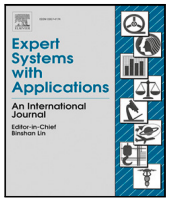
Expert Systems With Applications 260 (2025) 125457

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



CARROT: Simultaneous prediction of anomalies from groups of correlated cryptocurrency trends

Antonio Pellicani [a], Gianvito Pio [a,b,*], Michelangelo Ceci [a,b,c]

[a] Department of Computer Science, University of Bari, Via Orabona, 4, 70125 Bari, Italy
[b] Big Data Lab, National Interuniversity Consortium for Informatics (CINI), Via Volturno, 58, 00185 Roma, Italy
[c] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

ARTICLE INFO

Dataset link: https://figshare.com/collections/Processed_17S_Cryptocurrency_datasets/7359985

Keywords:
Anomaly prediction
Temporal clustering
Long short-term memory
Cryptocurrencies

ABSTRACT

Cryptocurrencies are virtual currencies that exploit cryptography to perform secure financial transactions. They gained widespread popularity in recent years due to their decentralized nature, (pseudo-)anonymity, and ability to facilitate cross-border transactions without the need for intermediaries. However, their price on the market exhibits a huge volatility, that makes them prone to market anomalies. Therefore, predicting anomalies in cryptocurrency time series can be considered an important task for financial institutions, traders, and investors, to maximize their profit or minimize losses.

In this paper, we propose a novel approach for predicting anomalies in cryptocurrency time series by exploiting temporal correlations among different cryptocurrencies. Our approach, called CARROT, is based on the idea that groups of cryptocurrencies exhibit similar trends, possibly due to common influencing factors. CARROT analyzes the temporal correlation between different cryptocurrencies, and identifies clusters showing similar patterns that can be useful for gaining insights into future anomalies. Subsequently, CARROT exploits multiple (i.e., one for each cluster) multi-target LSTM models to predict anomalies.

Our experiments, performed on a dataset of 17 cryptocurrencies, proved that CARROT outperforms single-target LSTM models of up to 20%, as well as other approaches based on neural networks, i.e., MLP and CNN, in terms of macro F1-score. Therefore, the proposed approach can be considered as a promising tool for predicting anomalies in cryptocurrency time series data and can potentially be used to improve risk management and trading strategies in the cryptocurrency market.

1. Introduction

Since its debut, the cryptocurrency industry has grown at an incredible rate, reaching a capitalization peak of $3 trillion in November 2021. A cryptocurrency is a digital currency that can be used to make online payments and exchanged for other cryptocurrencies or Fiat currencies. Cryptocurrencies are not subject to control by any central authority and their transactions are stored on the *blockchain*. Even if the blockchain was first introduced by Haber and Stornetta (1990) as a *cryptographically secured chain of blocks*, it gained popularity in 2008 after Satoshi Nakamoto introduced the first (and most popular) cryptocurrency named Bitcoin (Nakamoto, 2008). The goal of Bitcoin was to provide an algorithmic decentralized replacement to the traditional economic system, which entirely relies on centralized trusted third-party organizations (such as banks) to validate and execute monetary transactions. Besides Bitcoin, many other cryptocurrencies have been launched over time, each living on a blockchain with its own features and functionalities.

The environment related to the blockchain and cryptocurrencies, together with the technological advancements brought to several domains needing decentralization and transparency, has also attracted interest for its speculative aspects. Indeed, the huge market volatility observed for several cryptocurrencies (e.g., the Bitcoin price changed from a few cents in 2009 to around 60k$ in 2021, and is priced at around 25k$ at the time of writing) makes this environment incredibly appealing for individual and institutional investors, enthusiasts, and academics (Bhutta et al., 2021). An idea about the global trend of the cryptocurrency market capitalization (cap) and the daily trading volume from July 2018 to June 2022 is given in Fig. 1.

An accurate analysis of financial time series appears to be crucial for traders and institutional investors to increase the chances to make profits (or minimize losses), i.e., to discover patterns or manipulation activities, and to identify buy/sell signals. Although authoritative studies based on the "Efficient Market Hypothesis" (Fama, 1970) claim that

* Corresponding author at: Department of Computer Science, University of Bari, Via Orabona, 4, 70125 Bari, Italy.
E-mail addresses: antonio.pellicani@uniba.it (A. Pellicani), gianvito.pio@uniba.it (G. Pio), michelangelo.ceci@uniba.it (M. Ceci).

https://doi.org/10.1016/j.eswa.2024.125457
Received 20 October 2023; Received in revised form 23 July 2024; Accepted 24 September 2024
Available online 2 October 2024

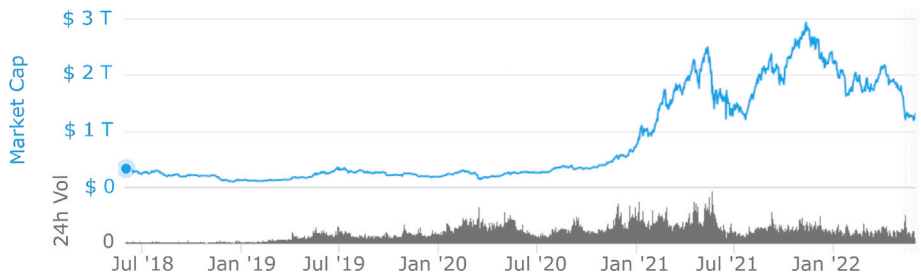


Fig. 1. Global cryptocurrency market cap (in blue) and daily volume (in gray) from June 2018 to May 2022.

financial markets follow a random walk and are unpredictable, the field of stock market analysis and prediction has attracted researchers in the Machine Learning field, whose obtained results suggest that stock (and cryptocurrency) trends can, at least partially, be predicted on the basis of historical data (Alvo et al., 2011; Caporale et al., 2018; Onali & Goddard, 2011). The cryptocurrency market, contrary to the traditional stock market, is also characterized by frequent, abrupt, sharp swings and falls, which make the predictive task even more difficult. Additional challenges come from the strong influence of external factors, such as global politics, market cycles, public opinion, or (also fake) news (Colon et al., 2021; Rognone et al., 2020; Wątorek et al., 2023). A relevant example can be observed in the effect caused by the publication of economic data from the US on the increase trading activity on cryptocurrencies described by Wątorek et al. (2023).

Existing state-of-the-art systems analyze historical market trends and additional factors influencing each cryptocurrency (Chen et al., 2020; La Morgia et al., 2020; Lahmiri & Bekiros, 2019). However, it is very common for the same factors to influence groups of related (in terms of technology, vision, or growth potential) cryptocurrencies. To the best of the authors knowledge, no existing method in the literature is able to capture the possible dependencies among different cryptocurrencies, possibly influenced by common external factors. In this paper we aim to fill this gap by proposing CARROT (*Clustering enhAnced cRyptocuRrency anOmaly predicTion*), a novel machine learning method that analyzes the cryptocurrency market and exploits possible dependencies among different cryptocurrencies, both in historical trends (in the descriptive space of the learned models) and in the future trends (in the target/output space of the learned models). Methodologically, CARROT first identifies groups of cryptocurrencies that appear to be influenced by the same factors since exhibited a similar trend in the past (in terms of price, statistical indicators and market sentiment). Subsequently, CARROT learns a deep multi-target model for each group, that is able to simultaneously predict the future occurrence of anomalous situations (i.e., abrupt price variations) for all the cryptocurrencies falling in the same group. Note that the task solved by CARROT is different from the classical anomaly detection task, which generally aims to learn a model able to label an already-observed instance as *normal* or *anomalous*. On the contrary, CARROT aims to **early predict the future occurrence of such anomalies**.

The remaining sections of the paper are structured as follows: in Section 2, we briefly review some related work; in Section 3, we describe the details of the proposed method; in Section 4, we show the results of our experimental evaluation. Finally, in Section 5, we draw some conclusions and outline potential directions for future research.

2. Related work

The analysis of time series related to cryptocurrencies belongs to the broader field of stock market and financial time series analysis, about which many studies have already addressed forecasting (Cao & Tay, 2001; Gupta & Dhingra, 2012; Sheta, 2006) and anomaly detection tasks (Al-Thani, 2017; Diaz et al., 2011; Sardar et al., 2022). To forecast the stock price, several existing studies adopted classical statistical time-series approaches based on historical data, such as moving average (MA) (Fifield et al., 2008), auto-regressive moving average (ARMA) (Tang, 2021), autoregressive integrated moving average (ARIMA) (Ariyo et al., 2014), generalized auto-regressive conditional heteroscedasticity (GARCH) (Franses & Van Dijk, 1996), Kalman filtering (Deepika & Nirupama Bhat, 2021) and exponential smoothing (Shukor et al., 2021).

More recently, researchers focused on the adoption of Artificial Intelligence (AI) and soft computing techniques. Indeed, the nonlinear, chaotic, noisy, and complex phenomena behind the stock market trends may be better governed through these techniques, possibly resulting in more accurate forecasts (Chen & Hao, 2017). The method CARROT proposed in this paper falls in this field of research, and, as already mentioned in Section 1, aims to predict the occurrence of future anomalies in the cryptocurrency market, i.e., abrupt price variation in the hourly market observations. For this reason, in Section 2.1, we briefly introduce existing methods aiming to solve forecasting tasks on financial time series. Moreover, since our method is tailored for the prediction of anomalies, in Section 2.2 we discuss some existing approaches for anomaly detection on financial time series, even if not able to specifically predict the future occurrence of such anomalies as done by CARROT.

