Deep Learning and Time Series-to-Image Encoding for Financial Forecasting

Silvio Barra, Salvatore Mario Carta, Andrea Corriga, Alessandro Sebastian Podda, and Diego Reforgiato Recupero

Abstract—In the last decade, market financial forecasting has attracted high interests amongst the researchers in pattern recognition. Usually, the data used for analysing the market, and then gamble on its future trend, are provided as time series; this aspect, along with the high fluctuation of this kind of data, cuts out the use of very efficient classification tools, very popular in the state of the art, like the well known convolutional neural networks (CNNs) models such as Inception, ResNet, AlexNet, and so on. This forces the researchers to train new tools from scratch. Such operations could be very time consuming. This paper exploits an ensemble of CNNs, trained over Gramian angular fields (GAF) images, generated from time series related to the Standard & Poor's 500 index future; the aim is the prediction of the future trend of the U.S. market. A multi-resolution imaging approach is used to feed each CNN, enabling the analysis of different time intervals for a single observation. A simple trading system based on the ensemble forecaster is used to evaluate the quality of the proposed approach. Our method outperforms the buy-and-hold (B&H) strategy in a time frame where the latter provides excellent returns. Both quantitative and qualitative results are provided.

Index Terms—Convolutional neural networks (CNNs), ensemble of CNNs, financial forecasting, Gramian angular fields (GAF) imaging.

I. INTRODUCTION

SINCE the dawn of the financial market, people have been trying to build tools able to provide insights and information about the stock price variations in the near future, so to increase the possibilities to invest on the right company [1], future, etc. Since then, the market has become much bigger, and the available instruments for financial forecasting have reached an unprecedented efficiency. Nowadays, the research in this area is one of the most active amongst the pattern recognition related topics, and at the same time it is one of the most challenging. This is mainly due to the fact that stock prices are often influenced by factors which are quite hard predictable like political events, the behaviour of the other stock markets and, last but not least, the psychology of the in-

vestors [2]; these aspects tend to model the market as an entity which is dynamic, non-linear, non-parametric, and chaotic [3].

Notwithstanding the above components, the research performed in recent years in this field is growing both in terms of literature production and in tools generation [4], [5]; also, an increasing number of studies involves the use of learning approaches like machine learning classifiers and neural network models. Essentially, these techniques cover most of the research achieved in the field. The neural networks approaches applied to financial forecasting started approximately across the end of 80's and the beginning of 90's. In those years, one of the first artificial neural network has been proposed [1], with the aim of predicting the Tokyo stock exchange price indexes (TOPIX Index), by taking six metric vectors as input. In the same year, the authors in [6] built a recurrent neural network for stock price patterns' recognition, exploiting the triangle pattern as a clue to the trend of the future stock prices. In [7], a similar approach has exploited eleven market indicators for building and training a recurrent neural network for monthly transition of the stock price index [8]. Several approaches have also been proposed in the field, facing the financial series prediction topic by using the dendritic neuron model (DNM) technique [9], whose performances have been shown both on Asiatic [10] and U.S. market [11]. These techniques have evolved a lot in the recent 10 years, as well as the finance itself [12]. As a consequence, dozens of scientific papers have been published, proposing approaches which aim at predicting the stock prices by exploiting news data extracted from the most popular social networks for modelling the uncertainty which lies behind the fluctuation of the market [13]. Financial forecasting is indeed one of the research branches in which sentiment analysis found a quite breeding ground [14], [15].

Alongside the artificial neural networks (ANN), also the machine learning approaches had the possibility to show their efficiency through the years. In [16], the authors have compared the capabilities of the support vector machines [17] in market prediction related issues against those obtained by using the radial basis function (RBF) networks and back propagation (BP). In [18], a recent literature review is performed, which compares the modern machine learning approaches in financial forecasting field.

Interesting results have also been obtained when fusing together the above described techniques: as an example, in [19] the authors have fused ANN with decision trees (DT). The rationale behind this hybrid approach is that where ANNs are able to provide quite good performances in forecasting the

market trend, a DT model is stronger in generating potential rules which describe the forecasting decisions. Similarly, in [20], a two-stage fusion approach is proposed for predicting CNX Nifty and S&P Bombay Stock Exchange (BSE) Sensex from Indian stock markets. Specifically, the first stage uses a support vector regressor (SVR), whereas the second one exploits, in turn, ANN, random forest, and SVR. Ten indicators have been selected as input to the prediction models. In general, however, predicting the daily direction (positive or negative) of the market requires to solve a classification problem, whereas directly predicting the profit needs to address a regression problem. A comparison of different market price prediction approaches is shown in [21], where classification-based approaches usually provide better results, such as [22]–[24], rather than some regression-based competitors. As reported in [25], moreover, "there is no general consensus on best forecasting technique for price prediction", particularly since "price series is inherently a non-stationary series having non-constant mean and variance", making it not always advantageous to represent the problem through linear models such as regression.

Furthermore, the actual trend of the research seems to be oriented towards the analysis of the data in their raw forms, therefore as time series, without doing any dimensionality changing. Inspired by recent successes of supervised and unsupervised learning techniques in computer vision, and with the aim to change this trend, in this paper we propose a trading system based on the forecast of the daily direction of a market index by exploiting the discriminatory capabilities of the convolutional neural networks (CNN) when dealing with GAF images. Indeed, CNNs has shown very good accuracy results when applied to pattern recognition on image and video data [26], [27]. We encoded time series as images to allow machines to visually recognise, classify and learn structures and patterns. Reformulating features of time series as visual clues has raised much attention in computer science and physics [28].

Model training, evaluation and testing are executed on Standard & Poor's 500 (S&P500) index future. Time series of the future prices are processed in a twofold way: firstly, different intervals of time are considered, in order to analyse the same trend under different points of view; then, GAF images are built for each of the defined time intervals. Therefore, the main contributions of the proposed approach are as follows.

• The proposed system exploits the GAF imaging approach for encoding time series data as images;

• The composition of GAF images in a multi-resolution structure helps improving the market prediction results;

• The classification phase is carried out by organising in an ensemble a set of CNNs which have the same architecture, but each of them is initialized with a different kernel function for initialization. A majority voting-based policy is adopted;

• Comparisons both with state of the art baseline approaches (e.g., buy-and-hold (B&H) strategy) and with the results of an existing competitor method (Calvi *et al.* [29]) have been performed, showing that the proposed system is capable of obtaining a higher profit in the same investment period.

