

Financial Market Prediction Using Recurrence Plot and Convolutional Neural Network

Tameru Hailesilassie

National University Science and Technology NUST-MISIS
Department of Computer Science and Engineering
Moscow, Russia
tameruh.s@gmail.com

Abstract. An application of deep convolutional neural network and recurrence plot for financial market movement prediction is presented. Though it is challenging and subjective to interpret its information, the pattern formed by a recurrence plot provide a useful insight into the dynamical system. We used a recurrence plot of seven financial time series to train a deep neural network for financial market movement prediction. Our approach is tested on our dataset and achieved an average of 53.25% classification accuracy. The result suggests that a well trained deep convolutional neural network can learn a recurrence plot and predict a financial market direction.

Keywords: Time series, Deep learning, Convolutional neural network, Recurrence plot, Financial market prediction

1 Introduction

A time series is a sequence of values exhibited at a successive time points [1]. Time series data can be multivariant or univariant. A multivariant time series is an observation of one or more values at a time. Since our day to day life produces lots of sequential data, time series data can be easily obtained. They are vital in financial market analysis, clinical prediction (for instance, electrocardiogram signals (ECG), physiological signals (EEG)), and study of natural phenomena (for instance, weather readings). One of the most interesting approaches to business intelligence is a stock price prediction using financial time series. However, due to the volatile and nonlinear nature of the stock market, a financial market prediction is a complex problem. Besides, the high level of noise in stock time series worsens the problem.

Many traditional methods have been applied for stock market prediction. The two conventional approaches are technical analysis and fundamental analysis. Technical analysis relies on historical prices [2]. It makes use of a chart to study the pattern of past stock prices. Several stock price prediction methods have been developed on the ground of this approach. Its goal is to analyze the existing trend and predict the future direction of the stock market. Technicians analyze technical indicators derived from historical data. The most common technical

indicators are moving average, moving average convergence divergence (MACD), relative strength index (RSI), and on balance volume (OBV). On the other hand, fundamental analysis is information dependent [3]. Its aim is to gather information such as companies economic data, reports, and news that has an impact on the supply and demand to forecast the market (for instance, [4–6]).

Despite its complexity and prone to overfitting, numerous works have studied machine learning algorithms for financial market price prediction. Nowadays, traders use intelligent trading systems to help their trading investment decision. For this reason, algorithmic trading has drawn public attention. Consequently, there is a need for an algorithm that is efficient to predict market prices and movement direction. The potential problem of applying a machine learning algorithm for market prediction is that a machine learning model decision is a black box system and not easily understandable [7], yet traders need a comprehensible decision. Machine learning approach includes but not limited to an artificial neural network (ANN), support vector machine (SVM), and a hybrid model [8]. The feed-forward artificial neural networks have shown a good performance in multivariant time series prediction [9]. The use of ANN approach is well researched in stock price prediction and achieved a good result [10, 11]. Notably, ANN enhanced with advanced training algorithms are performing well [12]. Moreover, studies showed that SVM is a promising approach for stock market prediction [13, 14].

In the field of deep learning recently deep neural network (DNN) gain considerable popularity in the machine learning community. DNN has achieved a profound result in image and speech recognition [15]. However, the use of DNN for financial market prediction is not researched adequately. Xiao Ding et al. [16] proposed a deep learning approach for event-driven stock market prediction and achieved overall 6% improvement in S&P index and individual stock forecasting. Matthew Dixon et al. [17] have proposed a deep neural network method to classify financial market movement. Their approach is tested on 43 CME commodities and future mid-prices and achieved an average of 42% accuracy.

Research showed that classification based stock market prediction outperform the level based models [18]. Herein, we are motivated to train a deep convolutional neural network with a recurrence plot of financial time series to classify market movement direction (Up or Down). Recurrence plot is a technique to reveal and visualize a complex deterministic pattern in a dynamic system [19]. Recurrence plot and recurrence quantification analysis (RQA) are popular methodologies for a nonlinear and dynamic system [20]. They are able to confirm the change in a dynamical system. However, to the best of the author knowledge, there has been no attempt to train a machine learning model using a recurrence plot for algorithmic trading. Thus, we divide a historical financial time series into segments of multiple time frames using a fixed sliding window size and then generate a corresponding recurrence plot for each time frame data which we use to train our deep convolutional neural network (CNN). More recently, Dawei Song et al. [21] have used recurrence plots to train Convolutional

neural network for affective state recognition based on physiological signal and gained a 5% to 9% performance over the baseline.

