



Generative Adversarial Network to evaluate quantity of information in financial markets

Domenico Santoro¹ · Luca Grilli²

Received: 21 December 2021 / Accepted: 4 May 2022 / Published online: 29 May 2022
© The Author(s) 2022

Abstract

Nowadays, the information obtainable from the markets are potentially limitless. Economic theory has always supported the possible advantage obtainable from having more information than competitors, however quantifying the advantage that these can give has always been a problem. In particular, in this paper we study the amount of information obtainable from the markets taking into account only the time series of the prices, through the use of a specific Generative Adversarial Network. We consider two types of financial instruments traded on the market, stocks and cryptocurrencies: the first are traded in a market subject to opening and closing hours, whereas cryptocurrencies are traded in a 24/7 market. Our goal is to use this GAN to be able to “convert” the amount of information that the different instruments can have in discriminative and predictive power, useful to improve forecast. Finally, we demonstrate that by using the initial dataset with the 5 most important feature used by traders, the prices of cryptocurrencies present higher discriminatory and predictive power than stocks, while by adding a feature the situation can be completely reversed.

Keywords Generative adversarial network · Deep learning applications · Market information analysis · Stock price analysis

1 Introduction

The problem of information present in the markets and of information that can be released at a company level has always been fundamental for understanding how a price is determined, as well as at a regulatory level. As early as 1945, Hayek [1] described how prices, in a system where relevant information is dispersed, can coordinate the actions of different subjects. In modern financial markets, investors benefit from the information available by selecting only what they believe is important.

In recent times, the amount of information available to investors has increased exponentially [2] thanks to the news provided by companies, creating a problem of optimal selection of the most important information compared to those that could only “deceive” the investor. Thus, an open question remains how much current information dissemination capabilities are able to impact price efficiency in incorporating new information.

The first forms of analysis of the price effects of information concern disclosures of corporate reporting [3]. The disclosure of information highlights managerial talent and is able to explain crisis situations [4, 5], or it is able to reduce the information asymmetry by preventing investors from exchanging private information [6]. In contrast to this view, the increase in complexity in the information provided could lead to problems with the legibility of a news [7], or it could be used to obfuscate negative news [8]. Some studies [9–11] examine the effectiveness of numerical and textual information on price discovery, highlighting how the combination of high levels of numerical information combined with graphic information can have a great impact [12] on investors and on the choice of price; while in this case we face the inverse situation, that is to understand from the prices what is the level of information

Domenico Santoro and Luca Grilli have contributed equally to this work.

✉ Luca Grilli
luca.grilli@unifg.it
Domenico Santoro
domenico.santoro@uniba.it

¹ Department of Economics and Finance, University of Bari, Bari, Italy

² Department of Economics, Management and Territory, University of Foggia, Foggia, Italy

contained and the consequences that this can have on the forecast. This is where this paper focuses, in investigating the market hours problem associated with financial information. However, our starting point is not to consider textual information (deriving for example from company statements), but to use only the prices recorded on the markets to understand what level of information these prices embody. In particular, this consideration is carried out over small time intervals to highlight the difference between closing markets and always open markets.

Many markets are subject to opening and closing times. In such types of markets, when news spreads or events occur after closing hours, price reactions can only occur after the next opening of the market. In contrast, the cryptocurrency market (and currencies in general) is not subject to closing times. In this type of market, the “opening” price of the new day and “closing” price of the previous day are recorded at midnight each day, creating continuity in the historical price series. Thus, the recorded prices are assumed to be the sum of events occurring in a 24-h period. Therefore, these prices may contain more information useful for forecasting. The objective of this study is to verify how this difference of *information in price* can result in forecast imbalances when using an appropriate neural network. We focus on the discriminative and predictive power of prices. Previous studies, such as those of Tang et al. [13] and Gorr [14], have demonstrated that neural networks can model seasonality and other factors, e.g., trends and cyclicity, autonomously. Therefore, different “*quantities of information*” contained in various types of prices would seem to be the only cause resulting in forecast imbalances. Datasets with different temporal structures would contain different amounts of information, resulting in differences with respect to the degree of predictability related to associative learning tasks [15].

The remainder of this paper is organized as follows. In Sect. 2, we present some of the most important literature in the sector, in Sect. 3, we analyze the architecture of *Generative Adversarial Network* (GAN). Herein, we focus on TimeGAN, which is used to extrapolate the characteristics of different features in time-series data. In Sect. 4, we define our model and methodology. In Sect. 5, we apply this special GAN to the time-series data of stocks and cryptocurrencies and compare the obtained results, extending the previous datasets by adding a feature and verifying the results. Finally, conclusions are presented in Sect. 6. In “Appendix”, the most used neural network architectures to make financial market predictions, *Long–Short Term Memory* (LSTM) and *Convolutional Neural Network* (CNN), are presented.

2 Literature review

Moving away from works in which give the information we try to understand what the impact on the price is, finding ourselves in the diametrically opposite situation, we can consider only time series, whose analyzes has always attracted the attention of academia, especially for predicting the future values in a series. Financial time series are optimal candidates for such an analysis. They base their assumptions on the *random walk hypothesis*, a concept introduced by Bachelier [16] in 1900, which has remained the central pivot in the theory of time series. Based on the random walk hypothesis, Kendall [17] assumed that the stock price movement was random, whereas Cootner [18] indicated how the stock price movement could not be explained in detail but could be better approximated based on the Brownian motion. Traditionally, the best practice has been to focus on logarithmic returns, which provides the benefit of linking statistical analysis with financial theory. In his *efficient market hypothesis* (EMH) theory, Fama [19] introduced a concept that historical prices are factored into the current prices in a given market. However, deploying such historical data in any analysis would be less useful (if not completely useless) for predicting future prices. LeRoy [20] demonstrated that the concentration on yields was unjustified and concluded that the stock markets are inefficient. Taylor [21] proposed an alternative price trend model and provided empirical evidence that the price trend model was useful for analyzing the future prices in markets.

From an econometric perspective, Box and Jenkins [22] introduced power transformations to statistical models and applied them to a time series. Specifically, they suggested to use power transformation for obtaining an adequate *Autoregressive Moving Average* (ARMA) model. Based on this, Hamilton [23] provided a formal mathematical definition of this model. Several evolutions have followed this pattern, e.g., *Autoregressive Integrated Moving Average* (ARIMA) and *Seasonal Autoregressive Integrated Moving Average* (SARIMA).

2.1 Machine learning

Recently, because of the developments of *Artificial Neural Networks* (ANNs) and their suitability in nonlinear modeling [24], there has been considerable interest in applying such methods to time-series prediction in the *Machine Learning* (ML) framework. For example, Foster et al. [25] were among the first to compare the use of neural networks as function approximators with the use of the same networks to optimally combine classical regression methods. Further, they highlighted the manner in which using

networks to combine forecasting techniques provided performance improvements. In addition, Refenes et al. [26] used a neural network system to forecast the exchange rates via a feedforward network, which faced difficulty in predicting the turning points despite an accuracy of 66%. Sharda and Patil [27] compared the predictions obtained using neural networks and the Box–Jenkins model. Based on this comparison, they verified that neural networks performed better than expected for time series with long memory. In contrast, the neural networks outperformed the prevision for time series with short memory. Andrawis et al. [28] combined the forecasts obtained via different time aggregation, and Adeodato et al. [29] proposed a methodology based on an ensemble of Multilayer Perceptron networks (MLPs) to achieve robust time-series forecasting. In addition, Wichard [30] proposed a hybrid model to forecast a time series with recurring seasonal periods using separately trained models. A considerable innovation was introduced in the *support vector machine* (SVM) model [31], thereby solving the pattern classification problem. In particular, its use was immediately extended to regression, with subsequent application to time-series forecasting [32]. Over time, several SVM-based models have been developed, e.g., the *least-square SVM* (LS-SVM) [33]. More recently, Xiao et al. [34] used a cumulative auto-regressive moving average which combines the least squares Support Vector Machine model (ARI-MA-LS-SVM), to make basic predictions for the stock market; highlighting how this model is more effective for stock price forecasting than a classical forecasting model. In addition, Kovalerchuk and Vityaev [35] used various models, such as the *Evolutionary Computation* (EC) and *Genetic Programming* (GP) models, Zaccagnino et al. [36, 37] developed automatic ML-based methods to provide privacy awareness to users and total control over their data during online activities, and Li and Ma [38] developed a model to forecast the stock price using an ANN. Mittelmayer and Knolmayer [39] compared different text mining techniques for extracting market response to improve prediction; and Mitra [40] focused on studying news to predict the anomalous returns associated with trading strategies.

2.2 Deep learning

In case of Deep Learning (DL) techniques, increasingly complex architectures have been used, especially in the previous decade [41]. For example, Liu et al. [42] used a CNN-LSTM for strategic analysis in financial markets; Zhang et al. [43] used *structure for motion* (SFM) to predict stock prices by extracting different types of patterns; Jin et al. [44] propose a decomposition for stock price by an Empirical Model Decomposition (EDM), subsequently

used an LSTM with Attention Mechanism to improve prediction; or Lu et al. [45] that used a combination of CNN for extracting features, biLSTM to predict a stock price for the next day and the Attention Mechanism to capture the influence of features on the closing price. However, many other types of more complex networks can be readjusted to the time series to make predictions, e.g., GANs for speech synthesis [46], denoising images [47], or an imbalanced generative adversarial fusion network (IGAFN) to integrate credit data into a unified latent feature space, as demonstrated by Lei et al. [48]. In addition, other combinations of architectures are represented by the *deep convolutional neural network* (DCNN) and *conditional random field* (CRF) networks as proposed by Papandreou et al. [49] for high-resolution segmentation and a *deep relative distance learning* (DRDL) network as proposed by Liu et al. [50] to calculate the distance between vehicles via graphical analysis by translating them into the Euclidean space. Finally, this methodology that we intend to apply to the financial sphere about quantities of information, can be implemented in other promising scenarios, e.g., Blockchain [51], R&D [52], or supply chain [53].

