Trading via Image Classification[*]

Naftali Cohen
AI Research, J.P. Morgan
New York, NY 10179
naftali.cohen@jpmorgan.com

Tucker Balch
AI Research, J.P. Morgan
New York, NY 10179
tucker.balch@jpmorgan.com

Manuela Veloso[†]
AI Research, J.P. Morgan
New York, NY 10179
manuela.veloso@jpmorgan.com

Abstract

The art of systematic financial trading evolved with an array of approaches, ranging from simple strategies to complex algorithms all relying, primary, on aspects of time-series analysis (e.g., Murphy, 1999; De Prado, 2018; Tsay, 2005). Recently, after visiting the trading floor of a leading financial institution, we noticed that traders always execute their trade orders while *observing* images of financial time-series on their screens. In this work, we built upon the success in image recognition (e.g., Krizhevsky, Sutskever, and Hinton, 2012; Szegedy et al., 2015; Zeiler and Fergus, 2014; Wang et al., 2017; Koch, Zemel, and Salakhutdinov, 2015; LeCun, Bengio, and Hinton, 2015) and examine the value in transforming the traditional time-series analysis to that of image classification. We create a large sample of financial time-series images encoded as candlestick (Box and Whisker) charts and label the samples following three algebraically-defined binary trade strategies (Murphy, 1999). Using the images, we train over a dozen machine-learning classification models and find that the algorithms are very efficient in recovering the complicated, multiscale label-generating rules when the data is represented visually. We suggest that the transformation of continuous numeric time-series classification problem to a vision problem is useful for recovering signals typical of technical analysis.

Introduction

Traders in the financial markets execute buy and sell orders of financial instruments as stocks, mutual funds, bonds, and options daily. They execute orders while reading news reports and earning calls, but also while observing charts of time-series data that indicate the historical value of particular securities, or leading financial indices (see Fig. 1 for a typical workstation of a professional trader[1]). Many algorithms have been developed to analyze continuous financial time-series data to improve a trader's ability to decide to buy or sell a particular security (Murphy, 1999). Conventional

algorithms process time-series data as a list of numerical data, aiming at detecting patterns as trends, cycles, correlations, etc. (e.g., De Prado, 2018; Tsay, 2005). In case a pattern is identified, the analyst can then construct an algorithm that will use the detected pattern (e.g., Wilks, 2011) to predict the expected future values of the sequence in hand (i.e., forecasting using exponential smoothing models, etc.).

Experienced traders with years of experience observing financial time-series charts and executing buy and sell orders start developing an intuition for market opportunities up to a point in which their intuition, based on observing charts, almost reflects the recommendation that their state-of-the-art model provides (personal communication with J.P. Morgan's financial experts Jason Hunter, Joshua Younger, Alix Floman, and Veronica Bustamante). In this perspective, financial time-series analysis can be thought of as a visual process; when experienced traders look at a time-series data, they process and act upon the image instead of mentally executing algebraic operations on the sequence of numbers.

Figure 1: Typical workstation of a professional trader. Credit: Photoagriculture / Shutterstock.com.

In this study, we create and analyze an extensive financial time-series data set and using a supervised classification approach (e.g., Bishop, 2006; Goodfellow, Bengio, and Courville, 2016; Aggarwal, 2015), we evaluate predictions using over 15 different classifiers when the input data is presented graphically as images. We make use of three

---

[*]Patent pending

[†]On leave: Machine Learning Department, Carnegie Mellon University

Preprint. Paper under review

[1]The photo was taken in a trading room at Rouen, Normandie, France, September 2015.

label-generating rules following three algebraically-defined binary trade strategies (Murphy, 1999) and show that the classifiers are very efficient in recovering the complicated, sometimes multiscale, labels.

Related Work and Main Contributions

The focus of this work is on the representation of financial time-series data as images. Previous work on time-series classification suggests transforming the data either locally using wavelets or globally using Fourier transforms and then compare the various data according to their relevant modes of variability in the transformed space (e.g., Wilks, 2011). Other methods apply similarity metrics as Euclidean distance, k-nearest neighbors, dynamic time warping, or even Pearson correlations to separate between the classes (e.g., Aggarwal, 2015). In addition to the above, other techniques focus on manual feature engineering, detecting a frequently occurring pattern or shape in the time series (e.g., Bagnall et al., 2017).