The overall contribution of this work is to study the performance of deep CNN to learn a pattern from recurrence plot of financial time series. A single deep learning model is used to classify financial market movement direction. The output of the model is either 0 interpreted as down or 1 interpreted as up.

The rest of this paper is organized as follow. The following section describes the nature of data used in this work. The methods, Section 3, introduces the theoretical background of the methods we follow. Also, it describes the deep learning model architecture (in our case deep CNN). Section 4 provides experimental result measuring the performance of our model. Finally, section 5 gives a concluding remark.

2 The Data

This research work considered seven data signals as shown in Table 1. All data represents a daily prices. The closing price of IBM, General Electrics and S&P 500 is obtained from Yahoo historical data source by the address [22]. 54 years historical closing price is taken for both IBM and General Electrics as depicted in Figure 1 and Figure 2 respectively. Since the market is less liquid before 1981, 35 years price data is considered for S&P 500 index price. 48 years Gold price data is obtained from Deutsche Bundesbank Data Repository [23]. The source of the rest of experimental data is Qundle financial data repository [24].

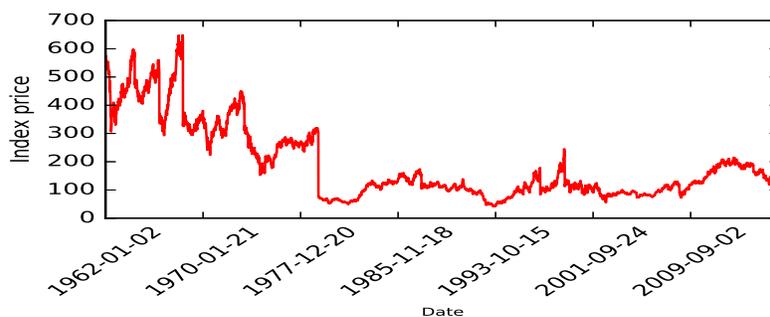


Fig. 1: IBM index price- 54 years historical data

The Standard and Poor's 500 composite stock price index known as S&P 500 and the world's largest information technology IBM are a well-known time series. The later is mentioned by Box et al. [25].

From autocorrelation and histogram plot in Figure 3 and Figure 4 we can see the nature of sample data from our experimental data set. As the value of lag increases, autocorrelation function coefficient drops to zero. These tells us

4 Tameru Hailesilassie

that the data points are highly correlated to each other. Thus, we can conclude that the signal is nonstationary. In fact, the majority of financial time series are nonstationary.

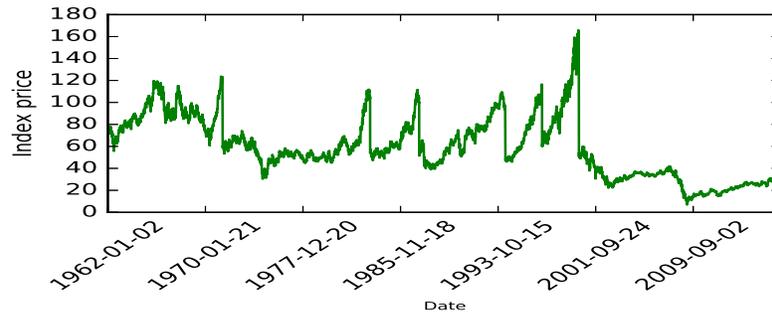


Fig. 2: General Electrics index price-54 years historical data

Table 1: The data used in the experiment

Time series data	Data Range	Number of data points
S&P 500 index price	1981-01-02 to 2016-12-09	9066
Copper price	1957-07-01 to 2016-12-15	14383
General Electrics Index price	1962-01-02 to 2016-11-22	13820
Soybean-oil price	1959-07-01 to 2016-12-15	14466
IBM index price	1962-01-02 to 2016-12-09	13832
Silver price	1983-12-15 to 2016-12-15	8320
Gold price	1968-04-01 to 2016-04-18	12152