2.1. Forecasting methods for financial time series

Despite the fact that we can easily identify several subtasks (each with some specifics), such as stock price forecasting, index prediction, forex price forecasting, commodity price forecasting, volatility forecasting, and cryptocurrency price forecasting, the underlying dynamics and, therefore, the methodologies proposed in the literature, are very similar.

Hu et al. (2013) exploited the Support Vector Machines (SVM) to predict the profitability of an investment over 15 different companies. The authors integrated data from the Federal Reserve Bank of St. Louis with four company-specific variables (net revenue, net income, price per earnings ratio of stock, diluted earnings per share) and six different macroeconomic variables (consumer spending, consumer investment, unemployment rate, inflation rate, federal funds rate, and the Dow Jones industrial average). To train the SVM model, the authors manually labeled each stock as *good* or *poor*, based on the performance of the company in the previous year. Similarly, SVM-based models have also been adopted by Gururaj et al. (2019), Huang et al. (2008) and Xia et al. (2013). However, in all these works, the results emphasized overfitting issues, mainly due to the presence of high amounts of noise in the dataset (Cawley & Talbot, 2010; Chiu & Chen, 2009; Yu et al., 2008). Moreover, these methods only consider one stock at a time, ignoring possible correlations that may exist among stocks of different companies in the construction of the predictive models.

Ballings et al. (2015) compared the performance of different machine learning algorithms when dealing with the stock price trend prediction. The experimental evaluation considered stocks of more than 5000 publicly listed European companies, with the goal of predicting their performance one-year ahead. The obtained results emphasized that Random Forests was the top-performing algorithm, followed by

SVM, AdaBoost, and K-NN. Moreover, the authors confirmed that considering financial technical indicators, like the Return on Assets (ROA), the Return on Equity (ROE), and the Return on Capital Employed (ROCE), can improve the predictive accuracy.

Due to the high accuracy demonstrated in several domains, deep learning methods and, more in general, approaches based on neural networks, found application also in the context of forecasting for financial time series. An example can be found in the work by Chong et al. (2017), who aim to predict the stock return. The authors exploited a simple multilayer perceptron, and investigated the influence of three unsupervised feature extraction methods. In the experiments, the performance of the neural network was compared against a traditional autoregressive model. The results obtained on a real dataset gathered from the KOSPI market indicated that the prediction ability was variable and dependent on a variety of environmental and user-determined parameters.

Focusing on the cryptocurrency market, Hasan et al. (2022) exploited convolutional neural networks (CNN) to predict the price of four different cryptocurrencies: Litecoin, Monero, Bitcoin, and Ethereum. The considered dataset consists of the OHLCV daily data, namely the opening price (O), the highest price (H), the lowest price (L), the closing price (C), and the exchange volume (V) of each analyzed cryptocurrency. Furthermore, the authors exploited Twitter APIs to scrape tweets about the crypto market in the analyzed time period, enriching the dataset with the sentiment score of the investors discussing the market conditions.

Awoke et al. (2021) used two recurrent neural networks (RNNs) for the daily forecast of the Bitcoin price. Specifically, they used the gated recurrent unit (GRU) (Bahdanau et al., 2014) and long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) architectures. The considered dataset covers the period from January 1, 2014 to February 20, 2018. Both models took into account a total of seven variables, including OHLCV information and market capitalization. Experimental results indicated that GRU-based models are more effective in forecasting highly volatile time series. Nevertheless, the analysis also revealed that LSTM exhibits superior performances when the measurements of only 7 previous days are used as descriptive features.

Multiple studies on sequence learning (Bao et al., 2017; Chimmula & Zhang, 2020; Fischer & Krauss, 2018; Ma et al., 2015; Yao et al., 2018) demonstrated that LSTM-based architectures can be considered state-of-the-art approaches for time series forecasting. They are, indeed, the most adopted deep learning model to deal with cryptocurrency price forecasting. A relevant example is the work by Yiying and Yeze (2019), who compared the performances of a simple fully-connected neural network with an LSTM model when trying to forecast the price of three popular cryptocurrencies (Bitcoin, Ethereum, and Ripple). Their results highlighted the effectiveness of LSTM in exploiting helpful information in historical memory.

Compared to existing approaches, the method CARROT proposed in this paper adopts several multi-target LSTM models instead of a single vanilla LSTM, each of which is trained on a group of similar cryptocurrencies. As a result, CARROT can consider the similar trend of other cryptocurrencies (which are likely to be influenced by similar factors) at training and prediction time, thus achieving better performances compared to existing models.

2.2. Anomaly detection methods for financial time series

Market anomaly detection, which aims to detect rapid changes in the direction of the market, can be considered a specific task of market trend analysis. Many existing works adopt anomaly detection approaches to recognize fraud or market manipulations. For example, Golmohammadi and Zaiane (2015) propose a Contextual Anomaly Detection (CAD) method to analyze financial time series from the S&P Index. First, the authors manually group the time series of the daily stock closing price, based on their industrial field. Then, for each identified group, they compute a centroid, which is used (together with other features) to predict the future price at two different granularities, i.e., daily and weekly. Finally, a prediction score is assigned to each analyzed point based on the Euclidean distance between the predicted and the observed values. If such a score exceeds the standard deviation of the analyzed time series, the point is considered as anomalous. Even though the approach is promising, we argue that an automated clustering phase based on the analysis of the historical trend, as done by CARROT, is much more appropriate than a grouping merely based on the industrial field.

Another significant example can be found in the work by Takara et al. (2021), who performed unsupervised anomaly detection via autoencoders on the Indice da Bolsa de Valores de Sao Paulo (IBOVESPA). The authors compared different autoencoder architectures and based the identification of anomalies on the reconstruction error. This represents a common strategy adopted by unsupervised anomaly detection methods, and requires comparing the actual data with the predicted value to obtain an anomaly score. Consequently, this approach cannot be adopted to predict the future occurrence of an anomalous situation, but only to detect it once it has been observed.

Focusing on the cryptocurrency market, several research groups shifted their focus on the identification of fraud and market manipulations. An example is the work by La Morgia et al. (2020), who adopted Random Forests and Logistic Regression to identify pump and dumps[1] in the Bitcoin time series data. Other valuable examples can be found in Sridhar et al. (2020) and Sridhar and Sanagavarapu (2021). The former work still focuses on detecting market manipulations by employing several ensembled feed-forward neural networks, while the latter has a more general focus on the identification of price and volume anomalies in the market.

We remind that none of the cited works focuses on predicting abrupt price variations that could affect the cryptocurrency market in the near future. This is a crucial task since it could allow investors to act proactively, selling or buying their securities and maximizing the obtained gain. Furthermore, only Golmohammadi and Zaiane (2015) tried to exploit the similarity between different financial time series to improve the final prediction. As already said, our framework CARROT uses the similarity between the different cryptocurrency time series to improve the ability to predict abrupt price variations and provide a concrete support to investors.

3. The proposed method CARROT

The workflow of the proposed method CARROT is illustrated in Fig. 2. The following subsections discuss the main steps performed to solve the anomaly prediction task on a specific cryptocurrency. Specifically, in Section 3.1 we describe the data preparation steps to solve the considered task; in Section 3.2 we explain the approach we adopt to identify groups of correlated cryptocurrencies, based on temporal clustering; finally, in Section 3.3 we provide details about the multi-target LSTM models we learn to capture such correlations.

3.1. Data preparation

CARROT starts with a preliminary phase consisting of acquiring, labeling, and adjusting data before the modeling phase. Specifically, the *Data Acquisition* phase relies on the Yahoo! Finance API, which allows access to financial information, including stock quotes, historical prices, company information, and crypto-market trends, with information on over 9000 unique coins. Using the API, we collected the following

[1] Pump and dump (P&D) is a type of securities fraud that entails boosting the price of an owned stock artificially by making false and deceptive positive statements, with the aim of massively selling the stock later at an inflated price, possibly causing a subsequent price drop.

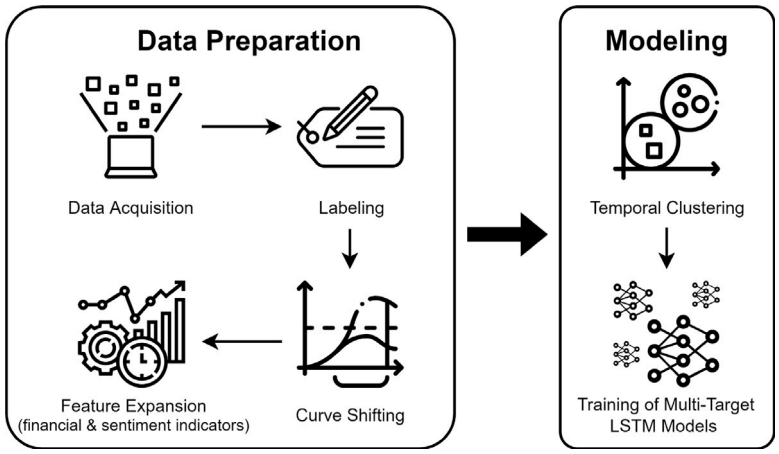


Fig. 2. General workflow of the proposed method CARROT.

features, at an hourly temporal granularity, in the form of multivariate time series: opening price, closing price, highest price, lowest price, adjusted closing price, and exchange volume. These features have often been adopted by other studies on the analysis of financial data (Awoke et al., 2021; Hasan et al., 2022).