More recently, it was suggested to approach time-series classification by first encoding the data as images and then utilize the power of computer vision algorithms for classification (Park et al., 2019). In an example, it was suggested to encode the time dependency, implicitly, as Gramian-Angular fields, Markov-Transition fields (Wang and Oates, 2015a; Wang and Oates, 2015b), or make use of recurrence plots (Souza, Silva, and Batista, 2014; Silva, De Souza, and Batista, 2013; Hatami, Gavet, and Debayle, 2018) as a graphical representation. Another work focused on transforming financial data into images to classify candlesticks patterns (Tsai, Chen, and Wang, 2019).

This paper is written under the assumption that, given public knowledge, markets are efficient (e.g., Pedersen, 2019). That is, future market movements are mostly random and have almost no predictability. However, the way professional people trade is systematic (i.e., consistent, backtested, and might even be profitable for a short duration) and can be identified using a set of rules. In many cases, systematic trading is done or at least augmented with visual representations. In this paper, we examine the value of using images alone for identifying trade opportunities typical for technical analysis. To the best of our knowledge, our work is the first work that built upon the great success in image recognition (e.g., Krizhevsky, Sutskever, and Hinton, 2012; Koch, Zemel, and Salakhutdinov, 2015; LeCun, Bengio, and Hinton, 2015) and tries to systematically apply it to numeric time-series classification by taking a *direct* graphical approach and recency-biased label-generating rules. The contributions of this paper are the following:

1. The main contribution is in bridging between the unrelated areas of numerical and quantitative finance and image recognition. The former involved a mixture of technical, quantitative analysis, and financial knowledge, while the second involves advanced algorithm design and computer science knowledge. In this paper, we show how the two distinct areas can leverage knowledge and techniques from each other.

2. The second contribution is that we show that the concept of visual time-series classification is effective and works on real data. A large fraction of the artificial-intelligence research is conceptual and work only on synthetic data. As will be shown, the concepts introduced in this paper are effective on real data and can be put to use right away as a marketing recommendation tool and as a forecasting tool.

3. The third significant contribution is that, in practice, there are financial domains in which professional investment decisions are made using visual representations alone (e.g., swap trade) – relying, in fact, on traders intuition, experience, skill, and luck. In cases like that, it is more than natural to use the visual representation as an input to the model.

Data and Methods

In this study, we analyze the daily values of all S&P 500 stocks for the period starting in 2010 (hereafter SP500 data). The S&P 500 stocks were issued by large-cap companies and are actively traded on American stock exchanges. The capitalization of these companies covers the vast majority of the American equity market (e.g., Berk et al., 2013).

Figure 2: Converting continuous time series to images.

Trading is done continuously (during trade hours which usually span between 9:30 am to 4:00 pm, not including pre- and after-market hours) but we are using a discretized form of the continuous data by accounting only for the start, max, min, and end values per stock per day. In the financial jargon, these values are termed as the Open, High, Low, and Close (OHLC) values (e.g., Murphy, 1999). We visualize the data using a box-and-whisker (also called candlestick) diagram, where box edges mark the Open and Close price, while the whiskers mark the Low and High values (i.e., daily min and max). The color of each box reveals whether the Open price ended up being higher or lower than the Close price for the same day; if Open > Close the box in filled in black indicating Bear's market, whereas if Open < Close the box is filled in white indicating Bull's market (e.g., Murphy, 1999). Figure 2 shows an example of this process by focusing attention on the AAPL ticker for Feb 9, 2019, and Feb 28, 2019.

The left columns show the 1-minute continuous trading data during trading hours, while the right column detail the discretization process. Notice that the upper left time-series experience a positive trend resulting in a white candlestick visualization, while the bottom left time-series data experience a negative trend resulting in a black candlestick.