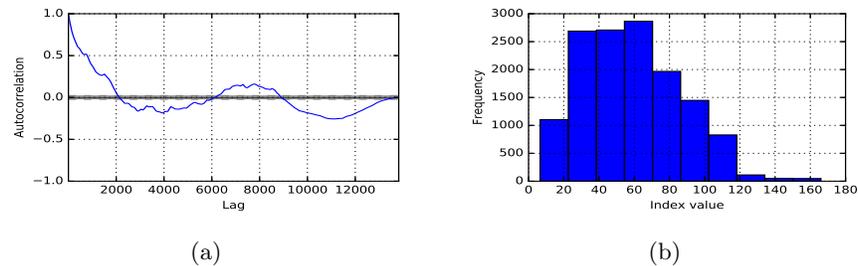


Fig. 3: General Electrics: (a) Autocorrelation plot (b)Histogram plot

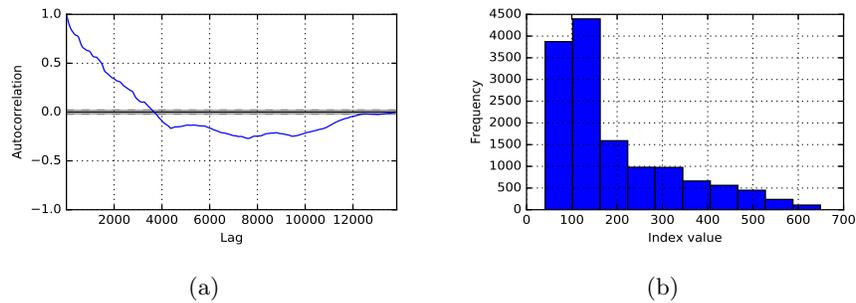


Fig. 4: IBM: (a) Autocorrelation plot (b)Histogram plot

The data set preprocessing includes segmentation of the raw signal into multiple time frames. Algorithm 1 depicted below is used to generate segments of the data and corresponding labels. The raw signal is segmented with a fixed sliding window and shifting size. Since the goal of this research is to predict a weekly market movement direction, a shift size of seven is employed. No further data preprocessing is applied to the raw signal.

Algorithm 1 Segment raw signal

```

1: procedure SEGMENTSIGNAL(signal, window, shiftSize)
2:    $i \leftarrow 0$ 
3:    $x \leftarrow \text{emptyList}$  ▷ segments
4:    $y \leftarrow \text{emptyList}$  ▷ Labels
5:    $n \leftarrow \text{signalLength}$ 
6:   while  $n \geq \text{window} + 1$  do
7:      $\text{start} \leftarrow i \times \text{shiftSize}$ 
8:      $\text{end} \leftarrow \text{window} + \text{start}$ 
9:      $x \leftarrow \text{signal}[\text{start} : \text{end}]$ 
10:     $y \leftarrow \text{signal}[\text{end} + 1]$ 
11:     $i \leftarrow i + 1$ 
12:     $n \leftarrow n - \text{shiftSize}$ 
13:  end while
14:  return  $x, y$ 
15: end procedure

```

3 Methods

Below, we present the methodology of this work. We discuss a recurrence plot and briefly explain how the training data set is generated. Furthermore, the proposed deep CNN architecture is presented.

3.1 Recurrence Plot

Recurrence plot is first introduced by Takens [26] in 1987 as a method to visualize recurrences of a dynamical system. A given time series data can be represented in its phase space using a method called time-delay embedding [27]. For a time series vector $x(t_i), i = 1, \dots, N$, time-delay embedding method can construct the attractor of the following lagged coordinate.

$$V(i) = [x(t_i), x(t_i + \tau), x(t_i + 2\tau), \dots, x(t_i + (m - 1)\tau)]. \quad (1)$$

Where, m is dimension of reconstructed phase space, and τ is the length of lag.