3 Neural network architectures

The most generally used network architectures in financial time series forecasting are *Long–Short Term Memory* (LSTMs), particularly for the ability of adapting to the features considered. However, more recently, this sector has opened up to new architectures such as *Convolutional Neural Networks* (CNNs, whose use is justified by considering features no longer as mere time series but, in many cases, images) and *Generative Adversarial Networks* (GANs). A more accurate description of how LSTM and CNN work can be found in “Appendix”. Below, however, we will focus only on GANs (and on a particular version) to highlight the amount of information available on the markets.

3.1 Generative adversarial network

Generative models are formulated based on an approach developed in accordance with the Bayes’ theorem. Generative models consider sensorial hypotheses about the input to modify the parameters characterizing them. The learning mechanism involves maximization of the likelihood of data relative to the generative model; this corresponds to the discovery of efficient methods for encoding the input information. For financial time-series forecasting, the most common generative model is *Generative Adversarial Network* (GAN), which was introduced by

Goodfellow et al. [54] in 2014. The GAN model comprises two networks, i.e., a generative network G that produces new data based on a certain distribution p_g and a discriminative network D that evaluates them, resulting the probability of $x \sim p_{\text{data}}$, where p_{data} indicates the distribution of training data. The network's objective [46] is to encourage D to find a binary classifier that provides optimal discrimination between the real and generated data and simultaneously encourage G to fit the true data distribution. D and G play the following two-player minimax game [54] with the value function $V(G, D)$.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

This type of neural network is one of the most complex and is generally used in complex sectors. Yoon et al. [55] adopted a version of it to accept time-series inputs and generate synthetic data.

3.1.1 TimeGAN

One of the main problems associated with time-series forecasting is the selection of optimal variables such that the neural network can capture their links and dynamics over time. In particular, Yoon et al. [55] proposed the *Time-Series Generative Adversarial Network* (TimeGAN) to generate realistic time-series data in various domains. TimeGAN considers unsupervised adversarial loss and stepwise supervised loss and uses the original data for supervision. TimeGAN comprises four networks [55]: the embedding function, the recovery function, the sequence generator, and the sequence discriminator. The autoencoding components are trained jointly with the adversarial components such that TimeGAN simultaneously learns to encode features, generate representations, and iterate over time. Typically, GANs are used (regarding a financial time series) for the generation and replacement of any missing values (NaN). However, in this case, the main objective is to recreate a time series based on the features considered as input.

In this model, the generator is exposed to two types of inputs during training, i.e., synthetic embeddings to generate the next synthetic vector and sequences of embeddings of actual data for generating the next latent vector. In the first case, the gradient is computed based on unsupervised loss, whereas the gradient is computed based on supervised loss in the second case.

4 Methods and materials

To highlight the generated results, Yoon et al. [55] proposed a graphical measure for visualization, i.e., t-SNE (van der Maaten and Hinton [56], to visualize the similarity of the generated distribution with respect to the original distribution) and two scores (obtained by optimizing a two-layer LSTM).

- *Discriminative score*: This factor indicates the error of a standardized classifier (RNN) when distinguishing the real sequence and the sequence generated based on a test set using a post-hoc RNN network.
- *Predictive score*: This factor evaluates the prediction performance of synthetic data. Here, a post-hoc RNN architecture is employed for prediction. Further, the performance is reported in terms of *mean absolute error* (MAE), which measures the ability of synthetic data to predict the next-step temporal vectors. For example, the MAE between paired observations is defined as

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (2)$$

In addition, the t-SNE algorithm also returns Kullback–Leibler (KL) divergence

$$D_{\text{KL}}(P||Q) = \sum_i P(i) \log_2 \left(\frac{P(i)}{Q(i)} \right), \quad (3)$$

which is a measure of the difference between two probability distributions P and Q [57] that indicates the information lost when using a distribution (in this case, the synthetic distribution) for approximating another distribution (the original distribution). We employ this network to demonstrate that the financial instruments listed on a market subject to time constraints have less predictive power than the instruments traded on a 24/7 market. The financial instruments that are subject to timetables during the continuous trading phase are representative of the information presented during those hours; however, the events that occur after closing is not reflected (immediately) by the price and will be recorded only on the following day during the opening auction. In contrast, for instruments not subject to schedules, this problem does not occur because any event that may affect the price will be recorded, affecting the price dynamics. Exchanges offer the possibility to conduct negotiations outside closing hours (*trading in premarket and after hours*), as in the case of Borsa Italiana, where the preauction phase is from 08:00 to 09:00 and after-hours trading is from 05:50 to 8:30 p.m., and NASDAQ, where premarket trading is from 04:00 to 09:30 (ET) and after-hours trading is from 4:00 to 8:00 p.m. (ET). However, certain time slots remain uncovered. Therefore, the “*amount of information*” associated with

Table 1 List of stocks and cryptos used

Stocks	Cryptocurrencies
AAPL (Apple Inc.)	BTC (Bitcoin Index)
GOOG (Alphabet Inc.)	ETH (Ethereum Index)
AMZN (Amazon.com)	BCH (Bitcoin Cash Index)
TSLA (Tesla Inc.)	

each price is an essential element for time-series forecasting. This hypothesis is confirmed by the fact that the corresponding financial instrument becomes more suitable for forecasting when considering a variable that links two successive days.

Here, GAN is used to generate synthetic data based on the input such that the network understands the links existing between data. This network can optimally identify the best variables (features) for forecasting by observing the improvements/reduction in the predictive score and by replacing different features in the datasets. Therefore, GAN can be used to validate financial instruments and variables based on which predictions can be subsequently conducted using any tool for obtaining the optimal result. In particular, many tools used to make predictions originate from econometrics. Thus, this type of network can be used to screen financial instruments such that the accuracy of the classical econometric tools is improved.

4.1 Dataset and experimental setup

Herein, empirical analyses of stocks and cryptocurrencies were conducted. However, first, we can consider different instruments because they are both time series and, therefore, have the same characteristics. The instruments used are four stocks and three cryptos listed in Table 1. All the stocks¹ are from NASDAQ, while the cryptocurrencies² are related to USD. The related datasets were divided into two categories, i.e., classic variables (features) obtained from any site that tracks negotiations and a category in which the yield feature was added. Each dataset for stocks is about 500 days, while those for cryptocurrencies are from 700 days (due to the 24 h opening). The classical features used in the financial sector are Open (**O**), High (**H**), Low (**L**), Close (**C**), and Volume (**V**), which give the name to the corresponding datasets. For simplicity, we indicate with *Close* the *Adjusted Close* feature, which represents the price recorded at closing after some updates made by the Stock Exchange. Prices were considered based

on a daily time frame from 12/20/2017 to 12/31/2019. Table 2, shows an extract of how the dataset is composed. Here, the Yield feature (indicate with (**Y**)) was determined as $Y = \ln \frac{P_t}{P_{t-1}}$, where P_t represents the current stock price and P_{t-1} is the stock price of the previous day. Table 3 shows how the dataset changes with the addition of the new feature (the Close feature in cryptocurrencies is called Last). An immediately striking peculiarity is that in cryptocurrency datasets, the opening and closing prices are the same in most cases. This is linked to the 24 h opening described above since the registration of the closing and opening price of the following day take place with a few moments of difference. Previous datasets were again divided into several types to test the predictive ability of prices. In particular, the macro division concerns the use or not of the Yield feature and some combinations between the previous OHLCVs.

- Dataset without yield:
 - **OHLCV** dataset, with five features: open, high, low, close (last in the case of cryptocurrencies), and volume;
 - **OC** dataset, with only two features: open and close (last).
- Dataset with yield:
 - **OHLCVY** dataset, with six features (yield was added to the previous **OHLCV**).
 - **OCY** dataset, with three features (yield was added to the previous **OC**).
 - **CY** dataset, with only two features: close/last and yield.

In the “with yield” case, to test the predictive power of the Close feature (the price that is most considered in the financial field), we decided to combine it alone with Yield (**CY** dataset), which represents a summary of the trend between the previous day and the current one.

As previously defined, the TimeGAN we will use comprises 4 component networks, of which 2 are encoders and 2 decoders. These encoders and decoders, in practice, are based on Recurrent Neural Networks (RNNs) with LSTM cell. Figure 1 represents the structure of the TimeGAN instantiated with RNNs and their connections are visible. In this case, the \mathcal{L} functions represent the losses; $g_{S,\mathcal{X}}$, $r_{S,\mathcal{X}}$, $d_{S,\mathcal{X}}$ and $e_{S,\mathcal{X}}$ represent the functions that operate for each type of network (e.g., g for generator); u_t and $h_{S,t}$ the hidden state sequences; while \tilde{s} , s , \tilde{x}_t , x_t , \tilde{y}_S , \tilde{y}_t , z_S , z_t are the random vectors extracted from a certain distribution. It is precisely the various g , r , d and e functions implemented with particular network architectures.