We compare three well-known binary indicators (Murphy, 1999), where each is based on prescribed algebraic rules that depend solely on the Close values. Each indicator alerts the trader only for a buying opportunity, thus if a trader decides to follow (one of) the signals he/she might do so no earlier than the day after. The three "buy" signals are defined as follows:

• BB crossing: The Bollinger Bands (BB) of a given time-series consists of two symmetric bands of 20-days moving two standard deviations (Colby and Meyers, 1988). The bands envelop the inherent stock volatility while filtering the noise in the price action. Traders use the price bands as bounds for trading activity around the price trend (Murphy, 1999). Hence, when prices approach the lower band or go below, prices are considered to be in an oversold position and trigger a buying opportunity. Here, the bands are computed using the (Adjusted) Close values, and hence a buy signal is *defined* to trigger when the daily Close value *crosses above* the lower band.

Figure 3 shows an example of a Buy signal opportunities for the AAPL stock during 2018. In solid black one can see the daily Close values for the ticker while the red line shows the 20-days moving average (inclusive) of the price line. The dashed black lines mark the two standard deviations above and below the moving average line. The BB crossing algorithm states that a Buy signal is initiated when the price line (in solid black) crossed above the lower dash black line. In this Figure, marked by the red triangles, one can identify eight such buy opportunities.
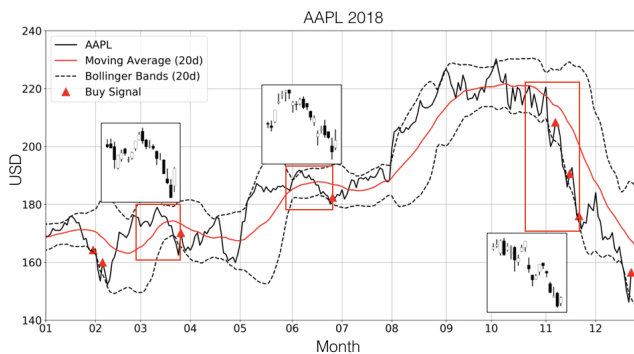

Figure 3: Labeling time series data according to the Bollinger Bands crossing rule.

• MACD crossing: Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that compares the relationship between short and long exponential moving averages (EMA) of an asset (Colby and Meyers, 1988). As is common in finance (e.g., Murphy, 1999), we compute the MACD by subtracting the

26-days EMA from the 12-days EMA. When MACD falls to negative values, it suggests negative momentum, while conversely when the MACD rises to positive values, it indicates for upward momentum. Traders usually wait for consistent measures and thus further smooth the MACD line and compute the 9-days EMA of the MACD, called the signal line. Here, the MACD buy signal is *defined* to trigger when the signal line *crosses above*.

• RSI crossing: The Relative Strength Index (RSI) is an oscillating indicator that summarizes the magnitude of recent price changes to evaluate overbought or oversold conditions of an asset. As is common in finance (e.g., Colby and Meyers, 1988; Murphy, 1999), we compute RSI as the ratio 14-day EMA of the incremental increase to the incremental decrease in asset values. The ratio is then scaled to values that vary between 0 and 100: it rises as the number and size of daily gains increase and falls as the number and size of daily losses increase. Traders use RSI as an indication for an overbought state which might trigger a sell order or an oversold state which might trigger a buy order. The standard thresholds for oversold/overbought RSI are 30/70, respectively (Murphy, 1999). Here, the RSI buy signal is *defined* to trigger when the RSI line *crosses above* the value of RSI=30.

Figure 3 shows three positively-labeled images that correspond to the BB-crossing algorithm. These images are generated by enveloping a 20-day of stock activity (the red rectangles) before and including the buy-signal day activity. It is also possible to create negatively-labeled images from this time series by enveloping activity, in the same way, for days with no buy signal. Note also that these images tightly bound the trade activity and do not contain labels, tickers, or title, which is the essential input-date standardization process we apply in this study.