The remainder of this paper is organised as follows: Section II outlines the developed method, from the generation of the GAF images to the description of the ensemble of CNNs. Sections III and IV describe, respectively, the experiments' settings, along with the data preparation and the dataset description, and show and discuss the qualitative and quantitative results. Section V ends the paper with conclusions and future work where we are headed.

II. THE PROPOSED APPROACH

As pointed in Section I, most of the research in market prediction and financial forecasting is based on ANN or machine learning approaches. These models are commonly trained on time series data describing a market index in the past, with the goal of predicting its future trend. The aim of this work is to achieve market prediction over the S&P500 index, by using an ensemble of CNNs, with the training phase executed over GAF images (particularly, the GADF). This particular kind of imaging technique is detailed in Section II-B. The rightmost part of Fig. 1 shows how the proposed trading system works; firstly, the data of the original time series are aggregated according to 4 intervals of time; then, consecutive time frames of 20 observations are extracted from each time series, in order to generate the related set of GADF images. Twenty similar CNNs (whose architecture is shown in the leftmost part of Fig. 1) are trained over these images, and the threshold-driven ensemble approach takes place for deciding which action to perform the day after the observations, as described in what follows.

A. Trading Strategy

We design our system to simulate a classic intraday trading strategy. This strategy generally consists of buying or selling a specific financial instrument (in our case, the S&P500 index future), by making sure that any open position is closed before the market closes in the same trading day. Specifically, we model our strategy such that, for each single trading day, the final output of our system is one of the following actions:

• A **long** action, which consists of buying the stock, and then selling it before the market closes;

• A **short** action, which consists of selling the stock (using the mechanism of the uncovered sale), and then buying it before the market closes;

• A **hold** action, which consists of deciding not to invest in that day.

The ideal target of this strategy requires the system to choose the action that maximizes the economic return (i.e., the profit) of the day, given a prediction about the stock price trend in that day (i.e., whether the price will rise or fall). Thus, a long action is performed whenever our system predicts that the price will rise in that day; conversely, a short action is chosen whenever our system predicts that the price will fall in that day; last case, a hold action is performed whenever the system is not enough confident about the market behaviour.

B. Gramian Angular Fields Imaging

The GAF imaging is an elegant way to encode time series as images. This has been proposed by Wang and Oates in [30].
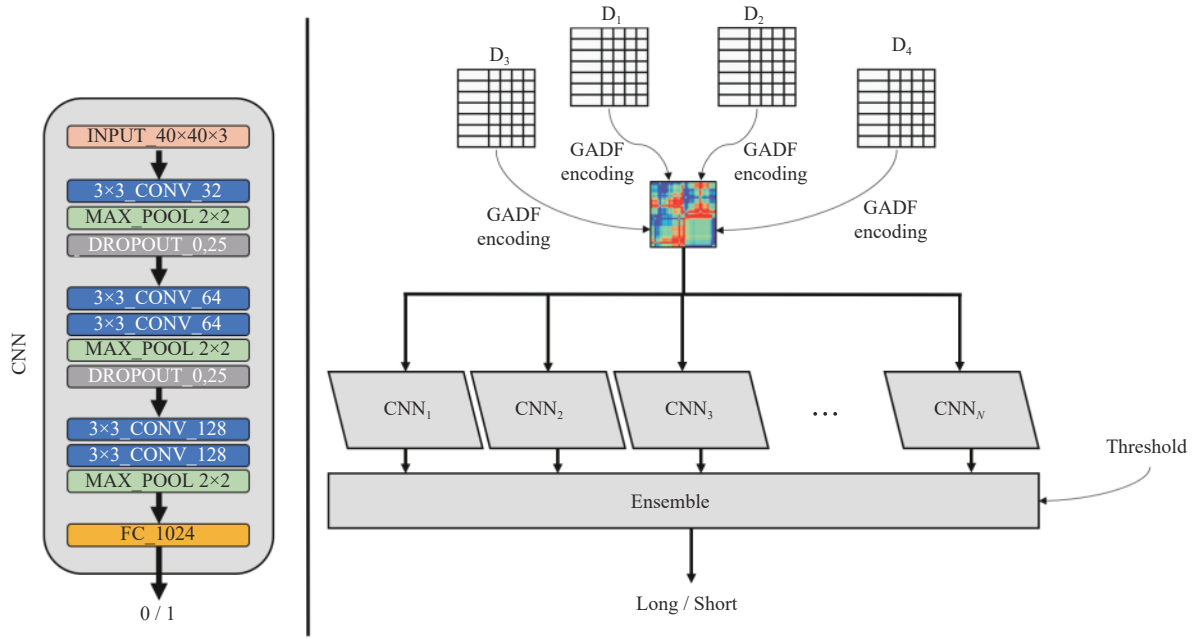
Fig. 1. On the leftmost side of the figure the architecture of the convolutional neural network is shown. On the rightmost side, the overall process of the proposed trading system is depicted.

The main reasons which led to the definition of this approach regards the possibility to use existing pre-trained models, rather than training recurrent neural networks from scratch or using 1D-CNN models. The last two models may result inconvenient.

In order to build the GAF images, first a rescaling of the real observations of the time series is needed; therefore, let $X = \{x_1, x_2, \ldots, x_n\}$ be the considered time series with n components, the rescaling to the interval $[-1, 1]$ is achieved by applying the mean normalization

$$\tilde{x}^i = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)}. \tag{1}$$

Hence, the scaled series is represented by $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n\}$. This is transformed to a polar coordinates system by computing the angular cosine of the single components of the scaled time series

$$\begin{bmatrix} \theta_i = \arccos(\tilde{x}_i), & \tilde{x}_i \in \tilde{X} \\ r_i = \dfrac{i}{N}, & \text{with } t_i \leq N \end{bmatrix}. \tag{2}$$

Finally, Gramian summation angular field (GASF) and Gramian difference angular field can be easily obtained by computing the sum/difference between the points of the time series

$$\text{GASF} = [\cos(\theta_i + \theta_j)] = \tilde{X} \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}' \cdot \sqrt{I - \tilde{X}^2}$$
$$\text{GADF} = [\sin(\theta_i + \theta_j)] = \sqrt{I - \tilde{X}^2}' \cdot \tilde{X} - \tilde{X} \cdot \sqrt{I - \tilde{X}^2}. \tag{3}$$

Fig. 2 shows the process for transforming a time series to the GADF and GASF images. It is worth to notice that the equations in (3) produce a 1D matrix as an output of the encoding process. This matrix actually represents a heatmap, whose values range from 0 (blue) to 1 (red). In a successive step, we applied the RGB color map to the image, thus resulting in a three channel matrix (further details on this process can be found in [28] and [30]). Note that the application of the color map is not strictly required by our approach; however, preliminary experiments showed us that applying the color map to the images led us to obtain better results and, additionally, to achieve a faster network convergence and stability.