¹ Source: www.finance.yahoo.com.

² Source: www.investing.com.

Table 2 Extract of the Apple price dataset without Yield

Date	Open	High	Low	Close	Volume
12/20/2017	174.89	175.42	173.25	168.82	23475600
12/21/2017	174.17	176.02	174.01	169.46	20949900
12/22/2017	174.69	175.42	174.50	169.46	16349400
12/26/2017	170.08	171.47	169.68	165.16	33185500
12/27/2017	170.10	170.78	169.71	165.19	21498200
⋮	⋮	⋮	⋮	⋮	⋮

Regarding the choice of the hyperparameters, the first choice was the *module* (cell), for which we opted for the LSTM, given its diffusion in the financial field and the excellent results obtained on data of this type. On the other hand, the hyperparameters contained in Table 4(a) were selected by choosing the configuration of Yoon et al. [55], as they also analyzed the financial case and proposed this choice as the best result. Only for *seq_len* (sequence length of time-series data) we have operated as choice 24, this to make it represent approximately one month of trading. Otherwise, the number of neurons in the RNN (the so-called parameter *hidden_dim*) was chosen variable based on the dataset type. To make a choice, we made, with each dataset, 10-fold cross-validation by using the Grid-SearchCV method (as proposed by Guarino et al. [58]). The number of neurons size was tested with values in the range between 2 and 40 (for even number), and the best results are those shown in Table 4(b) (as the values that minimize the KL divergence). Figure 2 summarizes how it works, highlighting a concatenation of LSTM cells whose input X_t is represented by the tensor constituted by the different datasets. Training and testing were performed using Google Colab.

5 Results

At this point we can compare the behavior of the network in the different types of subdivisions created in the dataset.

5.1 Case 1: Dataset without yield

Here, we compare different datasets from a graphical perspective. Then, based on the scores, we compare the capabilities of different time series in terms of the difference with respect to the amount of information. In this comparison, we compared the OHLCV datasets with the OC datasets for all financial instruments beginning from the dataset without yield. The graphical analysis was performed by comparing the t-SNE plots, and score-based

comparison was performed by comparing the KL divergence, discriminative score, and predictive score.

The first t-SNE analysis considered the OHLCV dataset (Fig. 3). This analysis indicates the potential of TimeGAN relative to data generation. In the cases shown in Fig. 3b, f (both cryptocurrencies), there is very precise adherence of the synthetic data to the original real data. Obviously, with this type of dataset, greater features are combined, allowing the network to improve the forecast. Graphical analysis allows us to observe how synthetic data extend over a greater surface than the original data in the case of cryptocurrencies Fig. 3b, d, f compared with the stocks (where they are much more concentrated).

The second t-SNE analysis was based on the OC dataset, as shown in Fig. 4.

In this case, at first glance, synthetic data are observed to be different from original data. However, careful analysis indicates that in Fig. 4a, c, f (all cryptocurrencies), synthetic data are similar to the original data or imitate (to a limited extent) their distribution. In case of stocks, the distribution of synthetic data is dispersive and not entirely consistent with the distribution of original data.

Here, the hypothesis is that the prices of cryptocurrencies contain a greater amount of information and have greater discriminative and predictive power. This hypothesis is supported based on a graphical analysis. However, to eliminate doubt, we introduce the results of analysis based on discriminative and predictive scores, where D_s denotes the discriminative score, P_s denotes the predictive score, and D_{KL} denotes the KL divergence, as shown in Table 5. The most important score is the predictive score (corresponding to MAE) because it specifies the ability to use the input data for making predictions, whereas the discriminative score specifies the ability of data to deceive the GAN discriminator.

By analyzing the values, it can be noticed how the cryptocurrency scores were the lowest and therefore the most significant, especially in the case of the OC dataset.

We also assume that in markets subject to opening/closing time, investors are struck by “euphoria” by being trapped in a bottleneck and going to generate a price that may not be representative of the real trend, especially close to the closing minutes. In contrast, such situations may not exist in 24/7 markets because such markets are always open.

5.1.1 Outliers

We observe a particular situation when analyzing the stocks of Tesla Inc. (TSLA) listed on NASDAQ. Despite being listed on a market subject to timetables, this analysis using TimeGAN (both graphically and based on scores) resulted in a price type containing plenty of information

Table 3 Extract of the Bitcoin price dataset with Yield

Date	Open	High	Low	Last	Volume	Yield
12/20/2017	17521.70	17813.60	15642.70	16462.00	227680	− 0.06239
12/21/2017	16461.10	17301.80	14953.00	15631.10	163740	− 0.05173
12/22/2017	15632.10	15823.70	10875.70	13665.00	466980	− 0.13449
12/23/2017	13665.00	15492.20	13356.10	14396.50	170170	0.052147
12/24/2017	14396.60	14413.70	12166.50	13790.00	182420	− 0.04304
⋮	⋮	⋮	⋮	⋮	⋮	⋮

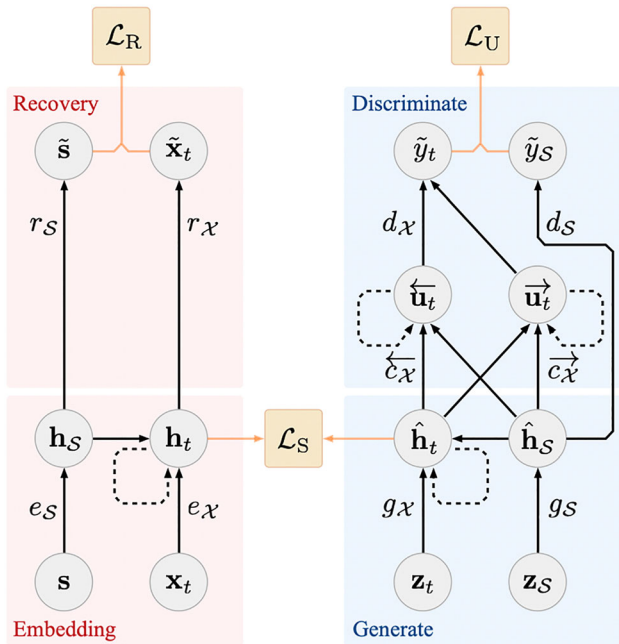


Fig. 1 TimeGAN [55] structure instantiated with RNNs

such that it achieved almost better results than cryptocurrencies (Fig. 5).

In the second dataset, the synthetic data reproduced the distribution of the original data very well. The results of the score analysis are presented in Table 6.

In this case, especially in the OC dataset, the price range indicated very low discriminative and predictive scores, even lower than that associated with cryptocurrencies.

Because of this “outlier”, we deduce that some financial instruments listed on stock exchanges are subject to timetables, which can completely absorb information despite the above limitation. This situation could be linked to, for example, the hypothesis that negative events never occurred outside the opening hours of the stock exchange or (in a less realistic but still possible hypothesis) that no external situations occurred in the considered time range that could influence the price. In these cases, the use of this type of GAN can be a “form of control” on prices, especially when they are to be used for forecasting. In addition, there may be hidden elements that affect the price. However, we assume that a financial instrument whose price has a “large amount of information” could result in improved prediction power when compared with a financial instrument with less information.

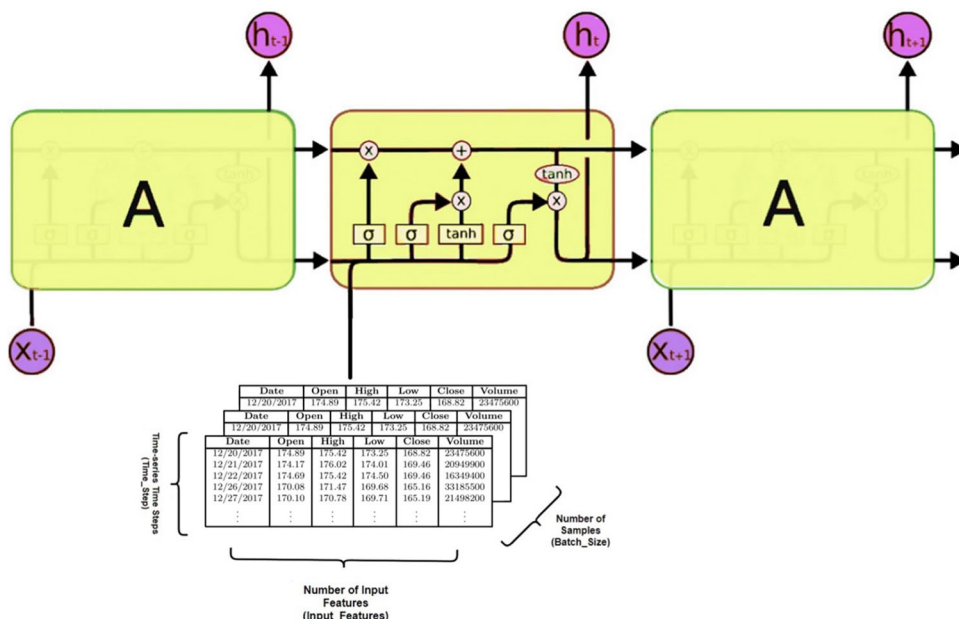
5.2 Case 2: Dataset with yield

Here, as in the previous case, we compare the financial instruments from a graphical perspective relative to the discriminative and predictive scores in reference to the datasets with the yield feature. We included the yield feature and verified its usefulness with respect to the amount of information. The hypothesis we seek to validate is that yield can link the information of the current day with the previous day, eliminating (or at least attenuating) the information gap created in markets subject to closing/opening times, e.g., stocks. In addition, under this new condition, the Tesla stock is no longer regarded as an outlier (the proof is presented later).