Results

The objective of this study is to examine whether we can train a model to recover trade signals in time-series data that are typical of technical analysis and defined algebraically. Hence, in the following, we examine the supervised classification predictions of the SP500 images that are labeled according to the BB, RSI, and MACD algorithms.

The data set is balanced, containing 5K samples per class per indicator. That is, for each of the S&P500 tickers, we compute all buy triggers for the period between 2010 and the end of 2017. We then choose, at random, ten buy triggers for each ticker and create corresponding images. In the same way, we choose, at random, ten no-buy triggers per time-series and create similar images. This process results in 10K high-resolution images per trigger.

A key difference between the three algorithms, besides their various complexity, is the time-span each considers. While the BB algorithm takes into account 20-days of action, RSI, which uses exponential-moving averaging, considers, effectively, 27 days, while MACD, which also uses exponential moving averages, span effectively over 26 days. For each of the triggers, we crop the images according to the number of effective trading days they consider. Thus, the BB

images include information of 20 trade days, while RSI contains data for 27 days, and MACD, the most sophisticated algorithm that compares multiple time-scales, contains data of 26 days. Other words, each sample has 80-108 features depending on the size of the window that is required to compute the label (i.e., 4x20 for the BB crossing, and 4x26, 4x27 for the MACD and RSI respectively).
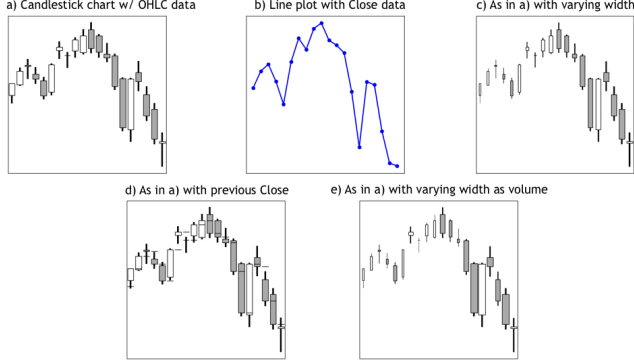

Figure 4: Various visual representations of the same time-series data.

Figure 4a-e show an example of the five different visual designs we use in this study. The design in 4a includes the OHLC data encoded as in Fig. 2, while panel 4b shows only the Close data as a line plot. A key point in this study is how to deliver the notion of time, or recency, in static images. Naively, we expect the trained classifiers to identify time-dependent *labels* and so should be able to deliver the notion of time explicitly or, at least implicitly, in the static images. The design of panels 4c and 4d aim on explicitly represent the direction of time by either linearly vary the width of the boxes towards the right (4c), or by overlaying the previous Close value as a horizontal line on each of the candlesticks (4d). Lastly, in panel 4e we add more information to the OHLC data by incorporating the trade volume in the candlestick visualization by varying the width of each box according to the relative change of the trade volume within the considered time frame. Remember that all three algorithms consider only the Close value, but this value is just one scalar conveying the last price per day, which is influenced by the previous daily volatility. We expect the trained classifiers to either filter out unnecessary information (i.e., noise) or discover feature relationships in the encoded image that will help to identify the label-generating rule.

Following the above process, we create high-resolution images based on the discretized form of the data. Another question we have to consider is what resolution do we need to keep for proper analysis. The problem is that the higher the resolution is, the more we amplify the feature space introducing more noise to the classification problem. We examine this point by varying the resolution of the input images in logarithmic scale and compare the accuracy score of a hard voting classifier over the following 16 trained classifiers[2]: Logistic Regression, Gaussian Naive-Bayes, Lin-

ear Discriminant Analysis, Quadratic Discriminant Analysis, Gaussian Process, K-Nearest Neighbors, Linear SVM, RBF SVM, Deep Neural Net, Decision Trees, Random Forest, Extra Randomized Forest, Ada Boost, Bagging, Gradient Boosting, and Convolutional Neural Net. Again, the focus of this study is not on finding the best predictive model but on comparing the aggregated performance of the models where we only change the representation of the input space.
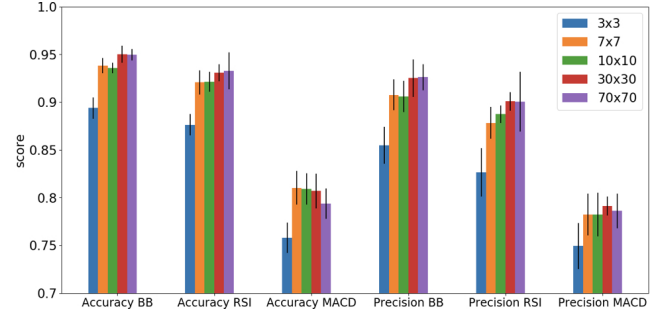