Selecting time-delay parameters τ and m is a non-trivial task. According to Taken's embedding theorem (1980) and other related theorems, $m > (2D + 1)$, where D is an observable dimension of the underlying system, provide an attractor that represent the same dynamical system as the original attractor in different coordinate systems. The brute force approach of choosing an appropriate lag length is a try and error. However, from literature review, a lag length $\tau = 1$ is found to be more suitable for financial time series [28]. There are a lot of approaches to determine the minimum sufficient embedding dimension (for instance, [29, 30]). The false nearest neighborhood is one of the algorithms to determine the minimum embedding dimension by measuring the percentage of false nearest neighbours [31]. The capable dimension is which the proportion of false nearest neighbors drop to zero for a given tolerance level. This work considered false nearest neighborhood to select an appropriate embedding dimension. Besides, the delay length of $\tau = 1$ is utilized. A recurrence matrix can be constructed from embedding vector as follow,

$$R_{i,j} = \begin{cases} 0, & \text{if } \|v(i) - v(j)\| > \epsilon \\ 1, & \text{if } \|v(i) - v(j)\| \leq \epsilon \end{cases}, \quad i, j = 1, \dots, n \quad (2)$$

Where $\|\cdot\|$ is an L2-norm and ϵ is a fixed threshold value. There are several rules of thumbs to select an appropriate threshold. In this work, after multiple experiments, we set a threshold value to be 5% of the maximal phase space diameter [32]. Figure 5 shows generated recurrence plots for segmented signals with a window size of 300 data points.

After the recurrence plot is produced for each segment, the next phase is treated as a classification problem.¹ The pattern formed by a recurrence plot provide a useful insight into the system. However, interpretation of this information is often challenging and subjective.

3.2 Convolutional Neural Network

Convolutional neural network (CNN) becomes an important model for image classification in machine learning because of several reasons. It is invariant for tilt, scaling or other forms of alteration in the input images. Besides, hand crafted

¹ The recurrence plot is generated using MATLAB CRP toolbox, www.recurrence-plot.tk

feature extraction, which is the tedious part of computer vision, can be accomplished by convolution layers in CNN. A sample convolutional neural network architecture is depicted in Figure 6.

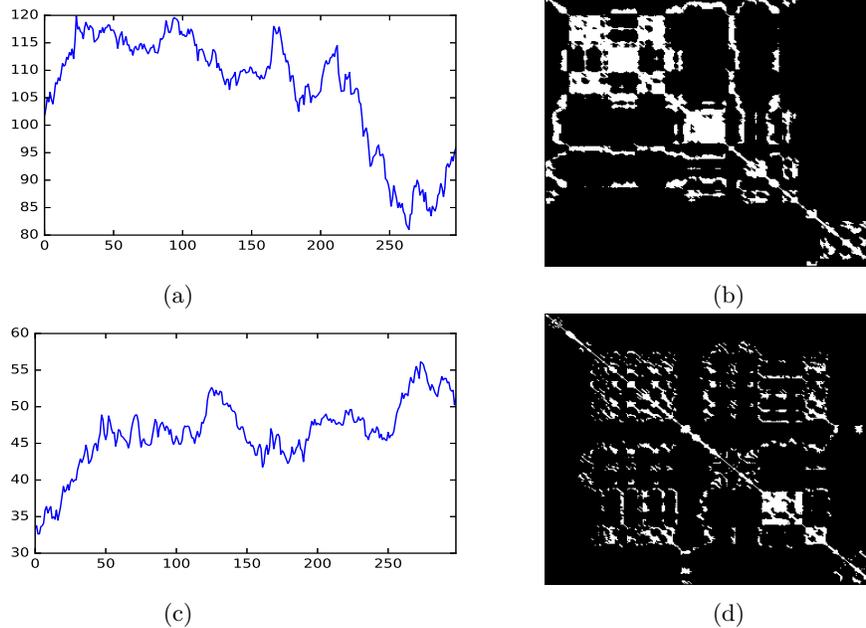


Fig. 5: (a)&(c): sample segmented signals.(b)&(d):corresponding Recurrence plots

A deep CNN comprises of several layers which contain a convolution layer followed by subsampling operation. The convolution operation is performed on the input image or feature maps from previous layers using a particular kernel (filter). Moving a filter across the image extracts a feature that is independent of the position of the feature in the input image. A pooling layer performs a subsampling operation to reduce the resolution of the feature map. The last layer in CNN is a fully connected layer (FC). This layer performs the final high label reasoning based on the features from the previous layer. The overall advantage of deep learning is, its ability to learn a non-linear and complex problem.