Table 4 Parameter setting

Parameters	Choice	Dataset	N. neurons
(a) Hyperparameter configuration list		(b) Choice of <i>hidden_dim</i> for dataset	
module	LSTM	OHLCV	20
seq-len	24	OC	8
iterations	30,000	OHLCVY	24
batch_size	128	OCY	12
num_layer	3	CY	8

Fig. 2 LSTM cells concatenated in a RNN [59, 60] with an example of the input data



The first t-SNE analysis considered the OHLCVY dataset, as shown in Fig. 6. From the graphical perspective, it can be observed how the situation changed compared to the previous datasets (without yield). In case of stocks (Fig. 6a, c, e, g), synthetic data adhere nearly perfectly to the original real data and cover a more extensive surface on them unlike cryptocurrencies, which extend over more “uncovered” areas.

The second t-SNE analysis was based on the OCY dataset, as shown in Fig. 7. In this case, despite the small differences between the two types of financial instruments, the data generated by stocks overlap better with the original data when compared with those generated by cryptocurrencies even though the covered areas are limited in both cases.

Finally, the third t-SNE analysis was conducted based on the CY dataset, as shown in Fig. 8. In the latter case, the best performances were obtained in the Bitcoin Cash Fig. 8f and Google Fig. 8e cases, where the ability of cryptocurrencies to be more predictive was eliminated. The Tesla stock Fig. 8g presented original data that can be classified as outliers, making t-SNE less readable. However, it was possible to observe how the synthetic data can cover the rest of the originals very well.

In support of the graphical analyses, the scores shown in Table 7 demonstrate how the yield feature improved the information potential of stocks.

On an average, the performance allowed GAN to achieved improved predictive score for stocks and discriminative score (especially in the OCY and CY datasets). However, this feature, which was also associated with the dataset of cryptocurrencies, resulted in a worse

performance compared to the datasets in which it was not present, confirming that TimeGAN is an excellent prediction tool. The Tesla stock was no longer an outlier because it obtained very similar scores compared to the remaining stocks. The yield feature, which is commonly used in financial analysis, made it possible to eliminate the information gap between cryptocurrencies and stocks.

6 Conclusion

In this paper, we demonstrated that TimeGAN can identify which financial instruments have a time series of prices containing abundant information. First, the prices of cryptocurrencies were observed to have much higher discriminatory and predictive powers than stocks, especially in the dataset comprising only the opening and closing prices. In addition, in case of the complete OHLCV dataset, prices with high discriminative power (combined with the remaining features) made it possible to significantly improve the adherence of the synthetic data with original data. From this analysis, we observed that some stocks have the same discriminative and predictive power as cryptocurrencies. Thus, because the time-series forecasting is primarily performed on stocks, this neural network can be used to screen the optimal titles, which combined with different features, improve the forecasting procedure. Second, by adding the yield feature, the previous situation can be modified to obtain datasets with stock prices as predictive as cryptocurrencies (or more in some cases). In future, we plan to investigate other features based on which

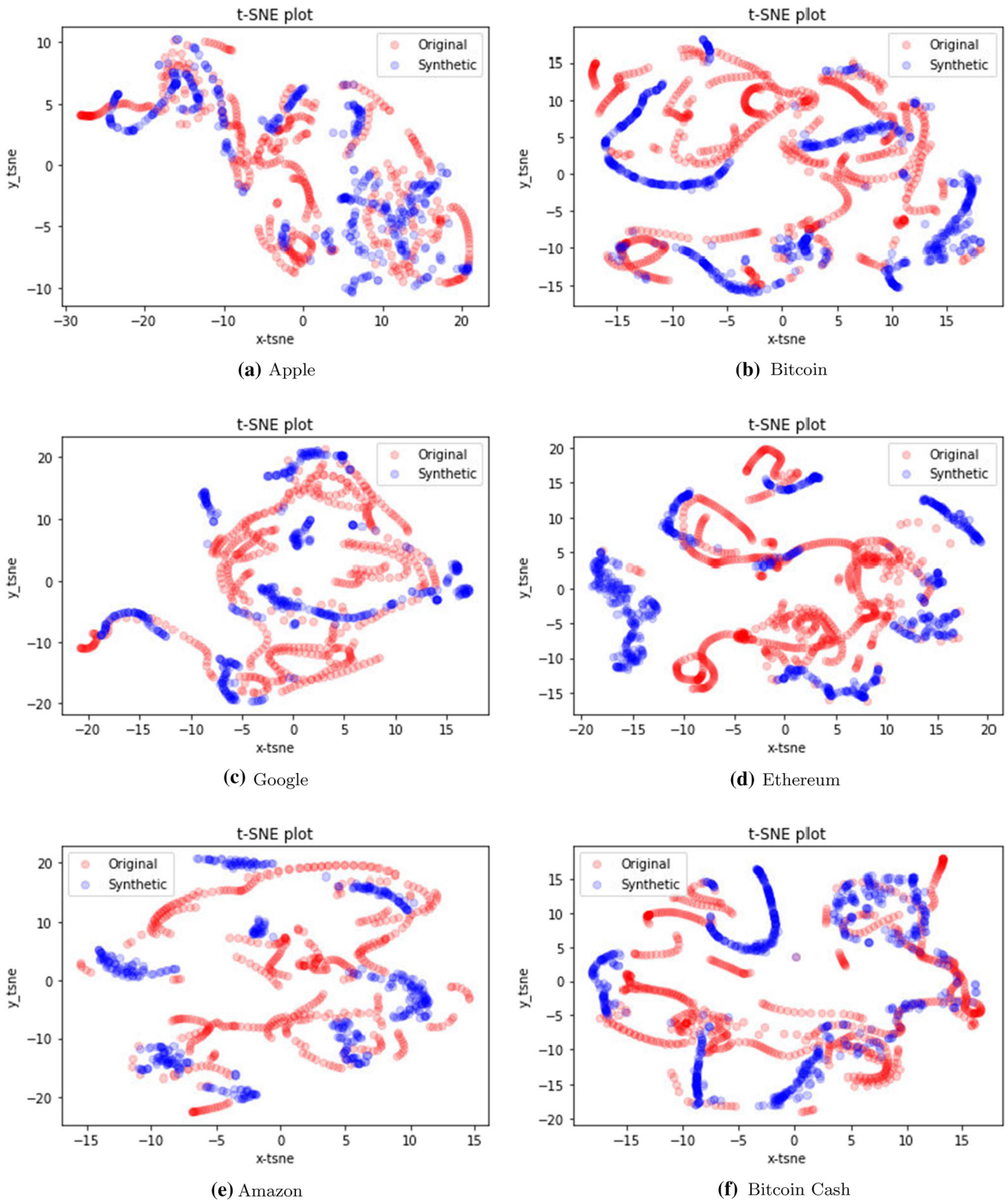


Fig. 3 t-SNE analysis of synthetic data (blue) and original data (red) (OHLCV dataset) (color figure online)