Figure 5: The effect of varying the image resolution on the classification accuracy and precision scores for the three label-generating rules.

Figure 5 shows the results of the classification scores when downscaling the resolutions of the images that are labeled following the BB, RSI, and MACD algorithms. For downscaling, we use the Lanczos filter, which uses sinc filters and efficiently reduce aliasing while preserving sharpness. To evaluate the model performance, we split and evaluate the hard voting classifier using the 5-fold cross-validation technique. This allows us to infer not only the mean prediction of the voting classifier but also the variability about the mean (the vertical black lines in Fig. 5 show one symmetric standard deviation about the mean prediction). Figure 5 shows that, regardless of the labeling algorithm, the averaged accuracy and precision scores go up with finer resolutions but matures around 30x30 grid resolution. For this reason, the following analysis is done using a 30x30 grid resolution.

Figure 6 compares the predictability skill in the various image representations of the same input data for the three label-generating rules. All input representations perform remarkably well, and the predictability skill stands at about 95% for the BB and RSI label-generating rules, while at approximately 80% for the MACD labels. We are not surprised to see that the classifiers perform less good on the MACD labeled data as this labeling-rule is the most complex involving four different time-scales smooth operations convolving each other and acting at tandem.

The best performing input data is the one that uses the Close values exclusively as line plots, while the various OHLC representations fall only little behind. However, the

---

[2]The Deep Neural Net uses 32x32x32 structure, while the Con-

volutional Neural Net (CNN) uses three layers of 32 3x3 filters with ReLU activations and Max Pooling of 2x2 in between the layers. The last layer incorporates Sigmoid activation. The CCN model is compiled with Adam optimizer, binary-cross entropy loss function and run with a batch size of 16 samples for 50 iterations
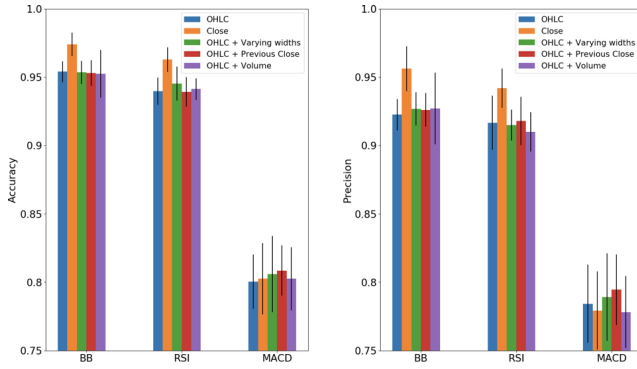
Figure 6: The supervised classification accuracy (left column) and precision (right column) scores for the various triggers as a function of the different input representations.

Close bars serve only as a point of reference – the Bayesian performance level. This is because the label-generating rule depends exclusively on the Close values[3]. The main focus is on the fact that the various OHLC representations manage the achieve performance almost comparable to the Bayesian level. Most importantly, this finding is robust for the BB and RSI, as well as for the MACD algorithm.

Close examination of Fig. 6 shows that modifying (augmenting) the OHLC input representation to include "better" time representation by linearly varying the bar widths or by incorporating the previous Close values was not so useful (i.e., didn't add value), except for, maybe, the MACD algorithm (a point we might explore further in a future study). On the other hand, encoding the irrelevant Volume information in the candlestick images, added to the uncertainty in the predictions for all label-generating rules.