Training a deep network is a real challenging task [33, 34]. The challenge is not only its computational cost but a careful hyper-parameter optimization has to be conducted. Hyper-parameter optimization is a method to find an appropriate combination of parameters that gives the best validation performance. There are many available approaches for hyperparameter optimization such as a grid search, random search, and some optimized techniques [35]. However, parameter tuning is the most tedious stage of deep learning.

8 Tameru Hailesilassie

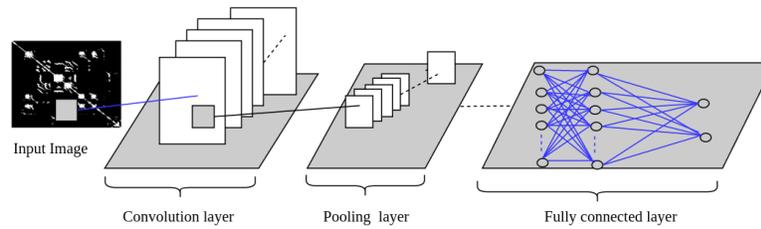


Fig. 6: Sample CNN structure

In this research, we proposed to use a deep CNN. The architecture of our model is summarized in Table 2. The convolution layers and the first two fully connected layers in this model have a *tanh* activation function as nonlinearity. *Tanh* activation function is chosen as a result of hyperparameter optimization. The objective function is *cross entropy*.

$$\tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \quad (3)$$

The final data set of our experiment contain a 64×64 pixel recurrence plot images and then was fed to the deep CNN.

Table 2: Proposed Deep CNN architecture: Size and Output represent (filter size \times number of filter), and (feature map size \times number of feature map) respectively

Layer	Size	Output
Conv1	$3 \times 3 \times 16$	$64 \times 64 \times 16$
Conv2	$3 \times 3 \times 16$	$64 \times 64 \times 16$
MaxPooling	2×2	$32 \times 32 \times 16$
Conv3	$3 \times 3 \times 32$	$32 \times 32 \times 16$
Conv4	$3 \times 3 \times 32$	$32 \times 32 \times 32$
MaxPooling	2×2	$16 \times 16 \times 32$
Conv5	$3 \times 3 \times 64$	$16 \times 16 \times 64$
Conv6	$3 \times 3 \times 64$	$16 \times 16 \times 64$
Maxpooling	2×2	$8 \times 8 \times 64$
FC	4096	
FC	1024	
FC	2	

4 Result and Discussion

Figure 7 summarize the performance of our approach, utilizing deep CNN and recurrence plot on financial market movement prediction. The average accuracy for considered seven data set is 53.25%. For S&P index price and gold price, a maximum of 56.62% and 56% accuracy is achieved respectively. We believe that the result shows a deep CNN is able to learn the complex pattern of a financial recurrence plot. As a first attempt to train CNN with recurrence plot, the result suggests that a well trained deep CNN can learn a recurrence plot and predict a financial market direction. Figure 8 further shows the filters of the first convolutional layers.

In this study, we trained a single deep CNN to classify the market movement direction using a generated recurrence plot from financial time series. A future work has to improve the classification performance of CNN. A deep learning requires a big data to converge. However, we have faced a lack of data in this study. Consequently, one can improve the model accuracy by utilizing different forms of data augmentation to increase the number of training set.