C. Multi-Resolution Time Series Imaging

The time-series data show a quite important factor, that is the variation of a feature across the time and, therefore, how quickly data change. The speed which regulates the change of the features provides many insights about the evolution of the event: unfortunately, this peculiarity is hidden or even impossible to identify when data granularity is too coarse. As a trivial example, let us think to how important is the granularity of information in weather forecasting: an actual changing from sunny to rainy which occurs in few minutes gives different information with respect to the same change in 24 hours period. Moreover, often the observations contained in a time series are not done at the same interval of time, i.e., the distance between two consecutive observations is not always the same, thus forcing the researcher to aggregate the data for uniforming this distance through the entire time series. Given the two factors described above, our approach proposes a multi-resolution imaging, which aggregates the data under K different intervals of time, thus creating K different, but analogous, time series. Let $D = \{T, F_1, F_2, \ldots, F_N\}$ be the original time series in which T defines the moment in which the observations of a given event are done, and F_i with $i = 1, 2, \ldots, N$ are the features describing the event. Given two consecutive observations in D, let us say D_j and D_{j+1}, let I_D be the distance between them in terms of time. The new K aggregated time series are $D_1, D_2, \ldots, D_K$, and the interval between two consecutive observations is $I_{D_k} > I_D$ for each $k = 1, \ldots, K$.
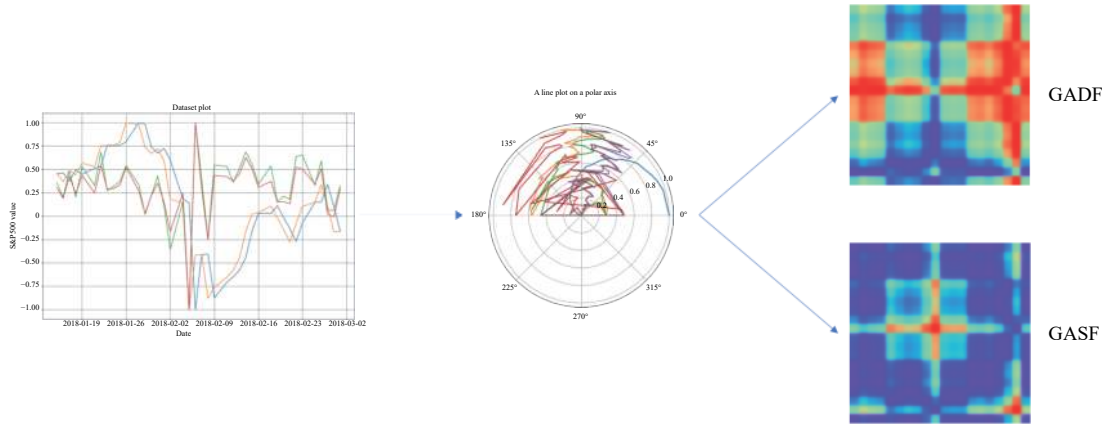
Fig. 2. The policy of composition of the multi-resolution GADF images; in particular, (a)–(d) refer to the same label, but the observations considered in each are aggregated in four different ways.

Fig. 3 shows the composition approach in which (a)–(d) are four GADF images built from four time-series which differ for their aggregation intervals. The composition aims at building a unique image, which considers the evolution of the time series in a fixed period of time.
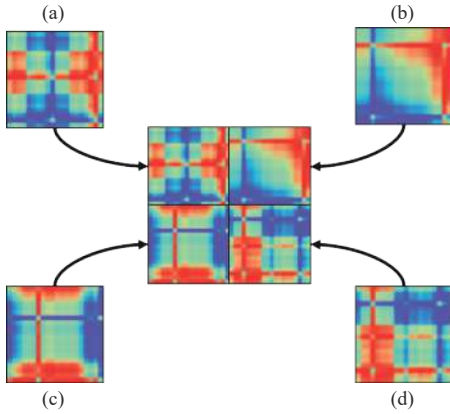

Fig. 3. The process leading to the generation of the GAF images: from left to right, the data are first plotted and then the coordinate system is transformed to a polar plane; finally, the GADF and GASF images are generated according to the function defined in (3).

D. Ensemble of CNNs

Once the time series are converted to GADF images, the training phase can take place. Fig. 1 shows the architecture of the convolutional neural network involved (on the left). It consists of a simplified version of the VGG-16 network [31], composed by 5 convolutional layers and a fully-connected one. Although our approach is independent of the network adopted, our choice was motivated by the fact that very deep networks were not suitable for the task, given the low number of samples at our disposal (S&P500 only has a few thousands of daily samples). Indeed, our early tests showed that the simplified VGG led us to significantly improve the results, learning stability and execution times, both when compared to the standard VGG-16 and ResNet [32] architectures.

Additionally, note that no padding is added, therefore the convolutional layers produce an output of the same size of the input. The activation function in each of them is a ReLU, while in the classification layer the softmax function is used.

The ensemble configuration involves the training of twenty convolutional neural networks, each with a different weight initialization method; the configuration parameters for each network are detailed in Table I. The choice of defining each network with different weight initialization methods arises from the need of excluding most of the randomness factors which can affect the final prediction. This ensures that the answer of the ensemble originates from a proper convergence of the weights, no matters how they are initialized. Once the

TABLE I
FOR EACH TRAINED CNN, A DIFFERENT WEIGHT INITIALIZATION
METHOD IS APPLIED. IN THE TABLE, THE SET OF CONFIGURATION
PARAMENTERS IS REPORTED

Init. method	Configuration parameters	Seed
Orthogonal	gain = 1.0	–
Lecun_uniform	–	–
VarianceScaling	scale = 1, mode = "fan_in," distr. = "normal"	–
RandomNormal	mean = 0.0, stddev = 0.05	–
RandomUniform	minval = –0.05, maxval = 0.05	–
TruncatedNormal	mean = 0.0, stddev = 0.05	–
Glorot_normal	–	–
Glorot_uniform	–	–
He_normal	–	–
He_uniform	–	–
Orthogonal	gain = 1.0	42
Lecun_uniform	–	42
VarianceScaling	scale = 1, mode = "fan_in," distr. = "normal"	42
RandomNormal	mean = 0.0, stddev = 0.05	42
RandomUniform	minval = –0.05, maxval = 0.05	42
TruncatedNormal	mean = 0.0, stddev = 0.05	42
Glorot_normal	–	42
Glorot_uniform	–	42
He_normal	–	42
He_uniform	–	42

nets are trained, the test samples are given as input to all the networks. The final decision of the ensemble is taken by applying a majority voting based approach, which returns a specific output $r \in \{0, 1\}$, according to the percentage of networks which agree with the same classification. In the experimental results section, six different thresholds (related to the agreement percentage of the networks) are tested, from 0.5 to 1, according to the percentage of networks which agree to the same classification.