Subsequently, the *Labeling* phase is responsible for labeling each observation as anomalous or normal. We remark that the model that we will learn from the observations labeled in such a way will aim to early *predict* the future occurrence of such anomalies, and not to identify them when they already occurred.

Methodologically, we rely on the definition of an anomaly in financial time series, that is, an abrupt variation in the price. For this purpose, given a threshold s, we label the data according to three different labels, i.e., *normal*, *upward anomaly (an_up)*, and *downward anomaly (an_down)*. Note that a low value of s may imply the presence of an unnaturally high number of anomalies. In contrast, a high value of s would lead to the identification of very few anomalous observations (Ko et al., 2022). However, it is noteworthy that defining a value for s does not represent a methodological choice, but can rather define a different goal: if we want to learn a model able to early predict slight changes, we can label the dataset with a low value of s; on the contrary, if we only want to make the model able to early predict huge changes, a higher value of s is more appropriate.

We also want to point out that the anomaly prediction task is generally more practically useful than forecasting the actual future value of a cryptocurrency. Indeed, in most financial time series applications, accurately forecasting the future price is not considered as crucial as accurately determining the direction of the market, which in turn may suggest a buy/sell signal.

More formally, given an observation x_t, its label is defined as:

$$label(x_t) = \begin{cases} an\_up & \text{if } \Delta_t \geq s, \\ an\_down & \text{if } \Delta_t \leq -s, \\ normal & \text{otherwise,} \end{cases} \tag{1}$$

where Δ_t is the percentage of variation computed on the pair $\langle C(x_t), C(x_{t-1})\rangle$ with $C(x_t)$ and $C(x_{t-1})$ corresponding to the hourly closing price for observation x_t and x_{t-1}, respectively.

After preliminary labeling all the hourly observations of the time series using Eq. (1), we employ the *Curve Shifting* approach, that is a label shift process, which is crucial for the considered anomaly prediction task. In fact, investors are interested in being alerted early about possible upcoming market anomalies to sell their assets before a market collapse, or to proactively buy new assets before a pump. Accordingly, we adopted some label-shifting rules to label the observations as anomalous, up to n hours before a real anomaly happens. Algorithm 1 briefly describes the adopted strategy. Note that if a n-hours interval contains market oscillations (with both up and down anomalies), we label all the observations as *normal*, since in that case, the market does not indicate a clear trend, and any trading action would not be based on clear indications.

Algorithm 1: The adopted curve-shifting algorithm

Data:
$D = \langle x_1, x_2, ..., x_{|D|}\rangle$: the dataset to process,
n: the desired number of previous observations to which propagate an anomaly label

begin
  for $t \leftarrow n$ **to** $|D|$ **do**
    // $l(x)$ is the label of the observation x
    if $l(x_t) = an\_up$ **then**
      if $l(x_{t-1}) \neq an\_down \wedge ... \wedge l(x_{t-n+1}) \neq an\_down$ **then**
        // We propagate the *an_up* label to the previous n observations
        $\langle l(x_{t-1}), ..., l(x_{t-n+1})\rangle \leftarrow \langle an\_up, ..., an\_up\rangle$
      else
        // We set to *normal* the label of the current and of the previous observations, since a market oscillation happened
        $\langle l(x_t), ..., l(x_{t-n+1})\rangle \leftarrow \langle normal, ..., normal\rangle$
      end
    else if $l(x_t) = an\_down$ **then**
      if $l(x_{t-1}) \neq an\_up \wedge ... \wedge l(x_{t-n+1}) \neq an\_up$ **then**
        // We propagate the *an_down* label to the previous n observations
        $\langle l(x_{t-1}), ..., l(x_{t-n+1})\rangle \leftarrow \langle an\_down, ..., an\_down\rangle$
      else
        // We set to *normal* the label of the current and of the previous observations, since a market oscillation happened
        $\langle l(x_t), ..., l(x_{t-n+1})\rangle \leftarrow \langle normal, ..., normal\rangle$
      end
  end
end

In Fig. 3, we show an example of the learning setting considered by CARROT, where the value of the parameter n for the curve shifting is set to 2.

Finally, CARROT performs a *Feature Expansion* phase, in which the feature set is expanded by considering 16 financial indicators, that, by

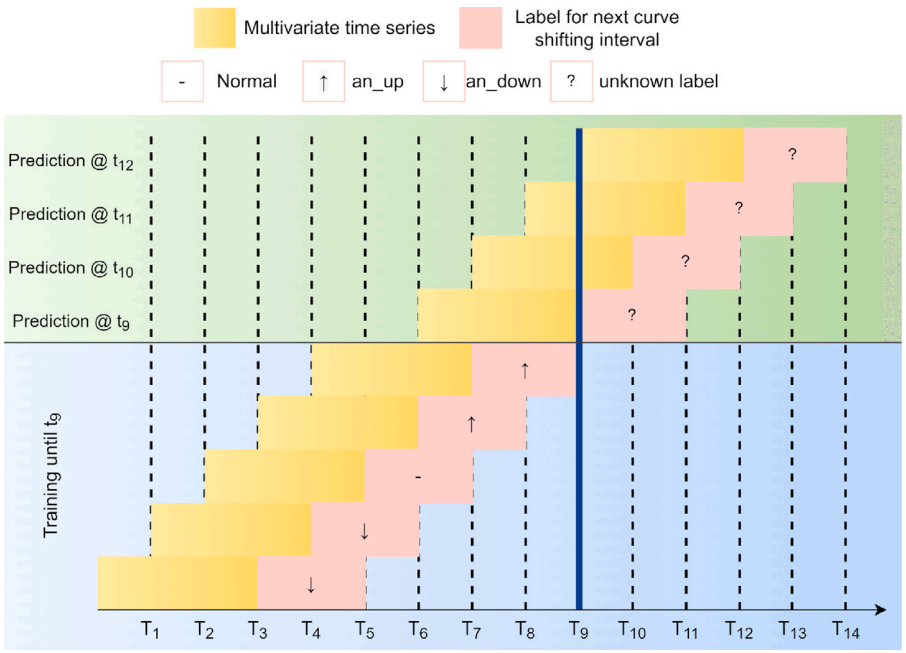


Fig. 3. An example of the CARROT learning setting with $l = 3$ (length of the sequence considered for each instance) and $n = 2$ (number of hours considered by the curve shifting).

varying the value of their parameters, lead to 48 new features. Table 1 shows an overview of the considered financial indicators, while further information about them can be found in the work by Achelis (2001). In addition, analogously to similar studies (de Oliveira Carosia et al., 2021; Karthikeyan et al., 2021; Valle-Cruz et al., 2022), we enrich our dataset by adding a Fear and Greed[2] indicator that takes into account the market sentiment, by simultaneously considering different perspectives, i.e., the market volatility and the market momentum, as well as social parameters like the posted hashtags and Google Trends indicators. In summary, the considered data are multivariate time series with a total of 61 features. Further details on the actual dataset adopted in our experiments will be provided in Section 4.1.

3.2. Temporal clustering

The price of cryptocurrencies is usually influenced by multiple external factors, i.e., rumors about businesses, new product releases, or political events. In the specific field of cryptocurrencies, these factors can simultaneously affect multiple assets (Sovbetov, 2018), also amplified by social networks. Therefore, it is crucial to recognize groups of cryptocurrencies exhibiting similar patterns in their trends, since they are probably affected by common external factors.

In CARROT, we group together cryptocurrencies exhibiting similar trends in terms of closing price through a specific clustering phase that exploits the Dynamic Time Warping (DTW) distance measure (Vintsyuk, 1968) and the clustering algorithm K-Medoids (Kaufman & Rousseeuw, 1990). The identified groups will subsequently be used to learn multi-target predictive models, able to capture the correlations among the cryptocurrencies fallen in the same group (further details about this step will be provided in Section 3.3).

DTW is one of the strategies that can be used in time series analysis to compare temporal sequences with possibly different speeds. Besides this characteristics, it is known to be superior to Euclidean distance due to its ability to compare time series with some offset and with different amplitude and length (Guijo-Rubio et al., 2020; Wang et al., 2013). This is achieved by identifying one-to-many and many-to-one matches between the time points of the time series. Fig. 4 shows an example of DTW applied to the closing price time series of two major cryptocurrencies. Specifically, the bold green line represents the closing price time series of ETH, while the bold yellow line represents the closing price time series of DASH. Each black-dashed line connects a point in one time series to its corresponding (most similar) point in the other time series. Note that if the considered time series were identical, all the black-dashed lines would have been straight vertical, and no (time) warping would have been required to line up the time series. The DTW distance is finally computed by adding up the distance between each pair of points linked by vertical lines.

Table 1
The financial indicators added to the dataset in the *Feature Expansion* phase. Each indicator may lead to multiple features, by varying the value of its parameter x.