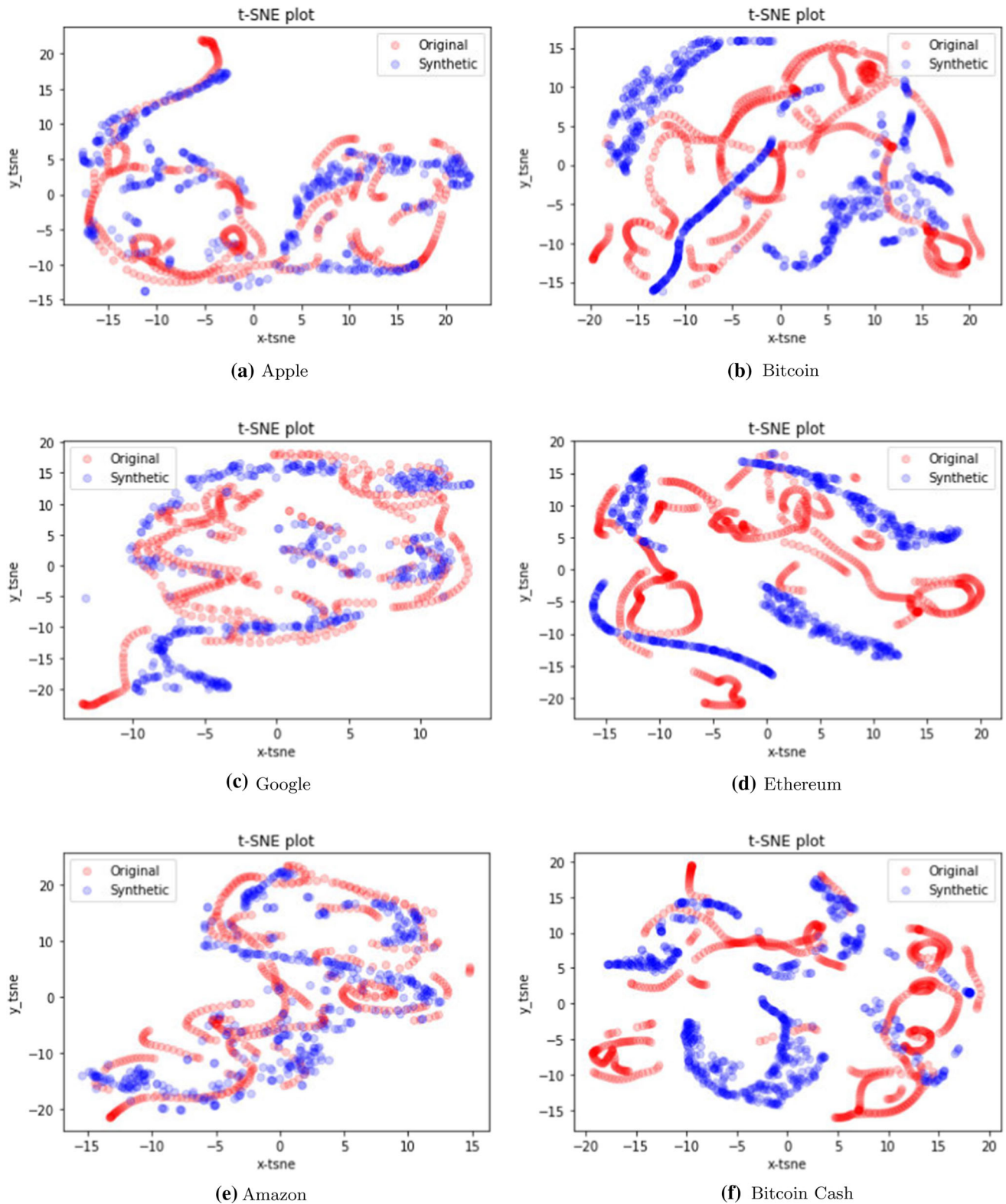


Fig. 4 t-SNE analysis of synthetic data (blue) and original data (red) (OC dataset) (color figure online)

the information capacity of various financial instruments may be improved to mitigate errors in forecasts.

Appendix: LSTM and CNN architectures

A neural network is a parallel computational model containing artificial neurons. Each network comprises a series of neurons [61] with a set of inputs and a corresponding output signal. The neuron model was modified by Rosenblatt [62], who defined *perceptron* as an entity with input and output layers based on error minimization. The study of associative memories and development of the *back-propagation* algorithm by Rumelhart et al. [63] have paved the way for the application of feedforward networks, drawing attention to recurrent networks. The fundamental unit of the neural network, i.e., the neuron, involves three fundamental elements: connections (each characterized by a weight), an adder that produces a linear combination of the inputs, and an activation function that limits the

amplitude of the output. We can describe a neuron as follows:

$$\hat{y} = g\left(w_0 + \sum_{i=1}^m x_i w_i\right), \tag{4}$$

where \hat{y} is the output, g is the activation function, x_i represent the inputs, w_0 represents the bias, and w_i represents weights. This can be expressed in matrix form as follows:

$$\hat{y} = g(w_0 + \mathbf{X}^T \mathbf{W}), \tag{5}$$

where \mathbf{X} and \mathbf{W} are the vectors of inputs and weights, respectively. The most common activation functions are the *sigmoid*, *hyperbolic tangent*, and *rectified linear unit* (ReLU) functions. Neural networks are characterized by a *learning algorithm*, i.e., a set of well-defined rules that can be used to solve a learning problem, which allows us to adapt the free parameters of the network. The learning algorithms can be of three types.

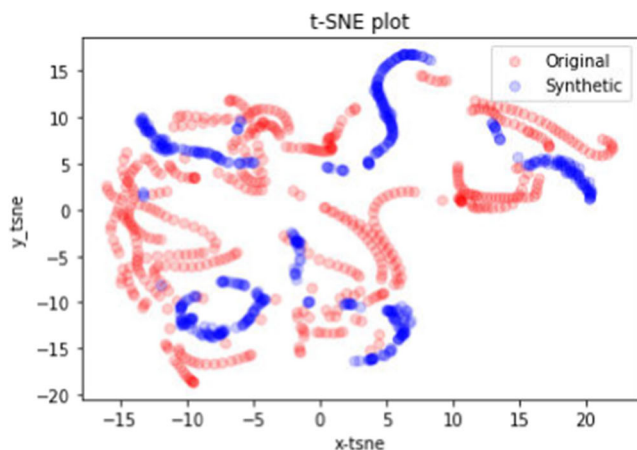
- *Supervised learning*: The network learns to infer the relation that binds the input values with the relative output values.
- *Unsupervised learning*: The network only has a set of input data and learns mappings autonomously.

Table 5 Scores of both datasets (the lower the value, the better the result)

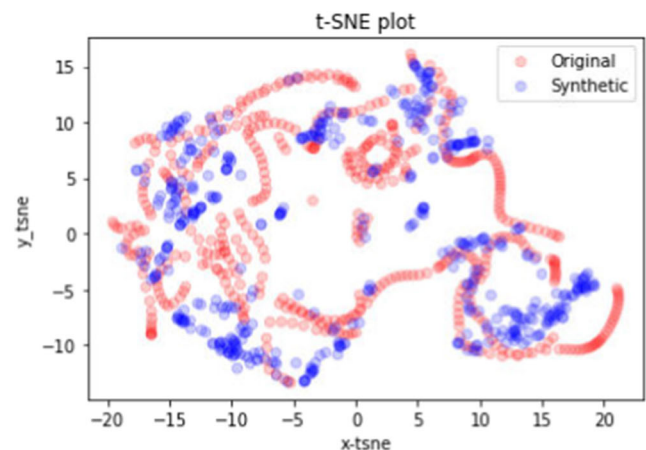
	OHLCV			OC		
	D_{KL}	D_S	P_S	D_{KL}	D_S	P_S
AAPL	54.3214	0.1133	0.1084	53.6518	0.0347	0.0354
GOOG	53.9618	0.1449	0.0722	55.6298	0.0378	0.0431
AMZN	52.2606	0.0755	0.1079	54.9458	0.0316	0.0254
BTC	54.7267	0.0868	0.0672	55.6924	0.0674	0.0109
BCH	53.6472	0.0569	0.0273	53.5539	0.0528	0.0136
ETH	52.9701	0.2458	0.0443	52.0232	0.0382	0.0129

Table 6 Scores of TSLA in both datasets

	OHLCV			OC		
	D_{KL}	D_S	P_S	D_{KL}	D_S	P_S
TSLA	54.3992	0.2163	0.1071	55.6569	0.0082	0.0182



(a) OHLCV dataset



(b) OC dataset

Fig. 5 t-SNE analysis of the TSLA stock on different datasets

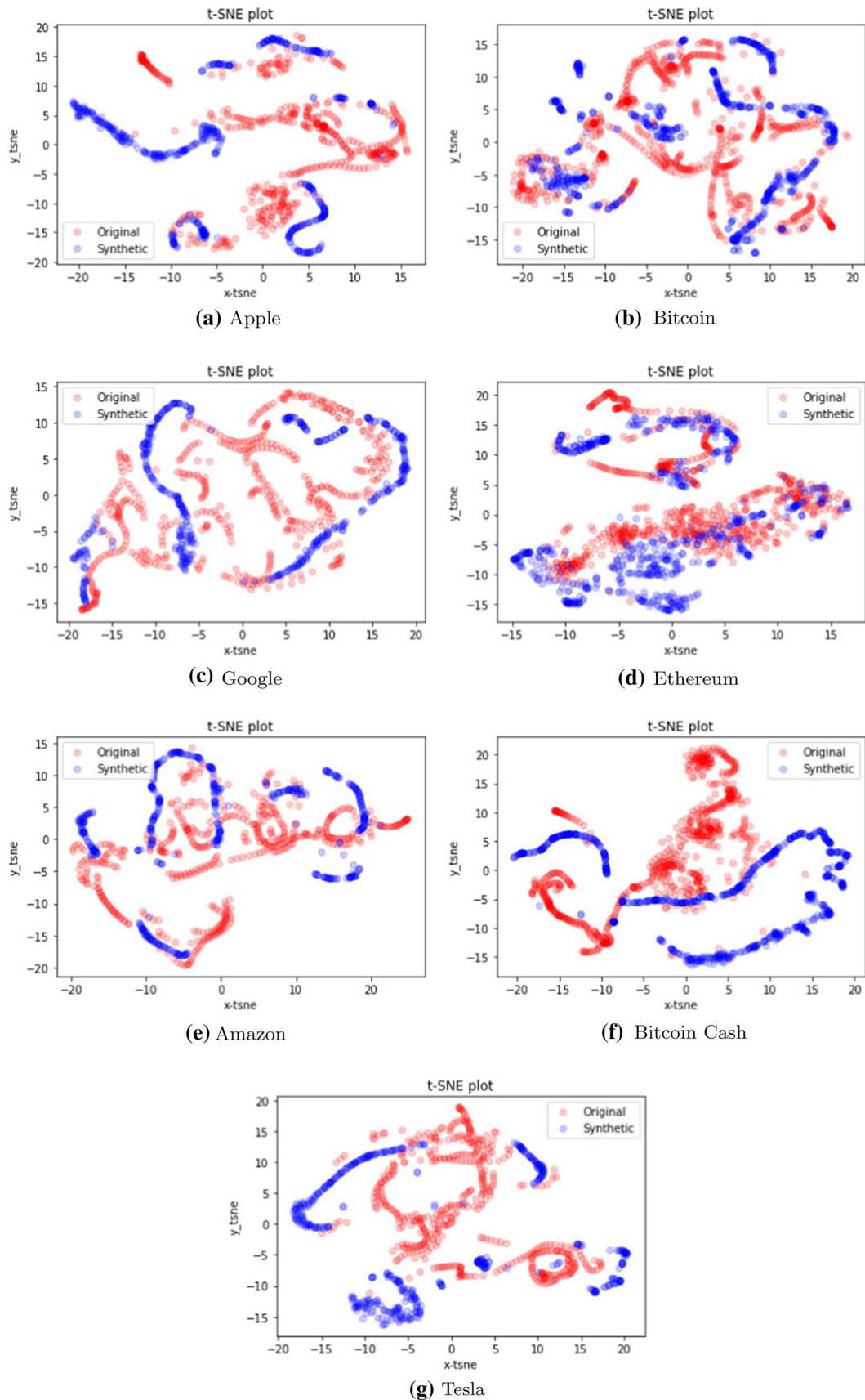


Fig. 6 t-SNE analysis of synthetic data (blue) and original data (red) (**OHLCVY dataset**) (color figure online)