The precision scores results in the right column of Fig. 6 are almost identical to the accuracy and, remember, that these can be tuned to perform better by optimizing the threshold level of the probability of the various binary classifiers.

Discussion

In this paper, we examine the supervised time-series classification task using large financial data sets and compare the results that are achieved when the data is represented visually in various ways. We find that even at very low resolutions (see Fig. 5), time-series classification can be resolved effectively by transforming the task into an image recognition problem. This finding is in accordance with (Cohen, Balch, and Veloso 2019) who concurrently showed that classification using special visual designs or smoothed downscaling relates far-apart data and reveal global information that aid the classifiers in identifying the driving pattern and achieve better performance comparing to the raw tabular form.

Visualizing data, in particular, time-series data is an essential pre-analysis step. Visualization by itself is not straightforward, especially for high-dimensional data, and it

might take some time for the analyst to find a proper graphical design that will encapsulate the full complexity of the data. In this study, we essentially suggest considering that display as the input over the raw data. Our research indicates that even very complex multi-scale algebraic operations can be discovered by transferring the task to an image-recognition problem.

A key question in this study is how can a time-dependent signal be detected in a static image? To be more explicit, if the time axis goes left to right, it means that data points to the right are more recent and therefore may be more critical to the classifier than data points to the left. But how can we convey that kind of information without an independent ordinal variable, i.e., in a static image? In principle, there are two ways to incorporate time-dependency in static images. One is to make the labels deliver the notion of time, and the second is by augmenting the images with sequential features. In this paper, we used both approaches. Incorporating time-dependency via labeling was done throughout the paper: The candlestick diagrams are labeled using the above algorithms where each computes a time-dependent function. Thus, each image via its corresponding label, implicitly, encapsulates the notion of time. Other words, the signal to be detected is located on the right-hand side of the image, as the cross-above triggers always occur because of the *last* few data points. In an example, the BB crossing algorithm effectively yields images with suspected local minima on the right-hand side of the picture. A trained classifier should be able to detect that kind of activity close to the bottom right corner of the image[4], but from time-series analysis perspective it is the local minima that convey the possibility for a mean-reverting opportunity. Incorporating time-dependency via image augmentation is considered in various ways: by linearly varying the width of the boxed in the candlestick diagram and overlaying the previous Close value on each candlestick. However, compared to the labeling approach, we find the augmentation to be less useful.

In this study, we blended all S&P 500 stocks and did not try to solve the classification problem per category or sector. We used specific window sizes corresponding to the length of information that the algorithms need to compute their labels. We also examined the classification results when all window sizes are of 30 days. Indeed, the performance goes down when the window size gets get longer and include unnecessary information, but we found this effect not to be dramatic (a few percents of a decrease. Not shown). We saw no need to account for the increase in the overall market performance during the last decade as our analysis is done on less than month-scale of variability. One can complement this study by similarly analyzing for sell signals. We have repeated this analysis for sell signals and found that the overall results are quite similar (not shown).

We end this paper by noting that the supervised classification task can be most efficiently applied as a forecasting tool (e.g., Hyndman and Athanasopoulos, 2018). In Fig. 7

---

[3]Using the Close value alone is comparable to using the actual numerical data that the labeling rules is based upon.

[4]i.e., tree-based algorithm should give higher importance to the grid points on the right-hand side over their corresponding grid points on the left-hand side
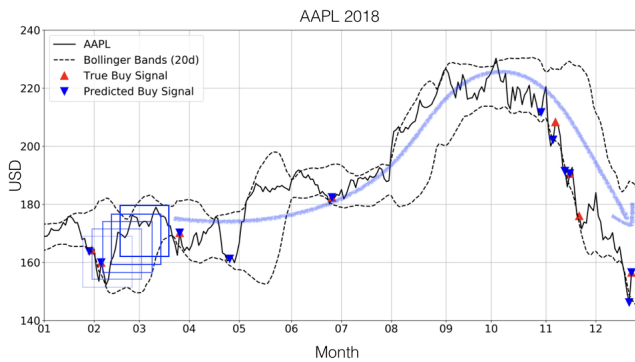
Figure 7: Time-series forecasting using a 20-days rolling window.

we take the daily trading data from 2018 (remember that the previous training and evaluation was done over data from the period between 2010 and the end of 2017) and create 20-days images for *every* day in the data. Then we feed these images to the tuned voting classifier, as a test set, and for each image, predict what the label is. Figure 7 corresponds to Fig. 3 but also include blue triangles showing the predicted buy signal. One can see that at least five buy signals were correctly classified, but even the missed ones are incredibly close in the sense that there is *almost* cross-above the lower BB. Depending on the use case, one can modify the binary probability threshold and achieve better precision scores.