Indicator	Description	Parameters
SMA_x	Simple Moving Average on x days	$x \in \{5, 12, 13, 14, 20, 21, 26, 30, 50, 100, 200\}$
EMA_x	Exponential Moving Average on x days	$x \in \{5, 12, 13, 14, 20, 21, 26, 30, 50, 100, 200\}$
RSI_x	Relative Strength Index on x days	$x \in \{5, 12, 13, 14, 20, 21, 26, 30, 50, 100, 200\}$
MACD	Moving Average Convergence/Divergence	Fast = 12, slow = 26, signal = 9
MACDH	Moving Average Convergence/Divergence Histogram	Fast = 12, slow = 26, signal = 9
MACDS	Moving Average Convergence/Divergence Signal	Fast = 12, slow = 26, signal = 9
STOCHF_x	Fast Stochastic Oscillator on x days	$x \in \{3, 14\}$
STOCH_x	Slow Stochastic Oscillator on x days	$x \in \{3, 5\}$
BBL_20	Bollinger Bands Lower	Length = 20
BBM_20	Bollinger Bands Mid	Length = 20
BBU_20	Bollinger Bands Upper	Length = 20
VWAP	Volume-weighted Average Price	–
MOM	Momentum	–
CMO	Chande Momentum Oscillator	–
DPO	Detrend Price Oscillator	–
UO	Ultimate Oscillator	–

[2] www.alternative.me/crypto/fear-and-greed-index.

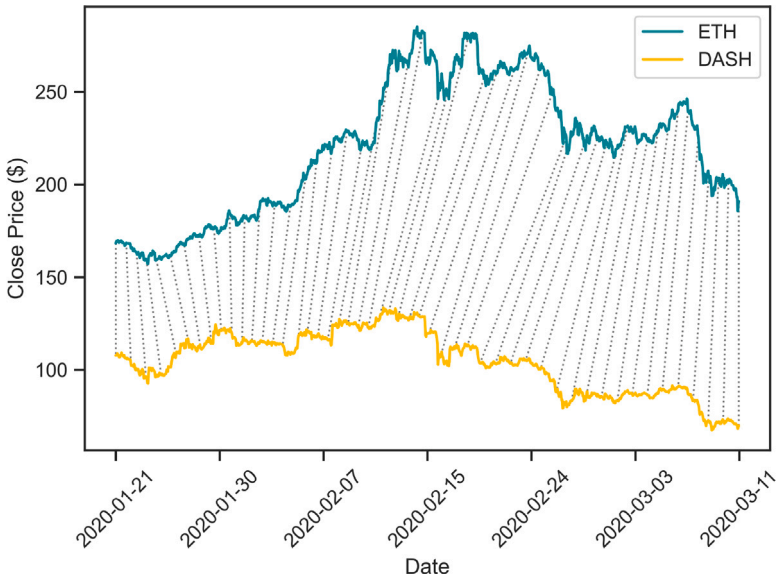


Fig. 4. A graphical example of DTW applied to the time series of the closing price of ETH and DASH, between 21 January 2020 and 11 March 2020. Each black dashed line connects a point in one time series to its corresponding (most similar) point in the other time series.

Formally, given two time series $\alpha = \langle \alpha_1, \alpha_2, \dots, \alpha_m \rangle$ and $\beta = \langle \beta_1, \beta_2, \dots, \beta_n \rangle$, the $DTW(\alpha, \beta)$ is computed as:

$$DTW(\alpha, \beta) = \sum_{i=1}^{z} |\alpha_{p_i} - \beta_{q_i}| \tag{2}$$

which is based on the so-called *warping path* $\langle (p_1, q_1), (p_2, q_2), \dots, (p_z, q_z) \rangle$, that is identified such that the warping path distance $\sum_{i=1}^{z} |\alpha_{p_i} - \beta_{q_i}|$ is minimized, subject to the following constraints:

- the starting point of the path corresponds to the first element of each time series, i.e., $(p_1, q_1) = (1, 1)$;
- the ending point of the path corresponds to the last element of each time series, i.e., $(p_z, q_z) = (m, n)$;
- for any given point (i, j) in the path, the previous point can assume only one of the following values: $(i-1, j)$, $(i, j-1)$, and $(i-1, j-1)$. This constraint guarantees that all the elements in α are matched to at least one corresponding element in β, and vice versa.

Algorithmically, the warping path can be found using dynamic programming (Eddy, 2004) to evaluate the cumulative distance $D(i, j)$ between two elements α_i and β_j of the two time series α and β. $D(i, j)$ is formally computed as the Euclidean distance between α_i and β_j plus the minimum of the cumulative distances of the adjacent elements, namely:

$$D(i, j) = (\alpha_i - \beta_j)^2 + min \begin{Bmatrix} D(i-1, j) \\ D(i, j-1) \\ D(i-1, j-1) \end{Bmatrix}, \tag{3}$$

having $D(1, 1) = (\alpha_1 - \beta_1)^2$.

In CARROT, we first compute a square and symmetric matrix Q by computing the pair-wise DTW distance among the hourly closing prices of the considered cryptocurrencies. When analyzing long-term time series, it may be important to avoid focusing on outdated patterns and trends that do not hold anymore. Therefore, we specifically focus the computation of the DTW on the most recent w months, in order to capture similarities among different cryptocurrencies in terms of current patterns and trends.

Finally, we adopt the k-medoids clustering algorithm using the distance matrix Q, in order to obtain k groups of cryptocurrencies exhibiting similar trends. K-medoids is a clustering algorithm that works similarly to k-means. However, instead of using the mean of the points in a cluster as the cluster center, also known as *centroid*, it uses the most centrally located point in the cluster, referred to as the *medoid*. K-medoids is also more robust to outliers than k-means because it uses actual data points as cluster centers. Moreover, it is more computationally efficient, since it only requires the computation of distances between data points and medoids (which are actual data points), instead of iteratively re-computing distances between all data points and centroids, which are updated at each iteration. This aspect also allows k-medoids to be fed with a pre-computed distance matrix, in the place of a distance function. In our case, we feed it with the distance matrix Q, pre-computed through DTW.

3.3. Anomaly prediction through multi-target LSTM models

The last phase performed by CARROT is in charge of training one multi-target model for each of the identified clusters, which is responsible for predicting future anomalies. As explained in Section 2, different models can be exploited to achieve this complex task. When dealing with sequential data, Recurrent Neural Networks (RNNs) are commonly adopted (Hüsken & Stagge, 2003). However, their major weakness is that they cannot effectively learn long-range dependencies, which are crucial in financial time series (Niu & Wang, 2013). In the plethora of RNN variations, the Long Short-Term Memory (LSTM) emerged as a model capable of considering long-range dependencies in time series, thanks to its memory cell provided with gates (Greff et al., 2016). Additionally, LSTMs help to overcome the vanishing gradient problem in RNNs while learning long-term contexts (Hochreiter & Schmidhuber, 1997; Pascanu et al., 2013). For these reasons, in CARROT we rely on LSTM, for which in the following we provide a brief overview of its architecture.

The output of an LSTM at a specific time step mainly depends on three factors: *(i)* the network's current long-term memory, also known as the *cell state*, *(ii)* the output from the previous time step, also known as the *earlier hidden state*, and *(iii)* the input data at the current time step. A set of gates regulates how data in a sequence enters, is stored, and leaves the network. Usually, a standard LSTM has three gates (i.e., a forget gate, an input gate, and an output gate) and is described by the following equations:

$$f_t = sigmoid\left(W_f x_t + V_f h_{t-1} + b_f\right) \tag{4}$$

$$i_t = sigmoid\left(W_i x_t + V_i h_{t-1} + b_i\right) \tag{5}$$

$$\tilde{C}_t = tanh\left(W_c x_t + V_c h_{t-1} + b_c\right) \quad (6)$$

$$C_t = f_t C_{t-1} + i_t \times \tilde{C}_t \quad (7)$$

$$o_t = sigmoid\left(W_o x_t + V_o h_{t-1} + b_o\right) \quad (8)$$

$$h_t = o_t \times tanh\left(C_t\right) \quad (9)$$

where x_t and h_t are the input and the hidden state at time step t, while W, V, and b are trainable network parameters representing the weight matrices and the biases: weights determine the level of connectivity between two neurons and control the magnitude of input influence on the output; biases provide a fixed input to the next layer and are not affected by the previous layer. Despite having no incoming connections, biases have their own outgoing connections with specific weights. Therefore, a bias unit ensures that the neuron will still be activated even when all inputs are zero.

The information from the previous hidden state and the current input are combined in the forget gate (see Eq. (4)). The sigmoid function guarantees an output vector containing values in the range $[0, 1]$. Specifically, the forget gate is a filter trained to output a score close to 0 when the knowledge from the earlier steps is irrelevant. On the other hand, the forget gate will output a score closer to 1 when the previous knowledge still needs to be considered.

The input gate (see Eq. (5)) performs a similar operation with respect to the forget gate: it combines the previous hidden state and the current input, and applies the sigmoid function. However, in this case, the score in the range $[0, 1]$ is useful to determine the importance of the new information provided by the input. At this point, the network has sufficient information to calculate the new cell state (see Eq. (7)), which is the sum of two pieces of information: the scaled old state value and the scaled input. The former is obtained by pointwisely multiplying the forget gate score by the previous cell state. Similarly, the latter can be obtained by pointwisely multiplying the input gate score by the new cell memory (see Eq. (6)). This *tanh* layer combines the previous hidden state and new input data, resulting in a vector that contains information from the new input data, given the context from the early hidden state and ranges in the interval $[-1, 1]$. In the final step, the new hidden state is computed using the last filter, i.e., the output gate (see Eq. (8)). Also in this case, the inputs are the same as the forget gate and the input gate (previous hidden state and new data). Then, in order to output the new hidden state (see Eq. (9)), the cell state goes through *tanh* layer (to push the values in the interval $[-1, 1]$) and is multiplied by the output gate.