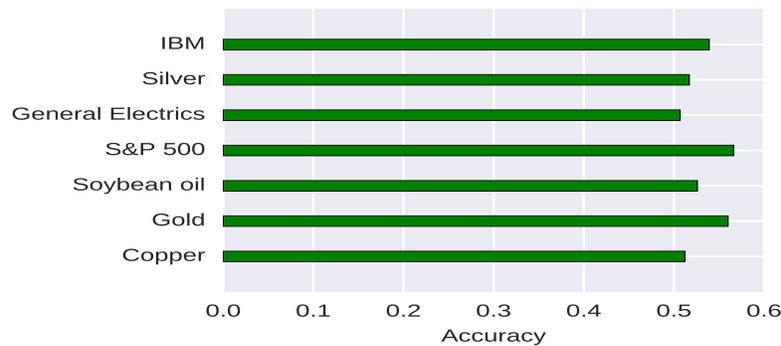


Fig. 7: Experiment result

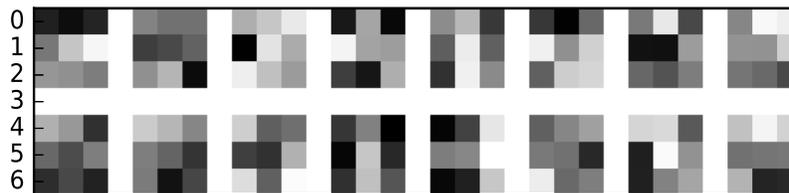


Fig. 8: Learned filters of the first convolution layer

5 Conclusion

In this study, we explored deep CNN and recurrence plot for a weekly financial market movement prediction. We segmented the time series data set and generated recurrence plot image to train a deep CNN. To the best of our knowledge, this study was the first attempt to train a deep neural network with a recurrence plot of financial time series. The system achieved an accuracy in the range of 50.68% to 56.62% in our dataset. We observed that a deep CNN can learn recurrence plot from historical price data and can make a market movement prediction.

Overall, this work has demonstrated the performance of deep CNN to learn a recurrence plot pattern and predict market movement direction. The future work will focus on improving the accuracy of the model. Besides, we will explore the performance to a large number of historical dataset.

References

1. Lee, S., Kwon, D., Lee, S.: Minimum distance queries for time series data. *Journal of Systems and Software*. 69, 105 – 113 (2004).
2. Pring, M.J.: *Technical analysis explained: the successful investor's guide to spotting investment trends and turning points*. McGraw-Hill, New York (1991).
3. Thomsett, M.C.: *Mastering fundamental analysis*. Dearborn Financial Pub., Chicago, IL (1998).
4. Yu, V., Lea, B.-R., Guruswamy, B.: A Theoretic Framework Integrating Text Mining and Energy Demand Forecasting. *International Journal of Electronic Business Management*. (2007).
5. Li, X., Wang, C., Dong, J., Wang, F., Deng, X., Zhu, S.: Improving Stock Market Prediction by Integrating Both Market News and Stock Prices. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., and Zhou, X. (eds.) *Database and Expert Systems Applications*. pp. 279 – 293. Springer Berlin Heidelberg (2011).
6. Schumaker, R.P., Chen, H.: Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System. *ACM Trans. Inf. Syst.* 27, 12:1 – 12:19 (2009).
7. Hailesilassie, T.: Rule Extraction Algorithm for Deep Neural Networks: A Review. arXiv:1610.05267 [cs]. (2016).
8. Abraham, A., Nath, B., Mahanti, P.K.: Hybrid Intelligent Systems for Stock Market Analysis. In: Alexandrov, V.N., Dongarra, J.J., Juliano, B.A., Renner, R.S., and Tan, C.J.K. (eds.) *Computational Science - ICCS 2001*. pp. 337 – 345. Springer Berlin Heidelberg (2001)
9. Chakraborty, K., Mehrotra, K., Mohan, C.K., Ranka, S.: Forecasting the behavior of multivariate time series using neural networks. *Neural Networks*. 5, 961 – 970 (1992).
10. Hamid, S.A., Iqbal, Z.: Using neural networks for forecasting volatility of S&P 500 Index futures prices. *Journal of Business Research*. 57, 1116 – 1125 (2004).
11. Chen, Y., Dong, X., Zhao, Y.: Stock Index Modeling using EDA based Local Linear Wavelet Neural Network. In: 2005 International Conference on Neural Networks and Brain. pp. 1646 – 1650 (2005).