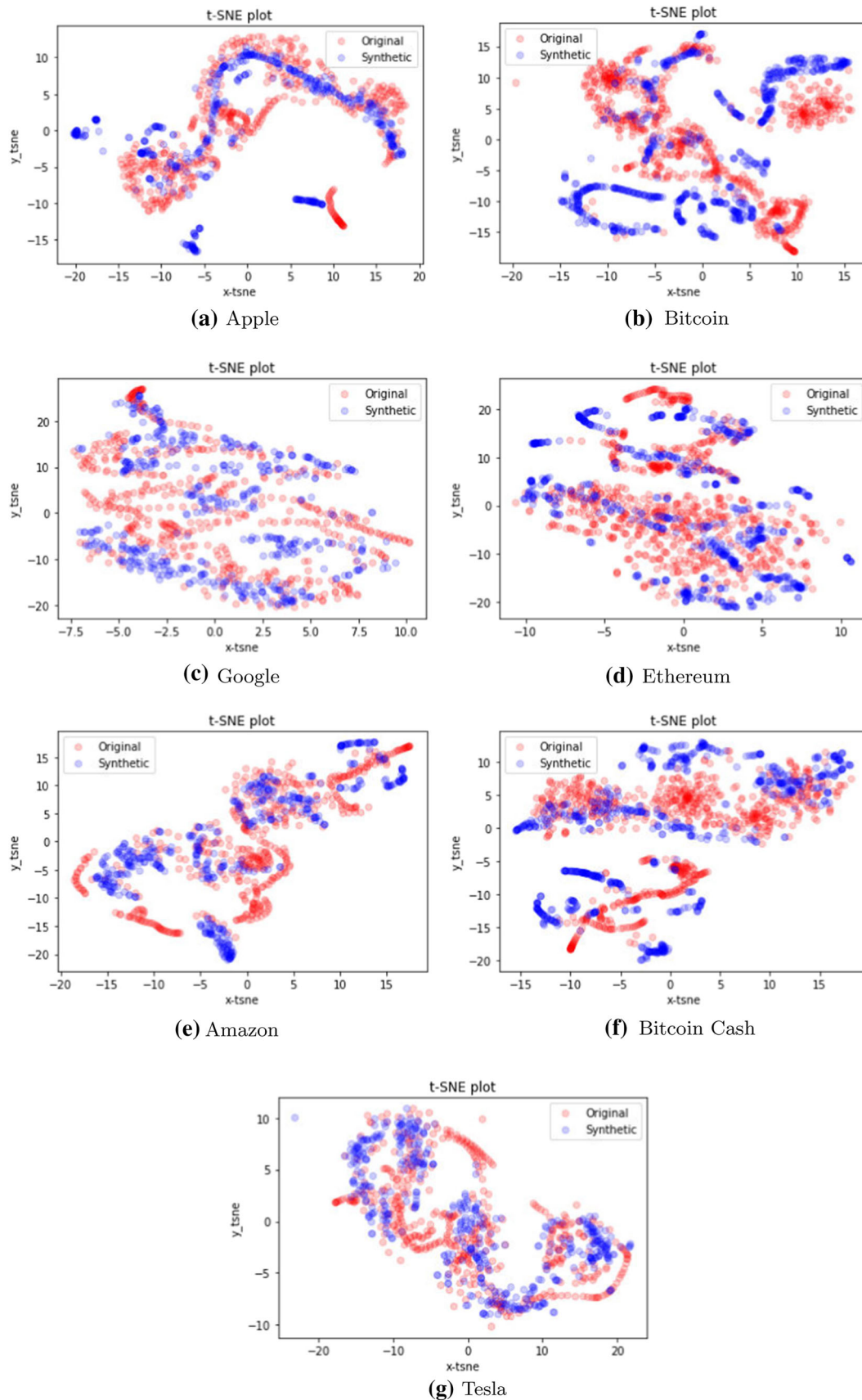


Fig. 7 t-SNE analysis of synthetic data (blue) and original data (red) (OCY dataset) (color figure online)

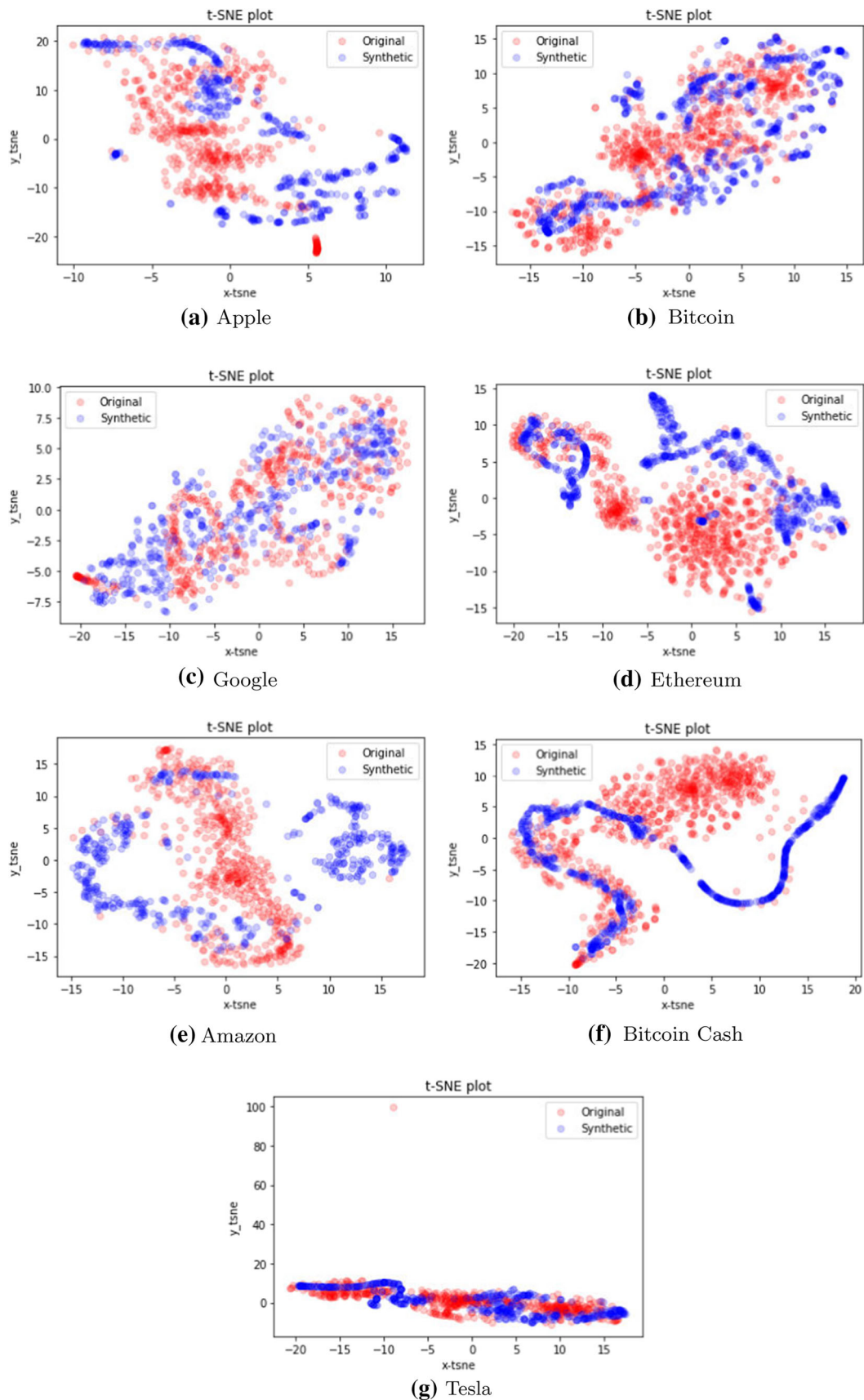


Fig. 8 t-SNE analysis of synthetic data (blue) and original data (red) (CY dataset) (color figure online)

Table 7 Scores of both datasets (lower scores are better)

	OHLCVY			OCY			CY		
	D_{KL}	D_S	P_S	D_{KL}	D_S	P_S	D_{KL}	D_S	P_S
AAPL	52.6359	0.3316	0.0815	55.8588	0.0408	0.0735	60.1909	0.0337	0.0731
GOOG	55.2421	0.1714	0.0686	58.3399	0.0357	0.0656	63.4955	0.0204	0.0654
AMZN	54.9933	0.2194	0.0866	55.6771	0.0337	0.0791	63.0692	0.1337	0.1014
TSLA	54.1207	0.2398	0.0825	59.2129	0.0184	0.0771	63.3792	0.0602	0.0771
BTC	55.4594	0.1764	0.0908	59.0598	0.0701	0.0842	63.4355	0.0161	0.0791
BCH	53.8986	0.3458	0.0839	58.2165	0.0326	0.0529	58.0093	0.0576	0.0594
ETH	58.7887	0.1979	0.0927	60.4058	0.0326	0.0901	63.4868	0.0911	0.0908

- *Semisupervised learning*: This is a mixed approach that combines a small number of labeled values and a large number of unlabeled values.