In CARROT we do not consider the standard version of LSTMs, but we extend them to work in a multi-target setting. In particular, the proposed architecture learns a model from multiple time series and is able to simultaneously produce predictions of anomalies for all of them. We learn k deep multi-target models, i.e., one for each of the clusters discovered in the previous phase. The goal is to capture possible dependencies in the target space (as suggested by Caruana, 1997) among the time series of the cryptocurrencies grouped in the same cluster.

Typically, multi-target neural networks exploit multiple *heads*, where a head refers to the final layer (or set of layers) which is responsible for making predictions for a specific target (i.e., one head for each target), based on the features learned by the base model. Therefore, training multiple heads in parallel, while using a shared base model, leads to the optimization of the predictions of each head also on the basis of the others, thus capturing possible dependencies among the time series. Fig. 5 shows an example of a multi-target LSTM architecture trained by CARROT from a cluster of 3 time series. In our architecture, after the LSTM layers, each accepting sequences of length l (see Fig. 3 for an example with $l = 3$), we employ: *(i)* a dropout layer (between the last LSTM layer and the first dense layer represented with green circles in Fig. 5) to alleviate overfitting issues and help the model to generalize; *(ii)* three dense layers exploiting the ReLU activation function (green circles in Fig. 5); *(iii)* a final dense layer exploiting the softmax activation function to finally predict the label (violet circles in Fig. 5).

The weights of each head are updated in order to minimize the categorical cross-entropy loss function, which is formally defined as:

$$Loss(y, \hat{y}) = -\sum_{i=1}^{|D|}\sum_{k=1}^{3} y_{(i,k)} \cdot \log \hat{y}_{i,k} \quad (10)$$

where $y \in \mathbb{R}^{|D|\times 3}$ refers to one-hot-encoded actual labels of the $|D|$ training instances, and $\hat{y} \in \mathbb{R}^{|D|\times 3}$ refers to predicted probability distribution for the $|D|$ training instances, returned by the final softmax layer of a given head (represented as violet circles in Fig. 5). Note that one-hot-encoded labels and predicted probability distributions are represented as 3-dimensional vectors, since we have 3 different labels (*an_up*, *an_down*, and *normal*).

In order to simultaneously consider all the heads of the multi-target network associated with a cluster, we apply a mean strategy to compute the global loss of the cluster, as follows:

$$Global\ Loss(K) = \frac{\sum_{j\in K} Loss(y^{(j)}, \hat{y}^{(j)})}{|K|} \quad (11)$$

where K represents the specific cluster under consideration, containing the time series of multiple cryptocurrencies, while $y^{(j)}$ and $\hat{y}^{(j)}$ denote the one-hot-encoded actual labels and the predicted probability distribution, respectively, for the cryptocurrency (i.e., head/target) $j \in K$.

4. Experiments

In the following subsections, we first provide some details about the dataset considered in our evaluation, the experimental setting and the considered competitor approaches. Finally, we show and discuss the obtained results.

4.1. Dataset and experimental setting

As introduced in Section 3.1, for the evaluation of the performance exhibited by CARROT, we retrieved a real dataset exploiting Yahoo! Finance APIs. The obtained dataset contains hourly observations of 17 popular cryptocurrencies from 21 January 2020 to 31 December 2021. The considered cryptocurrencies are Bitcoin (BTC), BitShares (BTS), Dash (DASH), Digibyte (DGB), Dogecoin (DOGE), Ethereum (ETH), IOCoin (IOC), Litecoin (LTC), MaidSafeCoin (MAID), Monacoin (MONA), Navcoin (NAV), Syscoin (SYS), Vertcoin (VTC), Counterparty (XCP), Stellar (XLM), Monero (MNR), and Ripple (XRP). These cryptocurrencies have been selected due to their popularity, market capitalization, and trading volume. Indeed, cryptocurrencies with lower trading volume and fewer transactions often exhibit price movements mainly due to the lack of liquidity, rather than to real trends in trading operations.

The labeling of the dataset was performed following the strategy explained in Section 3.1. As regards the threshold s, in other studies related to stock market prediction (Feng et al., 2019), a value of $s = 0.5$ was adopted. In our work, since cryptocurrencies generally tend to be much more volatile than traditional stocks, we labeled the dataset using $s = 1.0$ (see the obtained label distribution for each cryptocurrency in Table 2).

In our study, we adopted the time series cross-validation setting, which preserves the temporal order of observations. In particular, each month of the dataset, starting from the 7th, is alternatively considered as testing set, while a given historical window (i.e., observations collected during a given number of months preceding the testing month) is considered as training set. Considering that the testing period ranges from July 2020 to December 2021, we actually performed a 18-fold time series cross-validation. As regards the training set, we considered different window sizes: 1 month (henceforth denoted with $\boldsymbol{D_{1m}}$), 3

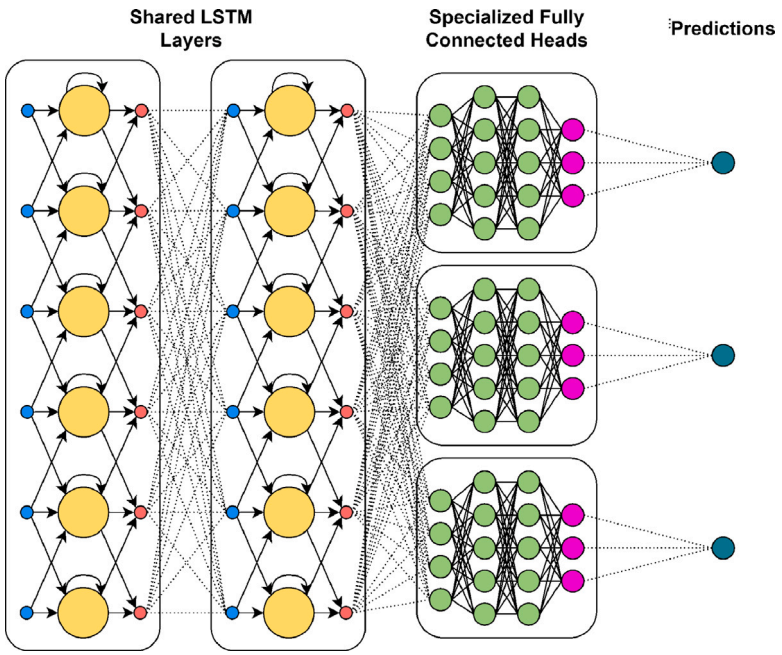


Fig. 5. An example of the multi-target LSTM architecture trained by CARROT for one cluster containing the time-series of three cryptocurrencies.

Table 2
Label distribution for each cryptocurrency in the considered dataset.

Cryptocurrency	Normal	an_up	an_down
BTC	10 579	3586	3379
BTS	4495	6574	6475
DASH	6191	5886	5467
DGB	3744	7033	6767
DOGE	6624	5289	5631
ETH	7695	5053	4796
IOC	3641	6921	6982
LTC	6486	5579	5479
MAID	5297	6200	6047
MONA	7255	5039	5250
NAV	3205	6616	7723
SYS	1921	7898	7725
VTC	769	8390	8385
XCP	6396	6263	4885
XLM	5684	6081	5779
MNR	5843	6028	5673
XRP	7273	5155	5116

months (henceforth denoted with $\boldsymbol{D_{3m}}$), 6 months (henceforth denoted with $\boldsymbol{D_{6m}}$), and the landmark setting (henceforth denoted with $\boldsymbol{D_{all}}$), that considers as training set all the available observations preceding the testing month. In Section 4.2, we will show and discuss the average results achieved over the 18 folds of the time series cross-validation.

We also evaluated the results considering different values for the time window w used to compute the DTW distance matrix $Q \in \mathbb{R}^{17\times17}$ for the clustering phase. Specifically, we collected the results with $w \in \{1, 3, 6\}$ (expressed in months). For k-medoids, we run the experiments with different values of its parameter k, namely, $k \in \{3, 5, 7\}$.

As a competitor approach, we considered multiple single-target LSTMs, each learned on one cryptocurrency (17 independent single-target LSTM models). The comparison with this approach allows us to properly assess the capability of CARROT to capture and exploit dependencies between different cryptocurrencies. Moreover, we run the experiments with two additional single-target neural network architectures, trained on each cryptocurrency: a multi-layer perceptron (MLP) and a convolutional neural network (CNN). The MLP consists of two hidden layers with 32 and 16 neurons, respectively, followed by an output layer with 3 neurons that adopt the softmax activation function. The CNN network uses two one-dimensional convolutional layers, followed by a max pooling layer, and a final dense architecture similar to that of the MLP.

For all the considered approaches, the size of historical data considered to represent each training instance was set to 30 h, which also corresponds to the size l of the sequences accepted in input by the LSTM layers.

As evaluation measure, we collected the macro average F1-score, which provides a general overview of the trade-off between precision and recall, and takes into account the possible unbalancing between the labels.