III. EXPERIMENTAL SETTINGS

This section describes the settings of the experimental phase. In particular, first, in Section III-A the dataset S&P500 is presented and described along with the B&H investment strategy applied on it. This represents the baseline comparison method for the majority of the trading systems. Then, in Section III-B, the validation approach is defined. Finally, in Section III-C the ensemble policy is shown.

A. Evaluation and S&P500

The S&P500 is maybe the most important U.S. index. Born in 1789, at the beginning it only consisted of 90 titles. Since 1957, the year corresponding to when computers started to be actively applied to the financial market, the number of quoted companies has grown up to 500. It followed that S&P500 became one of the most influencing market in the U.S., even overcoming the Dow-Jones index. For the trading operations the S&P500 futures have been used. They consist of futures contracts on a stock or a financial index. These derivative securities are used by investors and portfolio managers to hedge their equity positions against a loss in stocks; in other words, S&P500 index is used by those who want to hedge risk over a certain period of time.

Currently, S&P500 index is one of the most widely traded index future contracts in the U.S. and it is computed by multiplying the S&P500 value by $50. As an example, if the S&P500 is at a level of 2500, then the market value of a future contract is $2500 \times \$50$ or $125\,000.

The S&P500 data are publicly available on many platforms over Internet; also, these can be downloaded at different levels of granularity, according to the scope of the research that is being carried out. Historically the data are tracked and gathered once every 15 s, but some of the platforms only provide data daily. More in detail, for each of the acquired records, the following data are available.
- Date of the observation: mm/dd/yyyy;
- Interval of the observation: hh:mm:ss;
- Open: the value the index has opened in the specified date;
- Close: the value the index has closed in the specified date;
- High: the highest value reached by the index in the specified date;
- Low: the lowest value reached by the index in the specified date.

We decided to use S&P500 as a benchmark for several reasons. First of all, because it is one of the most relevant stock markets in the world. The second reason is that it is extremely challenging to forecast its behaviour. Indeed, it shows a high level of volatility, where this measure indicates the mean fluctuation of the market over a certain time interval. The practical importance of this factor relies on the assumption that there is a tight correlation between the mean fluctuation and the quantification of the risks related to the asset [33]. In the recent years, many studies have been conducted in this area to the point that a specific branch of the market-related research focuses on the prediction of this fluctuation [34]. On top of this, we can use the performances of the B&H applied on S&P500 as a direct benchmark. B&H is both a strategy easy to replicate, and also an extremely significant competitor when considering a time frame where the market performs a large and quite constant growth. Indeed, B&H is a passive investment strategy where an investor buys stocks and holds them for a long time, with the hope that stocks will gradually increase in value over a long period of time. This strategy works as follows: given an investment period, B&H "buys" a stock at the beginning, "holds" it for the entire investment period, and sells it at the end. The net profit is the difference between the price at the end period and the price at the beginning. In Fig. 4, the selected time frame for comparing our approach against the B&H strategy is shown (from 2009-02-01 to 2014-07-31); it is easy to notice that this investment period is extremely favourable for B&H and extremely challenging for our approach.

B. Walks' Definition

With respect to the common cross-validation approaches, which are typically applied when dealing with image classification, like the leave-one-out cross validation (LOOCV) or the *k*-fold cross validation, time series data need a more specific purpose approach, since they need to consider the semantic linking between the observation at time *t* and the one at time $t+1$. The walk-forward validation strategy properly fits in this scenario since the folds which are considered for the validation are temporally split, and internally processed as one sample. As a consequence, the use of GADF helps maintaining the correlation between two consecutive observations, thus keeping unaltered the semantic of the succession.

For our own research we considered an investment period which goes from 2000-02-01 to 2015-01-30, for a total of 16 years and 4569 observations which are related to the actual days in which the financial market has been opened. Each observation has been labelled according to the difference close-open of the day after which is
- 1, if the close-open value of the day after is positive;
- 0, otherwise.

The data are divided in training, validation and testing sets; in Table II, the walks are shown: in particular, each model has been trained over a period of ten years, validated over the following six months and finally tested over the last six months.

C. CNN Training and Ensemble Policy

According to the considerations exposed in Section II-C, we used $K = 4$ for the evaluation of our approach. Starting from the original time series of S&P500, in which the observations
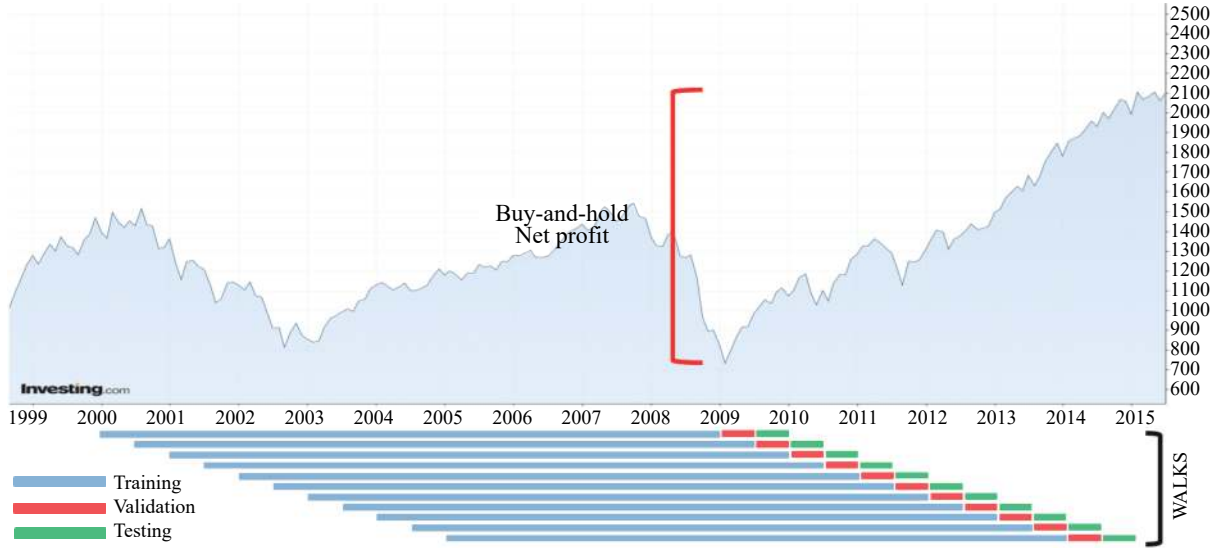
Fig. 4.   The net profit obtained by the application of the buy-and-hold strategy is shown over the period which has been taken into account for the experiments (https://www.investing.com/indices/us-spx-500-futures).