A neural network includes a set of inputs, several hidden layers, and a set of outputs. Each neuron has a nonlinear activation function and is equipped with high connectivity. In this case, we focus on *Multilayer Perceptron* (MLP) networks, in which learning involves the minimization of loss function $J(W)$ using a backpropagation algorithm [63].

Convolutional neural network

A *Convolutional Neural Network* (CNN) is a feedforward network introduced by LeCun et al. [64], which was initially designed for image processing. However, recently, CNNs have been applied to financial time series [65]. Based on convolution, a CNN comprises a hierarchy of levels in which the intermediate levels use local connections and the latter layers are fully connected and operate as classifiers. The key feature of this model is the presence of the *convolution* and *pooling* levels, which aggregate the information associated with the input volume to generate a feature map of small dimensionality to ensure invariance relative to transformations and avoid loss of information. In a financial case, some recent applications are those of Mittelman [66], who used an undecimated fully convolutional network to model a time series, Binkowski et al. [67], who based their idea on an autoregressive-type weighting system with a CNN, Tsantekidis et al. [68], who used the time series derived from an order book, and Livieris et al. [69] who propose a convolutional layer for extracting knowledge and a LSTM for identify dependencies in gold price time series.

The key element of such networks is *convolution* (in the discrete case). Here, an operator is defined as follows [70]:

$$(f * g)(i) = \sum_{j=-\infty}^{\infty} f(j)g(i - j), \tag{6}$$

where f and g are two functions defined on \mathbb{Z} . For example, convolution is used in the layer following the input layer

with a set of filters to create a feature map. As defined by Borovykh et al. [70], the feature map from the first layer is obtained by bringing together each filter w_h^i for $h = 1, \dots, M_1$ (where M_1 is the set of filters applied on each input channel) with the input.

$$a^1(i, h) = \sum_{j=-\infty}^{\infty} w_h^1(h)x(i - j), \tag{7}$$

where $w_h^1 \in \mathbb{R}^{1 \times k \times 1}$ and $a^1 \in \mathbb{R}^{1 \times N - k + 1 \times M_1}$ (in this case, a one-dimensional input of size N without zero padding). This process is repeated for each subsequent layer. The output of the network after L convolutional layers is the matrix f^L . The size of this matrix f^L is dependent on the filter size and the number of filters used.

Long–short term memory

In financial time-series forecasting, the most common *Recurrent Neural Network* (RNN) is the *Long Short-Term Memory* (LSTM), which was introduced by Hochreiter and Schmidhuber [71] in 1997. A characteristic of this network is that at each step, the network receives both the input and the output of the previous level. Thus, decisions can be made based on history. However, because distant memory tends to fade in base cells, LSTM prevents this through its long-term memory. Each LSTM cell controls the flow of information, i.e., irrelevant things are forgotten, the cell-state values are updated, and an output gate is used to output parts of the cell state. As defined by Sagheer and Kotb [72], the hidden state S_t , which is based on the input X_t and the hidden state from the previous time step S_{t-1} , can be described as follows:

$$\begin{aligned} f_t &= \sigma(X_t U^f + S_{t-1} W^f + b_f), \\ i_t &= \sigma(X_t U^i + S_{t-1} W^i + b_i), \\ \tilde{C}_t &= \tanh(X_t U^c + S_{t-1} W^c + b_c), \\ C_t &= C_{t-1} \odot f_t + i_t \odot \tilde{C}_t, \\ o_t &= \sigma(X_t U^o + S_{t-1} W^o + b_o), \\ S_t &= o_t \odot \tanh(C_t). \end{aligned} \tag{8}$$

Here, \odot represents the Hadamard product, σ is the logistic sigmoid activation function, f is the forget gate, i is the identify gate, o is the output gate, and C is the cell state. In addition, U represents the input weight matrix, W represents the recurrent weight matrix, and b represents the bias.

Funding Open access funding provided by Università di Foggia within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors certify that there is no actual or potential conflict of interest in relation to this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Hayek F (1945) The use of knowledge in society. *Am Econ Rev* 35:519–530
- Loughran T, McDonald B (2014) Measuring readability in financial disclosures. *J Finance* 69(4):1643–1671. <https://doi.org/10.1111/jofi.12162>
- Beattie V, McInnes B, Fearnley S (2004) A methodology for analysing and evaluating narratives in annual reports: a comprehensive descriptive profile and metrics for disclosure quality attributes. *Account For* 28(3):205–236. <https://doi.org/10.1016/j.accfor.2004.07.001>
- Healy PM, Palepu KG (2001) Information asymmetry, corporate disclosure, and the capital markets: a review of the empirical disclosure literature. *J Account Econ* 31(1):405–440. [https://doi.org/10.1016/S0165-4101\(01\)00018-0](https://doi.org/10.1016/S0165-4101(01)00018-0)
- Graham JR, Harvey CR, Rajgopal S (2005) The economic implications of corporate financial reporting. *J Account Econ* 40(1):3–73. <https://doi.org/10.1016/j.jacceco.2005.01.002>
- Brown S, Hillegeist S (2007) How disclosure quality affects the level of information asymmetry. *Rev Account Stud* 12:443–477. <https://doi.org/10.1007/s11142-007-9032-5>
- Li F (2008) Annual report readability, current earnings, and earnings persistence. *J Account Econ* 45(2):221–247. <https://doi.org/10.1016/j.jacceco.2008.02.003>
- Bloomfield R (2008) Discussion of “annual report readability, current earnings, and earnings persistence”. *J Account Econ* 45(2):248–252. <https://doi.org/10.1016/j.jacceco.2008.04.002>
- Chung D, Hrazdil K, Suwanyangyuan N (2016) Disclosure quantity and the efficiency of price discovery evidence from the Toronto stock exchange. *Rev Account Finance* 15:122–143. <https://doi.org/10.1108/RAF-06-2015-0081>
- Fisher MG, Hughes M (2016) Natural language processing in accounting, auditing and finance: a synthesis of the literature with a roadmap for future research. *Intell Syst Account Finance Manag* 23(3):157–214. <https://doi.org/10.1002/isaf.1386>
- Chen BM, Shevlin T (2015) A new measure of disclosure quality: the level of disaggregation of accounting data in annual reports. *J Account Res* 53(5):1017–1054. <https://doi.org/10.1111/1475-679X.12094>
- Chung DY, Hrazdil K, Novak J, Suwanyangyuan N (2019) Does the large amount of information in corporate disclosures hinder or enhance price discovery in the capital market? *J Contemp Account Econ* 15(1):36–52. <https://doi.org/10.1016/j.jcae.2018.12.001>
- Tang Z, Almeida C, Fishwick PA (1991) Time series forecasting using neural networks vs. Box–Jenkins methodology. *Simulation* 57(5):303–310. <https://doi.org/10.1177/003754979105700508>
- Gorr WL (1994) Editorial: research prospective on neural network forecasting. *Int J Forecast* 10(1):1–4. [https://doi.org/10.1016/0169-2070\(94\)90044-2](https://doi.org/10.1016/0169-2070(94)90044-2)
- Hamid OH (2014) The role of temporal statistics in the transfer of experience in context-dependent reinforcement learning. In: 2014 14th international conference on hybrid intelligent systems, pp 123–128. <https://doi.org/10.1109/HIS.2014.7086184>
- Bachelier L (1900) Théorie de la spéculation. *Annales scientifiques de l'École Normale Supérieure 3e s'erie* 17:21–86. <https://doi.org/10.24033/asens.476>
- Kendall MG, Hill AB (1953) The analysis of economic time series. Part I: prices. *J R Stat Soc Ser A (Gen)* 116(1):11–34. <https://doi.org/10.2307/2980947>
- Cootner PH (1964) The random character of stock market prices. MIT Press, Cambridge, p 510
- Fama E (1970) Efficient capital markets: a review of theory and empirical work. *J Finance* 25(2):383–417. <https://doi.org/10.2307/2325486>
- LeRoy SF (1989) Efficient capital markets and martingales. *J Econ Liter* 27(4):1583–1621
- Taylor SJ (1980) Conjectured models for trends in financial prices, tests and forecast. *J R Stat Soc Ser A (Gen)* 143(3):338–362. <https://doi.org/10.2307/2982133>
- Box GEP, Jenkins GM (2015) Time series analysis: forecasting and control. Holden-Day, New York
- Hamilton JD (1994) Time series analysis. Princeton University Press, Princeton
- Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50:159–175. [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0)
- Foster WR, Collopy F, Ungar LH (1992) Neural network forecasting of short, noisy time series. *Comput Chem Eng* 16(4):293–297. [https://doi.org/10.1016/0098-1354\(92\)80049-F](https://doi.org/10.1016/0098-1354(92)80049-F)
- Refenes AN, Azema-Barac M, Karoussos SA (1992) Currency exchange rate forecasting by error backpropagation. In: Proceedings of the twenty-fifth Hawaii international conference on system sciences, vol iv, pp 504–5154. <https://doi.org/10.1109/HICSS.1992.183441>
- Sharda R, Patil RB (1992) Connectionist approach to time series prediction: an empirical test. *J Intell Manuf* 3:317–323. <https://doi.org/10.1007/BF01577272>
- Andrawis RR, Atiya AF, El-Shishiny H (2011) Combination of long term and short term forecasts, with application to tourism demand forecasting. *Int J Forecast* 27(3):870–886. <https://doi.org/10.1016/j.ijforecast.2010.05.019>
- Adeodato PJL, Arnaud AL, Vasconcelos GC, Cunha RCLV, Monteiro DSMP (2011) MLP ensembles improve long term prediction accuracy over single networks. *Int J Forecast* 27(3):661–671. <https://doi.org/10.1016/j.ijforecast.2009.05.029>