4.2. Results and discussion

We first analyze the percentage of F1-score improvement achieved by CARROT in comparison with single-target LSTMs, i.e., $\frac{\mathrm{F1}_{CARROT}-\mathrm{F1}_{LSTM}}{\mathrm{F1}_{LSTM}}\cdot100$, where $\mathrm{F1}_{CARROT}$ is the average F1-score obtained by CARROT over all the crypturrencies, and $\mathrm{F1}_{LSTM}$ is the average F1-score obtained by the multiple single-target LSTM models, each learned on a single cryptocurrency. In particular, in Table 3 we show the improvements for all 36 considered configurations (4 training intervals × 3 values for the parameter w × 3 values for the parameter k), averaged over the 18 folds of the time series cross-validation. From the table, we can immediately observe that CARROT was always able to obtain better results with respect to multiple single-target LSTMs (improvement > 0), with an average improvement of the macro F1-score of 10%. This result confirms that the approach adopted by CARROT, based on clustering cryptocurrencies time series and on training multi-target models, is actually more effective than training one single-target model for each of the considered cryptocurrencies. This confirms our initial intuition that leveraging possible dependencies, or common influencing factors, among different cryptocurrencies is clearly beneficial.

Looking at the improvements obtained in each configuration, it emerges that the historical interval adopted for the computation of the DTW, governed by the parameter w, does not significantly influence the results. This means that considering one month preceding the testing month ($w = 1$) can be considered enough to properly capture the similarities among cryptocurrency trends. Also the value of k, which determines the number of clusters, does not appear to significantly

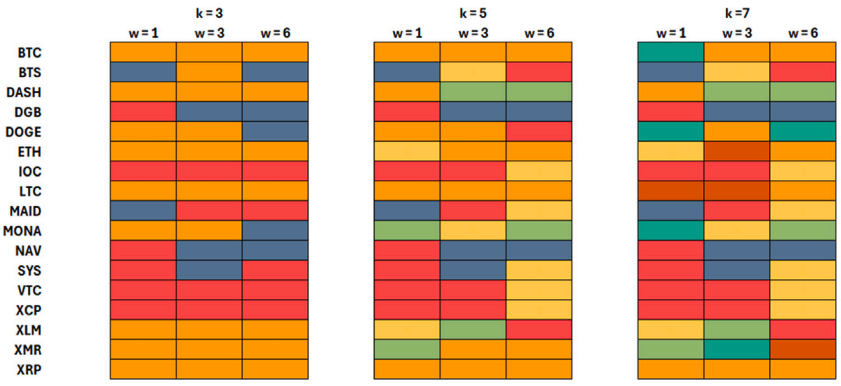


Fig. 6. Graphical representation of the clustering obtained by CARROT. Each color represents one of the k identified clusters. The graphical representation only refers to the first fold of the cross validation.

Table 3
Average percentage of improvement over the 18-fold of the cross validation, in terms of F1-score, obtained by CARROT with respect to multiple single-target LSTM models.

Training interval	DTW window (w)	# clusters (k)		
	(months)	3	5	7
D_{1m}	1	6%	6%	6%
	3	7%	7%	6%
	6	6%	7%	7%
D_{3m}	1	9%	8%	10%
	3	9%	10%	11%
	6	8%	9%	10%
D_{6m}	1	18%	19%	20%
	3	20%	19%	20%
	6	18%	20%	18%
D_{all}	1	3%	5%	8%
	3	2%	6%	8%
	6	3%	6%	7%

influence the results. This may be possibly motivated by the fact that, independently on the considered values of k, homogeneous groups of cryptocurrencies were identified and exploited by the learned multi-target models (see the clusters identified by CARROT for the first fold in Fig. 6).

Finally, analyzing the training interval, it emerges that the most significant improvement (in average, 19%) is obtained when the last 6 months of the dataset are considered during the training, i.e., in the setting D_{6m}. The reason for a lower improvement obtained when using all the available data (i.e., in the setting D_{all}) could lie in the fact that single-target LSTM models obtained good results because of more historical data available. This obviously leaves less margin for further improvement. In any case, also in the setting D_{all}, capturing and exploiting dependencies among cryptocurrency trends, as done by CARROT, led to an average improvement of the F1-score of 5%.

In Fig. 7, we draw some specific charts that allow us to better observe the influence of each of these parameters. Specifically, each chart represents the frequency (on the y-axis) of each value of the F1-score (on the x-axis), obtained by setting the value of a parameter and by varying the others. Looking at these charts, we can confirm that CARROT in terms of absolute F1-Score is rather stable with respect to different values of w and k. The first chart in Fig. 7 also confirms the superiority of the results when considering 6 months as training interval.

In Fig. 8, we report histograms that, for all the considered training intervals, depict the actual F1-scores obtained for each cryptocurrency by CARROT with $w = 3$ and $k = 7$, that can be considered one of the best configurations. Additionally, we show the performance obtained by the considered single-target competitors, namely MLP, CNN, and LSTM. In the same figure, we also represent, through horizontal lines,

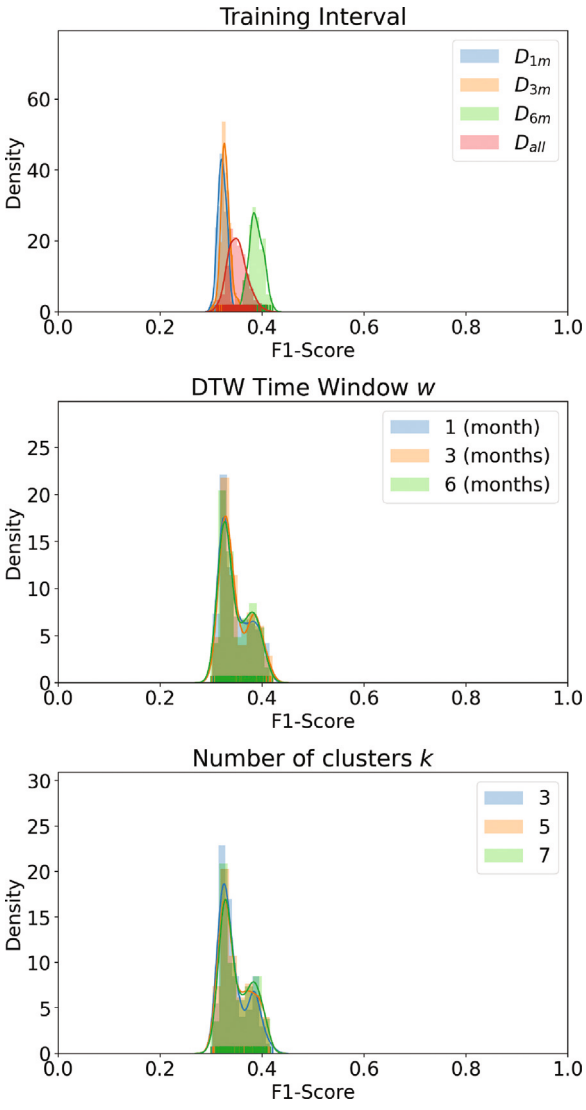


Fig. 7. Influence of the training interval (in the top), of the parameter w (in the middle), and of number of clusters k identified during the clustering phase by CARROT (in the bottom).

the average F1-score achieved by each method over all the considered cryptocurrencies, to evaluate its overall (cryptocurrency-independent)

Fig. 8. Histograms, representing the F1-score for each cryptocurrency, and horizontal lines, representing the average F1-score over all the cryptocurrencies, obtained by CARROT ($w = 3$ and $k = 7$) and by competitors.

performance. We can observe that the average F1-score increases as the training interval increases, even if the results show that exploiting more than 6 months during the training does not provide significant advantages. The training interval D_{6m} also appears to be the best when looking at individual cryptocurrencies. In this scenario, CARROT outperformed the set of 17 single-target LSTMs for all the 17 analyzed cryptocurrencies, with an average F1-score improvement of 20%.

In comparison with the CNN and the MLP, the superiority of CARROT is very clear. The CNN is able to compete with CARROT only in the D_{1m} training interval. This can be attributed to the limited amount of data available in the training phase, which may be insufficient for the multi-target models learned by CARROT to achieve a significant advantage by capturing useful patterns in the data and exploiting the inter-cryptocurrency relationships derived from the clustering phase. However, as shown by the average F1-scores represented by the horizontal lines in Fig. 8, when the training interval increases, the advantages of CARROT over single-target models becomes evident.

Finally, we analyzed the average rank achieved by CARROT and all the competitors along the different training intervals. Fig. 9 graphically depicts the results, from which we can observe that CARROT, with the training interval D_{6m}, obtains an average rank of 1.235, while its best competitor (i.e., single-target LSTMs with training interval D_{all}) obtains an average rank of 4.412. We additionally performed a Wilcoxon signed-rank test with the False Discovery Rate (FDR) correction for multiple tests proposed by Benjamini and Hochberg (1995) between the best configuration of CARROT and the best configuration of each of its competitors. The obtained results, that are reported in Table 4, emphasize that the difference is always statistically significant with a p-value < 0.01.

5. Conclusion

The prediction of anomalies in the trends of the cryptocurrency price is a complex task, mainly due to the high volatility of the market. However, by exploiting possible correlations among different cryptocurrencies, it is possible to improve the predictive accuracy of the models. This aspect can be valuable for investors and traders seeking to make more informed decisions.

