, Lai TL, Shek H, Wong SP-S (2017) Quantitative Trading: Algorithms, Analytics, Data, Models, Optimization. Chapman and Hall/CRC, Boca Raton, Florida, United States
36. Ma F, Wei Y, Huang D (2013) Multifractal detrended cross-correlation analysis between the chinese stock market and surrounding stock markets. Phys A Statist Mech Appl 392(7):1659–1670
37. Bonanno G, Lillo F, Mantegna RN (2001) High-frequency cross-correlation in a set of stocks. Quant Financ 1(1):96–104
38. Ramchand L, Susmel R (1998) Volatility and cross correlation across major stock markets. J Empir Financ 5(4):397–416
39. Webber CL Jr, Zbilut JP (1994) Dynamical assessment of physiological systems and states using recurrence plot strategies. J Appl Physiol 76(2):965–973
40. Marwan N, Kurths J (2002) Nonlinear analysis of bivariate data with cross recurrence plots. Phys Lett A 302(5):299–307
41. Lancia L, Fuchs S, Tiede M (2014) Application of concepts from cross-recurrence analysis in speech production: an overview and comparison with other nonlinear methods. J Speech Lang Hear Res 57(3):718–733
42. Webber CL, Marwan N (eds) (2015) Recurrence quantification analysis. Springer, Switzerland
43. Takens F (1981) Detecting strange attractors in turbulence. In: Rand D, Young L-S (eds) Dynamical Systems and Turbulence, Warwick 1980. Springer, Berlin, Heidelberg, pp 366–381
44. Eckmann J-P, Kamphorst SO, Ruelle D et al (1995) Recurrence plots of dynamical systems. World Sci Ser Nonlinear Sci Ser A 16:441–446

45. Marwan N (1999) Investigation of climate variability in NW Argentina using quantitative analysis of recurrence plots. Norbert Marwan
46. Fraser AM, Swinney HL (1986) Independent coordinates for strange attractors from mutual information. Phys Rev A 33(2):1134
47. Kennel MB, Brown R, Abarbanel HD (1992) Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys Rev A 45(6):3403
48. Schinkel S, Dimigen O, Marwan N (2008) Selection of recurrence threshold for signal detection. Eur Phys J Special Topics 164(1):45–53
49. Bastos JA, Caiado J (2011) Recurrence quantification analysis of global stock markets. Phys A Stat Mech Appl 390(7):1315–1325
50. Yin Y, Shang P (2016) Multiscale recurrence plot and recurrence quantification analysis for financial time series. Nonlinear Dyn 85(4):2309–2352
51. Zbilut JP (2005) Use of recurrence quantification analysis in economic time series. In: Salzano M, Kirman A (eds) Economics: Complex Windows. Springer, Milano, pp 91–104
52. Hailesilassie T (2019) Financial Market Prediction Using Recurrence Plot and Convolutional Neural Network. Unpublished manuscript
53. Li X, Kang Y, Li F (2020) Forecasting with time series imaging. Expert Syst Appl 160:113680
54. Sood S, Zeng Z, Cohen N, Balch T, Veloso M (2022) Visual time series forecasting: An image-driven approach. In: Proceedings of the Second ACM International Conference on AI in Finance, pp. 1–9
55. Han D, Orlando G, Fedotov S (2021) Identification of the nature of dynamical systems with recurrence plots and convolution neural networks: A preliminary test. arXiv preprint arXiv:2111.00866
56. Wallot S (2019) Multidimensional cross-recurrence quantification analysis (mdcrqa)-a method for quantifying correlation between multivariate time-series. Multivar Behav Res 54(2):173–191
57. Marwan N, Thiel M, Nowaczyk NR (2002) Cross recurrence plot based synchronization of time series. Nonlinear Process Geophys 9(3/4):325–331
58. Guhathakurta K, Marwan N, Bhattacharya B, Chowdhury AR (2014) Understanding the interrelationship between commodity and stock indices daily movement using ace and recurrence analysis. In: Marwan N, Riley M, Giuliani A, Webber CL Jr (eds) Translational Recurrences. Springer, Cham, pp 211–230
59. He Q, Huang J (2020) A method for analyzing correlation between multiscale and multivariate systems-multiscale multidimensional cross recurrence quantification (mmdcrqa). Chaos, Solitons & Fractals 139:110066
60. Tzagkarakis G, Dionysopoulos T (2016) Restoring corrupted cross-recurrence plots using matrix completion: Application on the time-synchronization between market and volatility indexes. In: Webber CL Jr, Ioana C, Marwan N (eds) Recurrence Plots and Their Quantifications: Expanding Horizons. Springer, Cham, pp 241–263
61. Addo PM, Billio M, Guegan D (2013) Nonlinear dynamics and recurrence plots for detecting financial crisis. North Am J Econ Financ 26:416–435
62. Fama EF, French KR (1993) Common risk factors in the returns on stocks and bonds. J Financ Econ 33(1):3–56
63. Sirignano J, Cont R (2019) Universal features of price formation in financial markets: perspectives from deep learning. Quant Financ 19(9):1449–1459
64. Riskmetrics T (1996) JP Morgan Technical Document. JP Morgan, New York
65. Marwan N, Romano MC, Thiel M, Kurths J (2007) Recurrence plots for the analysis of complex systems. Phys Rep 438(5–6):237–329

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