TABLE II
THE 11 WALKS USED FOR TRAINING, VALIDATION, AND TESTING THE MODELS AND HOW THEY ARE COMPOSED

	Training	Validation	Test
#	since → to	since → to	since → to
1	2000-02-01 → 2009-01-30	2009-02-01 → 2009-07-31	2009-08-02 → 2010-01-31
2	2000-08-01 → 2009-07-31	2009-08-02 → 2010-01-31	2010-02-01 → 2010-07-30
3	2001-02-01 → 2010-01-31	2010-02-01 → 2010-07-30	2010-08-01 → 2011-01-31
4	2001-08-01 → 2010-07-30	2010-08-01 → 2011-01-31	2011-02-01 → 2011-07-31
5	2002-02-01 → 2011-01-31	2011-02-01 → 2011-07-31	2011-08-01 → 2012-01-31
6	2002-08-01 → 2011-07-31	2011-08-01 → 2012-01-31	2012-02-01 → 2012-07-31
7	2003-02-02 → 2012-01-31	2012-02-01 → 2012-07-31	2012-08-01 → 2013-01-31
8	2003-08-01 → 2012-07-31	2012-08-01 → 2013-01-31	2013-02-01 → 2013-07-31
9	2004-08-01 → 2013-07-31	2013-08-01 → 2014-01-31	2014-02-02 → 2014-07-31
10	2004-08-01 → 2013-07-31	2013-08-01 → 2014-01-31	2014-02-02 → 2014-07-31
11	2005-02-01 → 2014-01-31	2014-02-02 → 2014-07-31	2014-08-01 → 2015-01-30

are sampled at intervals of 5 minutes, the data have been aggregated according to 4 new intervals:

• 1 hour;
• 4 hours;
• 8 hours;
• 1 day.

This means that starting from the original time series, we aggregated the data in four different ways and from each we selected 20 samples for predicting the market-day after (20 1-hour blocks make one GADF for the first time series, 20 4-hours blocks make one more GADF image, etc.).

The close-open value has been computed for each sample, and a $20 \times 20 \times 3$ GADF image has been built and composed according to the process shown in Fig. 3. By following this procedure, the aggregated GADF image will have dimension $40 \times 40 \times 3$. For each of the walks defined in the previous section, the ensemble of CNNs is executed over the test samples; the majority voting approach works as follows: each of the

CNN gives as output a single-value which is 0 or 1 whether the CNN suggests to perform a short or a long action, respectively. When working in ensemble, the final prediction is taken according to the answer of all the CNNs, according to the trading strategy defined in Section II-A. Note that in cases where not all the CNNs agree with the same result, a coverage-base approach is used: let N be the number of networks involved in the ensemble, $t \in [0.5, 1]$ be a threshold which indicates the required percentage of agreement, and A be the most predicted action (between long or short) by the nets for the considered day. Then, the system performs A in that day if and only if at least $n > tN$ networks have voted A; conversely, if the agreement threshold is not reached, the system does not perform any operation and just holds.

IV.   RESULTS AND DISCUSSION

In this section, the quantitative and qualitative results are shown and discussed. In the following we report the metrics we have used.

• The *Accuracy* is a well-known metric; it is defined as the ratio between the number of correct predictions against the total number of test samples;

• The *Coverage* is a metric (expressed as a percentage) which indicates how many times the networks in the ensemble agree on an action, according to the threshold set;

• The annualized *Sharpe Ratio* is a financial risk index used to help investors to understand the return of an investment compared to its risk; the greater the value of the *Sharpe Ratio*, the more attractive the risk-adjusted return;

• As the term suggests, the *Net Profit* is the actual profit we earned in the testing period;

• The *Sortino Ratio* is a variation of the *Sharpe Ratio* that differentiates harmful volatility from total overall volatility by using the asset's standard deviation of negative portfolio returns; it is defined as follows:

$$Sortino\ Ratio = \frac{R_p - r_f}{\sigma_d} \qquad (4)$$

where R_p is the actual portfolio return, r_f is the risk-free rate, and σ_d is the standard deviation of the downside;

• The return over maximum drawdown (*RoMaD*) is the average return in a given period for a portfolio; it is computed as follows:

$$RoMaD = \frac{NetProfit}{MDD} \qquad (5)$$

where the *Portfolio Return* is the effective gain or loss realized by an investment, and the *MDD* indicates the maximum draw down (*MDD*), which is defined as the maximum loss from a peak to a trough of a portfolio, before a new peak is attained. Currently the *MDD* is the preferred way to evaluate the riskiness of an investment.

The results of the proposed approach have been compared against the following baselines:

• B&H strategy: as explained in Section III, this represents the baseline comparison method for the majority of the trading system approaches;

• Random guessing: this easy technique exploits 10 random classifiers, included in an ensemble that using a majority voting approach tries to perform long or short operations. The comparison against this random predictor serves as proof that our approach does not act randomly, that the performed actions (longs or shorts) have a strong basis, and that the criteria are correctly learnt from the past trend of the market;

• 1D-CNN: this approach does not apply the GADF transformation to the time series; therefore, the time series are directly applied and processed by 1D-CNN. This test aims at showing the benefits of the GADF transformations in the proposed approach.