In this paper, we proposed a novel method for predicting anomalies in cryptocurrency trends, called CARROT. The proposed method

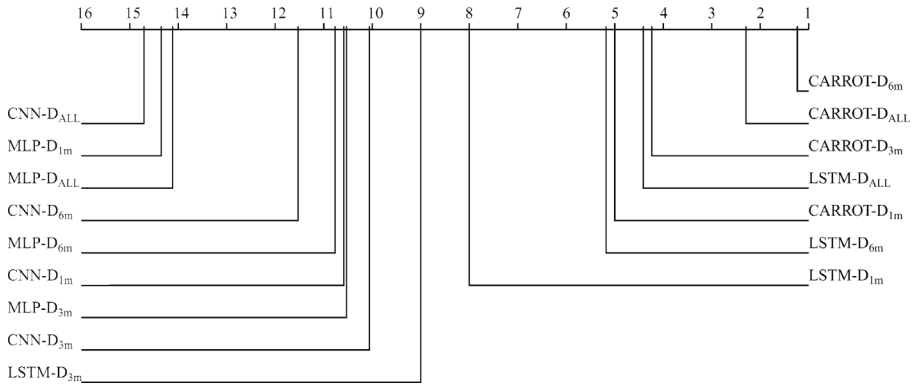


Fig. 9. Average rank obtained by all the methods along all the considered training intervals.

Table 4
Adjusted p-values of the signed Wilcoxon rank tests between the best configuration of CARROT and the best configuration of each of its competitors. We indicate in bold the statistically significant values (p-value < 0.01).

	Adjusted p-value	Winner
CARROT D_{6m} *vs.* LSTM D_{all}	**0.00027**	CARROT D_{6m}
CARROT D_{6m} *vs.* MLP D_{3m}	**0.00015**	CARROT D_{6m}
CARROT D_{6m} *vs.* CNN D_{3m}	**0.00015**	CARROT D_{6m}

exploits clustering of time series, based on DTW, to group the temporal data of related cryptocurrencies, to highlight temporal correlations among them, and to identify those potentially affected by common underlying factors. Then, for each group of cryptocurrencies, a multi-target LSTM model is trained to capture possible dependencies and simultaneously predict the occurrence of an anomaly for all the cryptocurrencies fallen in the group.

The performed experiments emphasized that capturing the dependencies among groups of correlated cryptocurrencies leads to significant improvements in terms of F1-score, with respect to multiple, independent single-target models.

Despite the promising results obtained, CARROT exhibits some limitations which will be the subject of future research. First, it operates in a batch learning fashion, by processing data at fixed intervals rather than continuously. This approach limits the model ability to adapt in real-time to rapidly-changing market conditions, which may be considered particularly crucial in the cryptocurrency domain. Future work could include the exploration of online learning techniques to address this limitation.

Second, our method demonstrates limited capability in detecting and adapting to concept drifts. This weakness may be the motivation behind the sub-optimal results obtained in the D_{all} setting. Indeed, the lower improvement over competitors obtained in this setting suggests that our approach may struggle to effectively leverage long-term historical data, possibly due to its inability to identify shifting patterns in the time series and adapt to them.

Last, the adoption of LSTM networks as base learners, while effective for capturing complex temporal dependencies, introduces some limitations in terms of explainability. The black box nature of LSTM models makes it difficult to interpret the motivations behind specific predictions. While obtaining an explanation in a trading context may not appear as critical as in other domains (e.g., medical), it may still improve the trustworthiness of the learned models. To address this limitation, future work could explore the integration of attention mechanisms or the adoption of base learners that are inherently more interpretable (e.g., tree-based methods).

CRediT authorship contribution statement

Antonio Pellicani: Conceptualization, Methodology, Formal analysis, Software, Investigation, Data curation, Validation, Writing – original draft. **Gianvito Pio:** Conceptualization, Methodology, Formal analysis, Data curation, Validation, Writing – review & editing. **Michelangelo Ceci:** Conceptualization, Formal analysis, Validation, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

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

Data availability

The pre-processed datasets for replication purposes are available at: https://figshare.com/collections/Processed_17S_Cryptocurrency_datasets/7359985.

Acknowledgments

The research presented in this paper was partially supported by the European Union - NextGenerationEU through the Italian Ministry of University and Research, projects: FAIR - Future AI Research (PE00000013) - Spoke 6 - Symbiotic AI, CN3 RNA - National Center for Gene Therapy and Drugs based on RNA Technology - CUP: H93C22000430007, and PRIN 2022 "COCOWEARS": A framework for COntinuum COmputing WEARable Systems - grant n. 2022T2XNJE - CUP: H53D23003650001.

References

Achelis, S. B. (2001). Technical analysis from A to Z.

Al-Thani, H. A. (2017). *Detecting market manipulation in stock market data* (Ph.D. thesis), Qatar University (Qatar).

Alvo, M., Firuzan, E., & Firuzan, A. (2011). Predictability of dow jones index via chaotic symbolic dynamics. *World Applied Sciences Journal*, *12*(6), 835–839.

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock price prediction using the ARIMA model. In *2014 UKSIM-AMSS 16th international conference on computer modelling and simulation* (pp. 106–112). http://dx.doi.org/10.1109/UKSim.2014.67.

Awoke, T., Rout, M., Mohanty, L., & Satapathy, S. C. (2021). Bitcoin price prediction and analysis using deep learning models. In *Communication software and networks* (pp. 631–640). Springer, http://dx.doi.org/10.1007/978-981-15-5397-4_63.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. http://dx.doi.org/10.48550/arXiv.1409.0473.

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, *42*(20), 7046–7056. http://dx.doi.org/10.1016/j.eswa.2015.05.013.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*, *12*(7), http://dx.doi.org/10.1371/journal.pone.0180944.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *57*(1), 289–300, URL: http://www.jstor.org/stable/2346101.

Bhutta, M. N. M., Khwaja, A. A., Nadeem, A., Ahmad, H. F., Khan, M. K., Hanif, M. A., Song, H., Alshamari, M., & Cao, Y. (2021). A survey on blockchain technology: Evolution, architecture and security. *IEEE Access*, *9*, 61048–61073. http://dx.doi.org/10.1109/ACCESS.2021.3072849.

Cao, L., & Tay, F. E. (2001). Financial forecasting using support vector machines. *Neural Computing and Applications*, *10*(2), 184–192. http://dx.doi.org/10.1016/S0925-2312(03)00372-2.

Caporale, G. M., Gil-Alana, L., & Plastun, A. (2018). Persistence in the cryptocurrency market. *Research in International Business and Finance*, *46*, 141–148. http://dx.doi.org/10.1016/j.ribaf.2018.01.002.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*(1), 41–75. http://dx.doi.org/10.1023/a:1007379606734.

Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, *11*, 2079–2107. http://dx.doi.org/10.5555/1756006.1859921.

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, *80*, 340–355. http://dx.doi.org/10.1016/j.eswa.2017.02.044.

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, *365*, Article 112395. http://dx.doi.org/10.1016/j.cam.2019.112395.

Chimmula, V. K. R., & Zhang, L. (2020). Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons & Fractals*, *135*, http://dx.doi.org/10.1016/j.chaos.2020.109864.

Chiu, D.-Y., & Chen, P.-J. (2009). Dynamically exploring internal mechanism of stock market by fuzzy-based support vector machines with high dimension input space and genetic algorithm. *Expert Systems with Applications*, *36*(2), 1240–1248. http://dx.doi.org/10.1016/j.eswa.2007.11.022.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, *83*, 187–205. http://dx.doi.org/10.1016/j.eswa.2017.04.030.

Colon, F., Kim, C., Kim, H., & Kim, W. (2021). The effect of political and economic uncertainty on the cryptocurrency market. *Finance Research Letters*, *39*, Article 101621. http://dx.doi.org/10.1016/j.frl.2020.101621.

de Oliveira Carosia, A. E., Coelho, G. P., & da Silva, A. E. A. (2021). Investment strategies applied to the Brazilian stock market: A methodology based on sentiment analysis with deep learning. *Expert Systems with Applications*, *184*, Article 115470. http://dx.doi.org/10.1016/j.eswa.2021.115470.

Deepika, N., & Nirupama Bhat, M. (2021). An efficient stock market prediction method based on Kalman filter. *Journal of the Institution of Engineers (India): Series B*, *102*(4), 629–644. http://dx.doi.org/10.1007/s40031-021-00583-9.

Diaz, D., Theodoulidis, B., & Sampaio, P. (2011). Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices. *Expert Systems with Applications*, *38*(10), 12757–12771. http://dx.doi.org/10.1016/j.eswa.2011.04.066.

Eddy, S. R. (2004). What is dynamic programming? *Nature Biotechnology*, *22*(7), 909–910.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417. http://dx.doi.org/10.2307/2325486.

Feng, F., Chen, H., He, X., Ding, J., Sun, M., & Chua, T.-S. (2019). Enhancing stock movement prediction with adversarial training. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 5843–5849). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2019/810.

Fifield, S. G. M., Power, D. M., & Knipe, D. G. S. (2008). The performance of moving average rules in emerging stock markets. *Applied Financial Economics*, *18*(19), 1515–1532. http://dx.doi.org/10.1080/09603100701720302.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. http://dx.doi.org/10.1016/j.ejor.2017.11.054.

Franses, P. H., & Van Dijk, D. (1996). Forecasting stock market volatility using (non-linear) garch models. *Journal of Forecasting*, *15*(3), 229–235. http://dx.doi.org/10.1002/(sici)1099-131x(199604)15:3<229::aid-for620>3.0.co;2-3.