Conclusion

Visual object recognition and object detection using machine learning and deep neural networks has shown great success in recent years (e.g., Krizhevsky, Sutskever, and Hinton, 2012; Szegedy et al., 2015; Zeiler and Fergus, 2014; Wang et al., 2017; Koch, Zemel, and Salakhutdinov, 2015; LeCun, Bengio, and Hinton, 2015). In this paper, we follow up on these studies and examine the value in transforming numerical time-series analysis to that of image classification. We focus on financial trading after noticing that human traders always execute their trades orders while *observing* images of financial time-series on their screens (see Fig. 1). Our study suggests that the transformation of time-series analysis to visual recognition is beneficial for identifying trade signals.

References

[2015] Aggarwal, C. C. 2015. *Data mining: the textbook*. Springer.

[2017] Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; and Keogh, E. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 31(3):606–660.

[2013] Berk, J.; DeMarzo, P.; Harford, J.; Ford, G.; Mollica, V.; and Finch, N. 2013. *Fundamentals of corporate finance*. Pearson Higher Education AU.

[2006] Bishop, C. M. 2006. *Pattern recognition and machine learning*. springer.

[2019] Cohen, N.; Balch, T.; and Veloso, M. 2019. The effect of visual design in image classification. *arXiv preprint arXiv:1907.09567*.

[1988] Colby, R. W., and Meyers, T. A. 1988. *The encyclopedia of technical market indicators*. Dow Jones-Irwin Homewood, IL.

[2018] De Prado, M. L. 2018. *Advances in financial machine learning*. John Wiley & Sons.

[2016] Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

[2018] Hatami, N.; Gavet, Y.; and Debayle, J. 2018. Classification of time-series images using deep convolutional neural networks. In *Tenth International Conference on Machine Vision (ICMV 2017)*, volume 10696, 106960Y. International Society for Optics and Photonics.

[2018] Hyndman, R. J., and Athanasopoulos, G. 2018. *Forecasting: principles and practice*. OTexts.

[2015] Koch, G.; Zemel, R.; and Salakhutdinov, R. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2.

[2012] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

[2015] LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature* 521(7553):436.

[1999] Murphy, J. J. 1999. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin.

[2019] Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

[2019] Pedersen, L. H. 2019. *Efficiently inefficient: how smart money invests and market prices are determined.* Princeton University Press.

[2013] Silva, D. F.; De Souza, V. M.; and Batista, G. E. 2013. Time series classification using compression distance of recurrence plots. In *2013 IEEE 13th International Conference on Data Mining*, 687–696. IEEE.

[2014] Souza, V. M.; Silva, D. F.; and Batista, G. E. 2014. Extracting texture features for time series classification. In *2014 22nd International Conference on Pattern Recognition*, 1425–1430. IEEE.

[2015] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

[2019] Tsai, Y.-C.; Chen, J.-H.; and Wang, C.-C. 2019. Encoding candlesticks as images for patterns classification using convolutional neural networks. *arXiv preprint arXiv:1901.05237.*

[2005] Tsay, R. S. 2005. *Analysis of financial time series*, volume 543. John wiley & sons.

[2015a] Wang, Z., and Oates, T. 2015a. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[2015b] Wang, Z., and Oates, T. 2015b. Imaging time-series to improve classification and imputation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[2017] Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3156–3164.

[2011] Wilks, D. S. 2011. *Statistical methods in the atmospheric sciences*, volume 100. Academic press.

[2014] Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.