Fig. 5 shows the performances of the simple trading system based on our forecasting approach (in blue) against those described before: the B&H strategy (in red), the 1D-CNN (in green) and the random guessing (in orange). The performances are computed over all the walks that have been taken into account; on the *x*-axis of each plot the considered threshold for the ensemble is indicated. Overall, the thresholds from 0.5 to 0.7 let us obtain the best achievements. As shown in the plot, the

proposed approach is the only one capable of sensitively overcome the baseline performance of the B&H. However, a peak in the performance is appreciable with the threshold set to 0.6, in which the highest net profit is obtained. The quantitative results are shown in Table III and further confirm the good qualitative scores. Moreover, the table shows the big difference between the net profit obtained by our approach and the B&H strategy. Using ensemble thresholds higher than 0.7, the results tend to degrade, mainly due to the fact that it becomes harder to obtain a total agreement among the nets in the ensemble. Anyway, even with a threshold set to 1 (meaning that all the nets must agree to trigger a long or short action), we managed to earn $ 1.375; thus, we still do not lose money, even though the risk is quite high, as shown in the *Sharpe Ratio* and *Sortino Ratio* plots. In general, the good performances for all the threshold values are a clear indicator of the robustness of our proposed approach.

In addition to comparing our results to the baselines, we have also considered the approach proposed by Calvi *et al.* [29], which consists of the daily prediction of the S&P500 index, as in our case, but by using a support tensor machine (STM) as a predictor (defined as a tensor extension of the better known support vector machine). Fig. 6 shows the gap between the net profit given by our method, and those returned by the B&H and the work in [29]. The comparison does not take into account the market period related to the 2008's Financial Crisis (whose effects impacted the years between 2007 and early 2009), when it results unfair to overcome the B&H strategy, due to the dramatic performances of the index in this period.

Conversely, the considered period (from late 2009 until 2016) highlights a strong increase of the S&P500 index, thus improving the B&H performances. Nevertheless, our method is able to outperform such performances, which is not the case of the approach in [29].

Finally, note that, in the financial forecasting domain, the prediction accuracy is very close to 50%, mainly due to the high variability of the indexes which complicates a lot the decision between long and short actions. In this scenario, using a single CNN (both in terms of prediction and, where appropriate, in terms of probability of the prediction) usually introduces a major drawback: the initialization seed may significantly affect the final result. Moreover, focusing on improving the accuracy does not usually imply the net profit to grow accordingly. It follows that, by slightly changing the initialization, the same network could give worse results for the same period, or in any case could behave too differently when tested on different markets. This reveals the need of stabilizing the results; the ensemble architecture, together with different weight initialization policies, has shown to have a much more robust behaviour, allowing us to obtain more stable results and thus alleviate the randomness. As a result, according to the set thresholds, we perform a long operation only when the ensemble suggests to buy when a certain ensemble voting policy is satisfied; otherwise we perform a short operation. The mitigation of the randomness yields two simple but significant consequences.

• When we lose, we tend to lose very little;

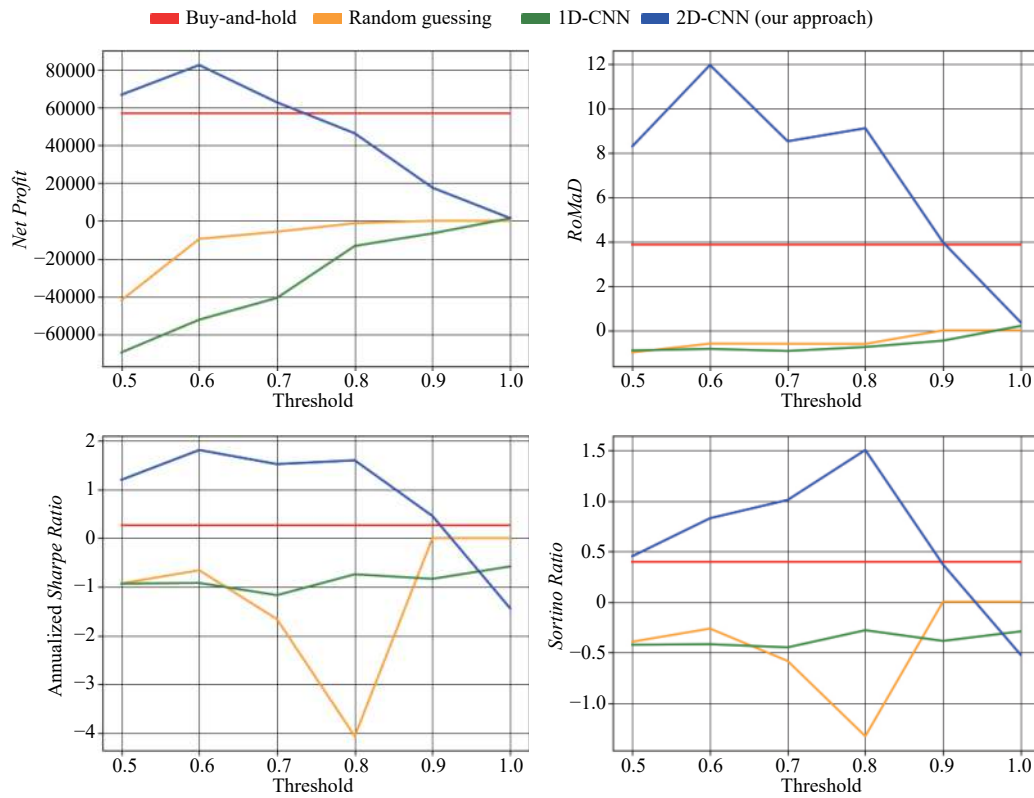• When we win, we tend to win considerably.

Fig. 5. The comparison among the tested approaches. In particular, the blue, orange and green lines represent, respectively, the results obtained by applying our approach, a random guessing approach and a 1D-CNN. The red line evidences the results which are obtained by the buy-and-hold strategy. In each plot, the *x*-axis indicates the ensemble threshold considered, while in the *y*-axis it is shown the obtained *Net Profit*, the *RoMaD* value, the annualized *Sharpe Ratio* and the *Sortino Ratio*, respectively, from the upper-leftmost image to the bottom-rightmost one.