Golmohammadi, K., & Zaiane, O. R. (2015). Time series contextual anomaly detection for detecting market manipulation in stock market. In *2015 IEEE international conference on data science and advanced analytics* (pp. 1–10). http://dx.doi.org/10.1109/DSAA.2015.7344856.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. http://dx.doi.org/10.1109/tnnls.2016.2582924.

Guijo-Rubio, D., Durán-Rosal, A. M., Gutiérrez, P. A., Troncoso, A., & Hervás-Martínez, C. (2020). Time-series clustering based on the characterization of segment typologies. *IEEE Transactions on Cybernetics*, *51*(11), 5409–5422. http://dx.doi.org/10.1109/tcyb.2019.2962584.

Gupta, A., & Dhingra, B. (2012). Stock market prediction using hidden Markov models. In *2012 students conference on engineering and systems* (pp. 1–4). http://dx.doi.org/10.1109/SCES.2012.6199099.

Gururaj, V., Shriya, V., & Ashwini, K. (2019). Stock market prediction using linear regression and support vector machines. *International Journal of Applied Engineering Research*, *14*(8), 1931–1934.

Haber, S., & Stornetta, W. S. (1990). How to time-stamp a digital document. In *Conference on the theory and application of cryptography* (pp. 437–455). Springer, http://dx.doi.org/10.1007/3-540-38424-3_32.

Hasan, S. H., Hasan, S. H., Ahmed, M. S., & Hasan, S. H. (2022). A novel cryptocurrency prediction method using optimum CNN. *Computers, Materials & Continua*, *71*(1), 1051–1063. http://dx.doi.org/10.32604/cmc.2022.020823.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hu, Z., Zhu, J., & Tse, K. (2013). Stocks market prediction using support vector machine. *Vol. 2*, In *2013 6th international conference on information management, innovation management and industrial engineering* (pp. 115–118). http://dx.doi.org/10.1109/iciii.2013.6703096.

Huang, C.-J., Yang, D.-X., & Chuang, Y.-T. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, *34*(4), 2870–2878. http://dx.doi.org/10.1016/j.eswa.2007.05.035.

Hüsken, M., & Stagge, P. (2003). Recurrent neural networks for time series classification. *Neurocomputing*, *50*, 223–235. http://dx.doi.org/10.1016/s0925-2312(01)00706-8.

Karthikeyan, D., Sivamani, B. A., Tummala, P. K., & Arumugam, C. (2021). Time series for forecasting stock market prices based on sentiment analysis of social media. In *Lecture notes in computer science*: *vol. 12955*, *Computational science and its applications - ICCSA 2021 - 21st international conference, Cagliari, Italy, September 13-16, 2021, proceedings, part VII* (pp. 353–367). Springer, http://dx.doi.org/10.1007/978-3-030-87007-2_25.

Kaufman, L., & Rousseeuw, P. J. R. (1990). Partitioning around medoids (program PAM). In *Finding groups in data* (pp. 68–125). John Wiley & Sons, Ltd, http://dx.doi.org/10.1002/9780470316801.ch2, chapter 2.

Ko, J. U., Na, K., Oh, J.-S., Kim, J., & Youn, B. D. (2022). A new auto-encoder-based dynamic threshold to reduce false alarm rate for anomaly detection of steam turbines. *Expert Systems with Applications*, *189*, Article 116094. http://dx.doi.org/10.1016/j.eswa.2021.116094.

La Morgia, M., Mei, A., Sassi, F., & Stefa, J. (2020). Pump and dumps in the bitcoin era: Real time detection of cryptocurrency market manipulations. In *2020 29th international conference on computer communications and networks* (pp. 1–9). http://dx.doi.org/10.1109/ICCCN49398.2020.9209660.

Lahmiri, S., & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals*, *118*, 35–40. http://dx.doi.org/10.1016/j.chaos.2018.11.014.

Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C (Emerging Technologies)*, *54*, 187–197. http://dx.doi.org/10.1016/j.trc.2015.03.014.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

Niu, H., & Wang, J. (2013). Volatility clustering and long memory of financial time series and financial price model. *Digital Signal Processing*, *23*(2), 489–498. http://dx.doi.org/10.1016/j.dsp.2012.11.004.

Onali, E., & Goddard, J. (2011). Are European equity markets efficient? New evidence from fractal analysis. *International Review of Financial Analysis*, *20*(2), 59–67. http://dx.doi.org/10.1016/j.irfa.2011.02.004.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310–1318). PMLR, http://dx.doi.org/10.48550/arXiv.1211.5063.

Rognone, L., Hyde, S., & Zhang, S. S. (2020). News sentiment in the cryptocurrency market: An empirical comparison with forex. *International Review of Financial Analysis*, *69*, Article 101462. http://dx.doi.org/10.1016/j.irfa.2020.101462.

Sardar, B. K., Pavithra, S., Sanjay, H. A., & Gogoi, P. (2022). Determining stock market anomalies by using optimized z-score technique on clusters obtained from K-means. In P. Shetty D., & S. Shetty (Eds.), *Recent advances in artificial intelligence and data engineering* (pp. 401–414). Singapore: Springer Singapore, http://dx.doi.org/10.1007/978-981-16-3342-3_32.

Sheta, A. (2006). Software effort estimation and stock market prediction using takagi-sugeno fuzzy models. In *2006 IEEE international conference on fuzzy systems* (pp. 171–178). http://dx.doi.org/10.1109/FUZZY.2006.1681711.

Shukor, S. A., Sufahani, S. F., Khalid, K., Wahab, M. H. A., Idrus, S. Z. S., Ahmad, A., & Subramaniam, T. S. (2021). Forecasting stock market price of gold, silver, crude oil and platinum by using double exponential smoothing, Holt's linear trend and random walk. *Journal of Physics: Conference Series*, *1874*(1), Article 012087. http://dx.doi.org/10.1088/1742-6596/1874/1/012087.

Sovbetov, Y. (2018). Factors influencing cryptocurrency prices: Evidence from bitcoin, ethereum, dash, litcoin, and monero. *Journal of Economics and Financial Analysis*, *2*(2), 1–27. http://dx.doi.org/10.1109/icimtech50083.2020.9211195.

Sridhar, S., Mootha, S., & Subramanian, S. (2020). Detection of market manipulation using ensemble neural networks. In *2020 international conference on intelligent systems and computer vision* (pp. 1–8). http://dx.doi.org/10.1109/ISCV49265.2020.9204330.

Sridhar, S., & Sanagavarapu, S. (2021). ELM-AD: Extreme learning machine framework for price and volume anomaly detection in stock markets. In *2021 international conference on computing and communications applications and technologies* (pp. 44–51). http://dx.doi.org/10.1109/I3CAT53310.2021.9629409.

Takara, L., Mariani, V., & Coelho, L. (2021). Autoencoder neural network approaches for anomaly detection in ibovespa stock market index. In C. J. A. B. Filho, H. V. Siqueira, D. D. Ferreira, D. W. Bertol, & R. C. L. ao de Oliveira (Eds.), *Anais do 15 congresso brasileiro de inteligência computacional* (pp. 1–8). Joinville, SC: SBIC, http://dx.doi.org/10.21528/cbic2021-37.

Tang, H. (2021). Stock prices prediction based on ARMA model. In *2021 international conference on computer, blockchain and financial development* (pp. i–iv). http://dx.doi.org/10.1109/CBFD52659.2021.00046.

Valle-Cruz, D., Fernandez-Cortez, V., López Chau, A., & Sandoval-Almazán, R. (2022). Does Twitter affect stock market decisions? Financial sentiment analysis during pandemics: A comparative study of the H1N1 and the COVID-19 periods. *Cognitive Computation*, *14*(1), 372–387. http://dx.doi.org/10.1007/s12559-021-09819-8.

Vintsyuk, T. K. (1968). Speech discrimination by dynamic programming. *Cybernetics*, *4*(1), 52–57. http://dx.doi.org/10.1007/BF01074755.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., & Keogh, E. (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, *26*(2), 275–309. http://dx.doi.org/10.1007/s10618-012-0250-5.

Wątorek, M., Skupień, M., Kwapień, J., & Drożdż, S. (2023). Decomposing cryptocurrency high-frequency price dynamics into recurring and noisy components. *Chaos. An Interdisciplinary Journal of Nonlinear Science*, *33*(8), Article 083146.

Xia, Y., Liu, Y., & Chen, Z. (2013). Support vector regression for prediction of stock trend. *Vol. 2*, In *2013 6th international conference on information management, innovation management and industrial engineering* (pp. 123–126). IEEE, http://dx.doi.org/10.1109/iciii.2013.6703098.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Li, Z., Ye, J., & Chuxing, D. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. In *32nd AAAI conference on artificial intelligence* (pp. 2588–2595). http://dx.doi.org/10.1609/aaai.v32i1.11836.

Yiying, W., & Yeze, Z. (2019). Cryptocurrency price analysis with artificial intelligence. In *2019 5th international conference on information management* (pp. 97–101). http://dx.doi.org/10.1109/INFOMAN.2019.8714700.

Yu, L., Chen, H., Wang, S., & Lai, K. K. (2008). Evolving least squares support vector machines for stock market trend mining. *IEEE Transactions on Evolutionary Computation*, *13*(1), 87–102. http://dx.doi.org/10.1109/tevc.2008.928176.