12. Krollner, B., Vanstone, B., Finnie, G.: Financial time series forecasting with machine learning techniques: A survey. *European Symposium on Artificial Neural Networks: Computational and Machine Learning*. 25 – 30 (2010).
13. Kim, K.: Financial time series forecasting using support vector machines. *Neurocomputing*. 55, 307 – 319 (2003).
14. Cao, L., Tay, F.E.H.: Financial Forecasting Using Support Vector Machines. *Neural Comput & Applic*. 10, 184 – 192 (2001).
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. pp. 1106 – 1114 (2012).
16. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep Learning for Event-driven Stock Prediction. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. pp. 2327 – 2333. AAAI Press, Buenos Aires, Argentina (2015).
17. Dixon, M.F., Klabjan, D., Bang, J.H.: *Classification-Based Financial Markets Prediction Using Deep Neural Networks*. Social Science Research Network, Rochester, NY (2016).
18. Leung, M.T., Daouk, H., Chen, A.-S.: Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting*. 16, 173 – 190 (2000).
19. Bastos, J.A., Caiado, J.: Recurrence quantification analysis of global stock markets. *Physica A: Statistical Mechanics and its Applications*. 390, 1315 – 1325 (2011).
20. Fabretti, A., Ausloos, M.: Recurrence plot and recurrence quantification analysis techniques for detecting a critical regime. *Int. J. Mod. Phys. C*. 16, 671 – 706 (2005).
21. Yu, G., Li, X., Song, D., Zhao, X., Zhang, P., Hou, Y., Hu, B.: Encoding physiological signals as images for affective state recognition using convolutional neural networks. In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. pp. 812 – 815 (2016).
22. YHOO Historical Prices, <https://finance.yahoo.com/quote/YHOO/history>.
23. Statistics, <https://www.bundesbank.de/Navigation/EN/Statistics/statistics.html>.
24. Quandl, <https://www.quandl.com/>.
25. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: *Time Series Analysis: Forecasting and Control*. John Wiley & Sons (2015).
26. Eckmann, J.-P., Kamphorst, S.O., Ruelle, D.: Recurrence Plots of Dynamical Systems. *EPL*. 4, 973 (1987).
27. Takens, F.: Detecting strange attractors in turbulence. In: Rand, D. and Young, L.-S. (eds.) *Dynamical Systems and Turbulence*, Warwick 1980. pp. 366 – 381. Springer Berlin Heidelberg (1981).
28. Zbilut, J.P.: Use of Recurrence Quantification Analysis in Economic Time Series. In: Salzano, M. and Kirman, A. (eds.) *Economics: Complex Windows*. pp. 91 – 104. Springer Milan (2005).
29. Liu, S., Ma, R., Cong, R., Wang, H., Zhao, H.: A new approach for embedding dimension determination based on empirical mode decomposition. *Kybernetes*. 41, 1176 – 1184 (2012).
30. Zhao, L., Wang, L., Wang, Y., Huang, Z., Fang, N.: Determining Minimum Embedding Dimension in Short Time Series using Precisely Averaged False-nearest-neighbors Approach. In: *2008 China-Japan Joint Microwave Conference*. pp. 554 – 557 (2008).
31. Cao, L.: Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D: Nonlinear Phenomena*. 110, 43 – 50 (1997).
32. Schinkel, S., Dimigen, O., Marwan, N.: Selection of recurrence threshold for signal detection. *Eur. Phys. J. Spec. Top*. 164, 45 – 53 (2008).

12 Tameru Hailesilassie

33. Larochelle, H., Bengio, Y., Louradour, J., Lamblin, P.: Exploring Strategies for Training Deep Neural Networks. *J. Mach. Learn. Res.* 10, 1 – 40 (2009).
34. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10)*. Society for Artificial Intelligence and Statistics (2010).
35. Domhan, T., Springenberg, J.T., Hutter, F.: Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. pp. 3460 – 3468. AAAI Press, Buenos Aires, Argentina (2015).