TABLE III
THE QUANTITATIVE RESULTS OF THE PROPOSED APPROACH. FOR EACH METRIC, THE BEST RESULT IS HIGHLIGHTED IN BOLD

|	Threshold	Coverage (%)	Accuracy (%)	Net Profit ($)	RoMaD	Annualized *Sharpe Ratio*	Sortino Ratio
	0.500	**95.433**	52.638	66.625	8.289	1.196	0.452
	0.600	72.423	54.810	**82.312**	**11.95**	**1.808**	0.828
Proposed approach	0.700	47.716	56.564	62.600	8.517	1.518	1.009
	0.800	22.950	**56.632**	46.2125	9.105	1.596	**1.504**
	0.900	7.494	53.125	17.4875	3.970	0.452	0.370
	1.000	1.990	55.882	1.375	0.365	−1.445	−0.524
	0.500	94.906	47.254	−69.525	−0.904	−0.936	−0.425
	0.600	74.824	46.791	−52.250	−0.832	−0.921	−0.420
1D-CNN	0.700	53.688	46.564	−40.725	−0.927	−1.172	−0.451
	0.800	33.723	46.875	−13.2875	−0.749	−0.743	−0.281
	0.900	15.456	48.106	−6.650	−0.463	−0.836	−0.387
	1.000	7.494	49.218	1.3375	0.200	−0.582	−0.293
	0.500	82.552	48.297	−41.912	−0.991	−0.930	−0.395
	0.600	26.639	50.109	−9.575	−0.596	−0.661	−0.265
Random guessing	0.700	4.449	42.105	−5.775	−0.604	−1.665	−0.586
	0.800	0.292	20.000	−1.275	−0.610	−4.067	−1.331
	0.900	0.000	0.000	n.a.	n.a.	n.a	n.a
	1.000	0.000	0.000	n.a.	n.a.	n.a.	n.a.
B&H	–	–	–	56.600	3.847	0.258	0.394
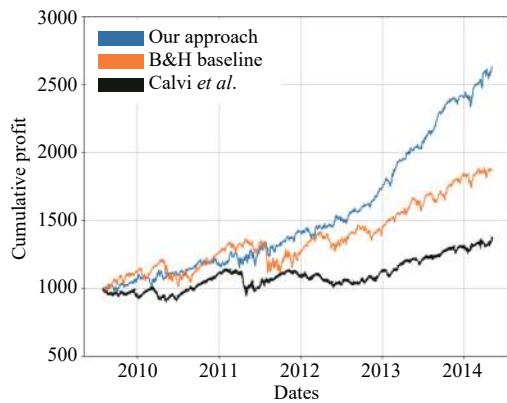
Fig. 6. The comparison, in terms of cumulative profit, between the proposed approach (in blue) and the approach in [29] (in black). Finally, in orange, we show the B&H baseline as a benchmark.

This result is to be considered particularly significant, thanks to the capability of our approach of beating the B&H strategy in the years in which the latter performs well.

V. CONCLUSIONS

In this paper we have proposed an innovative approach for the forecasting of market behaviour by using deep learning technologies and by encoding time series to GAF images. The developed CNNs have been applied to the GAF images for a classification task. Moreover, an ensemble was fed with the CNNs above ans a majority voting strategy has been used to select the final classification. High results have been obtained using the S&P500 future, the market where we have trained, validated and tested our networks and the overall ensemble. The GAF imaging technique has thus been applied within the financial technology domain bringing the benefits of CNN. Moreover, our approach, combination of deep learning and GAF images technologies, outperformed the baseline strategies consisting of B&H operations for a classification task where long and short actions can be performed. The analysis of the tuning of several hyperparameters is being carried out by our team and is subject of future works. Currently we are studying the stacking policy of the GAF images (as shown in Fig. 3), and how accuracy and net profit vary according to the value of K (currently set to 4). Note that our approach outperforms the B&H strategy, although the latter was still very competitive and profitable within the considered period. As our approach is highly promising, there are several directions we would like to explore.

• First of all, we would like to apply our method to other markets and understand the benefits it brings with respect to the baselines;

• Then, something we are already exploring consists of applying the results of our ensemble to real trading platforms. The goal is to simulate the real earnings we would obtain on the past data and on a certain market. The platform we are already playing with is MultiCharts[1];

• We would also like to test our approach on different classification tasks within several domains such as sentiment

analysis, emotion detection, credit scoring. The reason is to understand how the GAF imaging performs in presence of text or different kind of feature vectors.

REFERENCES

[1] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks," in *Proc. Int. Joint Conf. Neural Networks*, San Diego, CA, USA, 1990, pp. 1–6.

[2] Y. D. Zhang and L. E. Wu, "Stock market prediction of S&P500 via combination of improved BCO approach and BP neural network," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 8849–8854, Jul. 2009.

[3] T. Z. Tan, C. Quek, and G. S. Ng, "Brain-inspired genetic complementary learning for stock market prediction," in *Proc. IEEE Congr. Evolutionary Computation*, Edinburgh, Scotland, UK, 2005, pp. 2653–2660.

[4] S. Soni, "Applications of ANNs in stock market prediction: a survey," *Int. J. Comput. Sci. Eng. Technol.*, vol. 2, no. 3, pp. 71–83, 2011.

[5] T. Lintonen and T. Raty, "Self-learning of multivariate time series using perceptually important points," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 6, pp. 1318–1331, Nov. 2019.

[6] K. Kamijo and T. Tanigawa, "Stock price pattern recognition-a recurrent neural network approach," in *Proc. Int. Joint Conf. Neural Networks*, San Diego, CA, USA, 1990, pp. 215–221.

[7] C. H. Lee and K. C. Park, "Prediction of monthly transition of the composition stock price index using recurrent back-propagation," in *Artificial Neural Networks*, I. Aleksander and J. Taylor, Eds. Amsterdam, Netherlands: Elsevier, 1992, pp. 1629–1632.

[8] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10389–10397, Aug. 2011.

[9] S. C. Gao, M. C. Zhou, Y. R. Wang, J. J. Cheng, H. Yachi, and J. H. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.

[10] D. B. Jia, S. X. Zheng, L. Yang, Y. Todo, and S. C. Gao, "A dendritic neuron model with nonlinearity validation on Istanbul stock and Taiwan futures exchange indexes prediction," in *Proc. 5th IEEE Int. Conf. Cloud Computing and Intelligence Systems*, Nanjing, China, 2018, pp. 242–246.

[11] T. L. Zhou, S. C. Gao, J. H. Wang, C. Y. Chu, Y. Todo, and Z. Tang, "Financial time series prediction using a dendritic neuron model," *Knowl-Based Syst.*, vol. 105, pp. 214–224, Aug. 2016.

[12] J. D. Farmer and A. W. Lo, "Frontiers of finance: evolution and efficient markets," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 18, pp. 9991–9992, Aug. 1999.

[13] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of twitter data for predicting stock market movements," in *Proc. Int. Conf. Signal Processing, Communication, Power and Embedded System*, Paralakhemundi, India, 2016, pp. 1345–1350.