30. Wichard JD (2011) Forecasting the NN5 time series with hybrid models. *Int J Forecast* 27(3):700–707. <https://doi.org/10.1016/j.ijforecast.2010.02.011>
31. Vapnik V (1998) *Statistical learning theory*. Wiley, Hoboken, p 736
32. Adhikari R, Agrawal RK (2013) *An introductory study in time series modeling and forecasting*. LAP LAMBERT Academic Publishing, Sunnyvale, p 76
33. Suykens JAK, Vandewalle J (2000) Recurrent least squares support vector machines. *IEEE Trans Circuits Syst I Fundam Theory Appl* 47(7):1109–1114. <https://doi.org/10.1109/81.855471>
34. Xiao C, Xia W, Jiang J (2020) Stock price forecast based on combined model of ARIMA-LS-SVM. *Neural Comput Appl* 32:5379–5388. <https://doi.org/10.1007/s00521-019-04698-5>
35. Kovalerchuk B, Vityaev E (2000) *Data mining in finance: advances in relational and hybrid methods*. The springer international series in engineering and computer science. Springer, New York, p 308
36. Zaccagnino R, Capo C, Guarino A (2021) Techno-regulation and intelligent safeguards. *Multimedia Tools Appl* 80:15803–15824. <https://doi.org/10.1007/s11042-020-10446-y>
37. Guarino A, Malandrino D, Zaccagnino R (2022) An automatic mechanism to provide privacy awareness and control over unwittingly dissemination of online private information. *Comput Netw* 202:108614. <https://doi.org/10.1016/j.comnet.2021.108614>
38. Li Y, Ma W (2010) Applications of artificial neural networks in financial economics: a survey. In: 2010 international symposium on computational intelligence and design. IEEE, pp 211–214. <https://doi.org/10.1109/ISCID.2010.70>
39. Mittelmayer M, Knolmayer GF (2006) Text mining systems for market response to news: Aasurvey. In: IADIS European conference data mining 2007 (part of MCCSIS 2007), pp 164–169. ISBN: 978-972-8924-40-9
40. Mitra L, Mitra G (2011) Applications of news analytics in finance: a review. *The handbook of news analytics in finance*, pp 1–39. ISBN: 9781118467411
41. Sezer OB, Gudelek MU, Ozbayoglu AM (2020) Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. *Appl Soft Comput*. <https://doi.org/10.1016/j.asoc.2020.106181>
42. Liu S, Zhang C, Ma J (2017) CNN-LSTM neural network model for quantitative strategy analysis in stock markets. In: Liu D, Xie S, Li Y, Zhao D, El-Alfy E-SM (eds) *Neural information processing*. Springer, Cham, pp 198–206
43. Zhang L, Aggarwal C, Qi G-J (2017) Stock price prediction via discovering multi-frequency trading patterns. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. KDD'17*, pp 2141–2149. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3097983.3098117>
44. Jin Z, Yang Y, Liu Y (2020) Stock closing price prediction based on sentiment analysis and LSTM. *Neural Comput Appl* 32:9713–9729. <https://doi.org/10.1007/s00521-019-04504-2>
45. Lu W, Zi J, Wang Q (2021) A CNN-BiLSTM-AM method for stock price prediction. *Neural Comput Appl* 33:4741–4753. <https://doi.org/10.1007/s00521-020-05532-z>
46. Kaneko T, Kameoka H, Hojo N, Ijima Y, Hiramatsu K, Kashino K (2017) Generative adversarial network-based postfilter for statistical parametric speech synthesis. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 4910–4914. <https://doi.org/10.1109/ICASSP.2017.7953090>
47. Sun Y, Ximing L, Cong P, Li L, Zhao Z (2018) Digital radiography image denoising using a generative adversarial network. *J Xray Sci Technol* 26(4):523–534. <https://doi.org/10.3233/XST-17356>
48. Lei K, Xie Y, Zhong S, Dai J, Yang M, Shen Y (2020) Generative adversarial fusion network for class imbalance credit scoring. *Neural Comput Appl* 32:8451–8462. <https://doi.org/10.1007/s00521-019-04335-1>
49. Papandreou G, Chen L-C, Murphy KP, Yuille AL (2015) Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: 2015 IEEE international conference on computer vision (ICCV), pp 1742–1750. <https://doi.org/10.1109/ICCV.2015.203>
50. Liu H, Tian Y, Wang Y, Pang L, Huang T (2016) Deep relative distance learning: tell the difference between similar vehicles. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 2167–2175. <https://doi.org/10.1109/CVPR.2016.238>
51. Zhang J, Zhong S, Wang T, Chao HC, Wang J (2020) Blockchain-based systems and applications: a survey. *J Internet Technol* 21(1):1–14
52. Qi E, Deng M (2019) R&D investment enhance the financial performance of company driven by big data computing and analysis. *Comput Syst Sci Eng* 34(4):237–248
53. Zhou H, Sun G, Fu S, Fan X, Jiang W (2020) A distributed approach of big data mining for financial fraud detection in a supply chain. *Comput Mater Continua* 64(2):1091–1105
54. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: *Proceedings of the 27th international conference on neural information processing systems - volume 2. NIPS'14*. MIT Press, Cambridge, MA, USA, pp 2672–2680
55. Yoon J, Jarrett D, van der Schaar M (2019) Time-series generative adversarial networks. In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R (eds) *Advances in neural information processing systems*, vol 32. Curran Associates, Inc., New York, pp 1–11
56. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *J Mach Learn Res* 9(86):2579–2605
57. Kullback S, Leibler RA (1951) On information and sufficiency. *Ann Math Stat* 22(1):79–86. <https://doi.org/10.1214/aoms/1177729694>
58. Guarino A, Lettieri N, Malandrino D, Zaccagnino R (2021) A machine learning-based approach to identify unlawful practices in online terms of service: analysis, implementation and evaluation. *Neural Comput Appl* 33:17569–17587. <https://doi.org/10.1007/s00521-021-06343-6>
59. Olah C (2015) Understanding LSTM networks. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
60. Malakar S, Goswami S, Ganguli B, Chakrabarti A, Roy SS, Boopathi K, Rangaraj AG (2021) Designing a long short-term network for short-term forecasting of global horizontal irradiance. *SN Appl Sci*. <https://doi.org/10.1007/s42452-021-04421-x>
61. McCullock WS, Pitts WH (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:99–115. <https://doi.org/10.1007/BF02478259>
62. Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386–404. <https://doi.org/10.1037/h0042519>
63. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representation by back-propagation errors. *Nature* 323:533–536. <https://doi.org/10.1038/323533a0>
64. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324. <https://doi.org/10.1109/5.726791>
65. Chen J-F, Chen W-L, Huang C-P, Huang S-H, Chen A-P (2016) Financial time-series data analysis using deep convolutional neural networks. In: 2016 7th international conference on cloud

- computing and big data (CCBD), pp 87–92. <https://doi.org/10.1109/CCBD.2016.027>
66. Mittelman R (2015) Time-series modeling with undecimated fully convolutional neural networks. [arXiv:1508.00317](https://arxiv.org/abs/1508.00317)
67. Binkowski M, Marti G, Donnat P (2017) Autoregressive convolutional neural networks for asynchronous time series. In ICML 2017 time series workshop, pp 1–16
68. Tsantekidis A, Passalis N, Tefas A, Kannianen J, Gabbouj M, Iosifidis A (2017) Forecasting stock prices from the limit order book using convolutional neural networks. In: 2017 IEEE 19th conference on business informatics (CBI), vol 01, pp 7–12. <https://doi.org/10.1109/CBI.2017.23>
69. Livieris IE, Pintelas E, Pintelas P (2020) A CNN-LSTM model for gold price time-series forecasting. *Neural Comput Appl* 32:17351–17360. <https://doi.org/10.1007/s00521-020-04867-x>
70. Borovykh A, Bohte S, Oosterlee CW (2018) Conditional time series forecasting with convolutional neural networks. [arXiv:1703.04691](https://arxiv.org/abs/1703.04691)
71. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
72. Sagheer A, Kotb M (2019) Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* 323:203–213. <https://doi.org/10.1016/j.neucom.2018.09.082>

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