[14] A. Mittal and A. Goel, "Stock prediction using twitter sentiment analysis," 2012. [Online]. Available:http://cs229.stanford.edu/proj2011/ GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf

[15] N. Oliveira, P. Cortez, and N. Areal, "The impact of microblogging data for stock market prediction: using twitter to predict returns, volatility, trading volume and survey sentiment indices," *Expert Syst. Appl.*, vol. 73, pp. 125–144, May 2017.

[16] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Networks 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Como, Italy, 2000, pp. 348–353.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sept. 1995.

[1]https://www.multicharts.com/

[18] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: machine learning techniques applied to financial market prediction," *Expert Syst. Appl.*, vol. 124, pp. 226–251, Jun. 2019.

[19] C. F. Tsai and S. P. Wang, "Stock price forecasting by hybrid machine learning techniques," in *Proc. Int. MultiConf. Engineers and Computer Scientists*, Hong Kong, China, 2009, pp. 60.

[20] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2162–2172, Mar. 2015.

[21] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: a review and taxonomy of prediction techniques," *Int. J. Financ. Stud.*, vol. 7, no. 2, pp. 26, Jun. 2019.

[22] M. Ballings, D. van den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7046–7056, Nov. 2015.

[23] S. Basak, S. Kar, S. Saha, L. Khaidem, and S. R. Dey, "Predicting the direction of stock market prices using tree-based classifiers," *North Am. J. Econ. Finance*, vol. 47, pp. 552–567, Jan. 2019.

[24] S. Dey, Y. Kumar, S. Saha, and S. Basak, "Forecasting to classification: predicting the direction of stock market price using Xtreme gradient boosting," 2016. [Online]. Available: https://doi.org/10.13140/RG.2.2.15294.48968

[25] S. K. Aggarwal, L. M. Saini, and A. Kumar, "Price forecasting using wavelet transform and LSE based mixed model in Australian electricity market," *Int. J. Energy Sector Manage.*, vol. 2, no. 4, pp. 521–546, Nov. 2008.

[26] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 82–95, Jan. 2020.

[27] D. Freire-Obregón, F. Narducci, S. Barra, and M. Castrillón-Santana, "Deep learning for source camera identification on mobile devices," *Pattern Recognit. Lett.*, vol. 126, pp. 86–91, Sept. 2019.

[28] Z. G. Wang and T. Oates, "Imaging time-series to improve classification and imputation," in *Proc. 24th Int. Joint Conf. Artificial Intelligence*, 2015.

[29] G. G. Calvi, V. Lucic, and D. P. Mandic, "Support tensor machine for financial forecasting," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Brighton, United Kingdom, 2019, pp. 8152–8156.

[30] Z. G. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Proc. Workshops at the 29th AAAI Conf. Artificial Intelligence*, 2015.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv: 1409.1556*, 2014.

[32] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[33] P. Cizeau, Y. H. Liu, M. Meyer, C. K. Peng, and H. E. Stanley, "Volatility distribution in the S&P500 stock index," *Phys. A Stat. Mech. Appl.*, vol. 245, no. 3–4, pp. 441–445, Nov. 1997.

[34] M. Martens, "Measuring and forecasting S&P500 index-futures volatility using high-frequency data," *J. Futur. Mark.*, vol. 22, no. 6, pp. 497–518, Jun. 2002.

Silvio Barra received the B.Sc. (cum laude) and the M.Sc. degrees (cum laude) in computer science from University of Salerno, Italy, in 2009 and 2012, respectively. In 2017, he received the Ph.D. degree at the University of Cagliari, Italy. Currently he is a research Fellow at the University of Cagliari. He is a Member of the GIRPR. His main research interests include pattern recognition, biometrics, and video analysis and analytics.
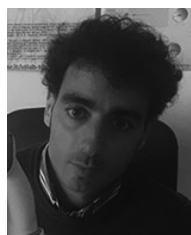

Salvatore Mario Carta is an Associate Professor at the Department of Mathematics and Computer Science of the University of Cagliari, Italy. He received the Ph.D. degree in electronics and computer science from the University of Cagliari in 2003 and in 2005 he joined the Department of Mathematics and Computer Science of the University of Cagliari as an Assistant Professor. In 2006 and 2007 he was a Guest Researcher at the Swiss Federal Institute of Technology as invited Professor, hosted by Laboratoire des Systmes Intgrs-LSI. His research interests include clustering algorithms, social media analysis, and text mining for behavioral pattern identification and recommendation in group of users and in single users; AI algorithms for credit scoring, fraud detection and intrusion detection; AI algorithms for financial forecasting and robo-trading; E-coaching platforms and AI algorithms for healthy lifestyles. He is author of more than 100 conference and journal papers in these research fields, with more than 1200 citations and is Member of the ACM and of the IEEE. He founded 3 hi-tech companies, spin off of the University of Cagliari, and is currently leading one of them.

Andrea Corriga is a Ph.D. candidate at the Department of Mathematics and Computer Science of the University of Cagliari, Italy. He received the master degree in computer science in 2018. His research interests include financial forecasting using deep learning and machine learning approaches.

Alessandro Sebastian Podda is a Postdoctoral Researcher at the Department of Mathematics and Computer Science of the University of Cagliari, Italy. In 2018, he received the Ph.D. degree in mathematics and informatics, supported by a grant from RAS (Autonomous Region of Sardinia). Previously, he received the B.Sc. and M.Sc. degrees in informatics (both with honours) at the University of Cagliari. Currently, he is a Technical Coordinator of the research projects DoUtDes and SardCoin. His current research interests mainly include deep learning, financial forecasting, information security, blockchains, and smart contracts.

Diego Reforgiato Recupero has been an Associate Professor at the Department of Mathematics and Computer Science of the University of Cagliari, Italy, since December 2016. He received the Ph.D. degree in computer science from the University of Naples Federico II, Italy, in 2004. From 2005 to 2008 he has been a Postdoctoral Researcher at the University of Maryland College Park, USA. He won different awards in his career (such as Marie Curie International Reintegration Grant, Marie Curie Innovative Training Network, Best Research Award from the University of Catania, Computer World Horizon Award, Telecom Working Capital, Startup Weekend). He co-founded 6 companies within the ICT sector and is actively involved in European projects and research (with one of his companies he won more than 30 FP7 and H2020 projects). His current research interests include sentiment analysis, semantic web, natural language processing, human robot interaction, financial technology, and smart grid. In all of them, machine learning, deep learning, big data are key technologies employed to effectively solve several tasks. He is author of more than 100 conference and journal papers in these research fields, with more than 1000 citations.