Stock Turning Points Prediction Using Convolution Neural Networks with Return Rate Heat Map

Mariusz Rafało
Warsaw School of Economics
Warsaw, Poland *mrafalo@sgh.waw.pl*

Ryszard Szupiluk
Warsaw School of Economics
Warsaw, Poland *rszupi@sgh.waw.pl*

Abstract

A turning point in the stock market is the moment when the stock price time series changes direction. Its identification is important both for economic theory and, above all, for the practice of investing in the stock market. In this article, we propose a new concept, based on a broad representation of the market in the form of heat maps visualizing histograms of return rates. We then use convolution neural networks (CNN) to process these images. By extracting and analyzing image-based features, we can classify stock market events and effectively detect potential turning points. We conduct the study on 4 datasets, verifying 9 CNN architectures. The results are promising, we achieved accuracy of up to 97%.
Keywords: convolutional neural network, time series classification, stock index, deep learning

1. Introduction

A turning point (TP) is the point at which the time series of the stock market changes its trend direction. Typically, we distinguish ascending, descending and side trends. Trends can have different durations: short-term, medium-term and long-term trends. Turning points are important both for economic theory and practical prediction (e.g. for crisis forecasting). They are often associated with business cycles and market perturbations [9], [1]. The possible promising approach in identifying turning points is the use of convolutional neural networks [12].

Convolutional neural networks are a specialized class of deep learning models designed to process data with spatial or temporal structure, such as images or video streams [4]. The core concept of CNN involves applying convolutional operations where learnable filters slide over input data to capture local patterns or dependencies. These filters generate feature maps, which highlight regions of interest in the data by detecting patterns such as edges in images or temporal correlations. An important feature of CNN is their ability to share weights across the data, making them computationally efficient and reducing the number of parameters compared to fully connected networks [15].

In our approach, we assume that the representation of TP can be drawn from the image of sliding window of stock time series histograms at specific market. The input data to the CNN model are heat maps representing histograms of return rates on a given market in a selected time window. This means that even if our goal is to analyze one specific instrument, it is based on data from the entire market.

The aim of this work is to use CNN to identify and classify turning points in stock market financial data. Our concept is to implement a generic approach, in which heat maps of histograms can constitute a universal learning base for various tasks (not only TP prediction on financial markets).

Fig. 1. A single turning point (left) and turning region (right)

2. Identifying turning points

Identifying stock turning points can be treated as one of the stock market forecasting tasks, using patterns recognitions in market data [2, 21]. This approach involves analyzing historical data to look for patterns and then generalizing those patterns so they can be used for forecasting. From the analytical perspective, turning points, especially for long-term trends, can be associated with the theory of business cycles, often modeled using Markov chain approach [5]. Another method that is used to identify a TP is a piecewise linear regression, where we can divide the time series into segments containing TP [14]. An extension of this method is to link the identification of turning points with the variability of the time series [18]. Moreover, numerous studies use Bayesian models to classify TP [6].

Practical market methods of indicating a turning point include technical analysis on linear prices formations, candlestick charts approach, as well as intersections of long-term moving averages with short-term moving averages. Moreover, we can use signals generated by various indicators and oscillators such as RSI, ROC, MACD, etc. [13]. An interesting methodological aspect is the fact that image analysis plays the main role in technical analysis. This can lead to the assumption, that stock markets forecasting can be performed as part of pattern recognition tasks with images representing stock data [19].

Turning points can be also identified as a trend-change points, but some authors emphasize the difference between them, pointing out that every turning point is changing point but not all changing points are turning points [10]. However, due to complex and non-linear structures of financial data, many TP analyses tend to use systems that base on machine learning techniques [3]. One of the currently dominant methods in this area are convolutional neural networks with numerous applications in financial markets [16], [19]. The main challenge is how to develop image snapshots of financial markets so that they are useful for generalizing and identifying features.

We propose a transformation of histograms of returns rates to heat maps. This concept is motivated by several aspects. First, our objective is to capture as much information as possible about the market state in a given time window. To ensure comparability across assets, we focus on rates of return, which standardize the values. Second, by employing histograms, we eliminate the influence of data ordering—ensuring that the market representation remains invariant to the sequence in which individual stocks are arranged.

Heat maps also allow for better consideration of investment practice perspective, where the concept of a turning point is more flexible than in traditional econometric models. In some cases, a turning point may correspond to a specific moment in time (Figure 1, left panel), while in others it may be represented as a broader turning range (Figure 1, right panel), therefore pinpointing a single definitive point is challenging. A market turning point does not necessarily coincide with a local extreme but may occur in its vicinity. Consequently, the training set for TP prediction is constructed using a sliding time window that captures the formation of the turning point, rather than isolating a single time instance. We focus on broad market instruments, such as major stock indices, in order to represent overall market behavior using a single aggregated instrument.

3. Method

The concept of the research involves a structured pipeline for identifying and classifying market turning points. The pipeline works in several steps: (1) time series data is processed to extract instances of turning points as well as periods representing normal market behavior, (2) daily return rates are computed, (3) histograms of return rates are then transformed into heatmaps, (4) dataset is prepared with images that contain normal stock data behavior, as well as images containing data from 100 days before the TP, (5) classification models are trained to predict the TP 100 days before it occurs.

From technical perspective, the system consists of three nested loops: a data loop, a cross-validation loop and a model loop. The data loop involves running simulations on different data sets (different stock exchanges). We work on stock exchange data from Warsaw (WSE), London (LSE), Tokyo (TSE) and New York (NYSE), spanning from 1995 to 2023. Inside data loop, we run the cross-validation loop, that involves running CNN models on different training, validation and test datasets. Finally, inside cross-validation loop, we run the model loop, that involves training different CNN model architecture (we verify 9 CNN architectures).

To sum up, each of the 9 model architectures are trained on 100 epochs. Each training process is performed 100 times (number of cross-validation iterations), which gives a total of 3600 ($4 datasets \times 100 iterations \times 9 modelarchitecures$) CNN models. The system is implemented in Python programming language. It is controlled by parameters, which configure the simulation in various conditions. The source code is published in the author's public GitHub repository[1].

3.1. Data Loop

The data loop involves running the simulation process for each stock exchange dataset. The data is preprocessed to extract meaningful features, followed by transforming numerical data into image representations suitable for CNN models. For each stock exchange there is a file containing three columns: ticker (which is a short alphanumeric symbol used to represent the company on its respective stock exchange), date and close price for this date. The number of rows in each file is around 2000000 (depending on the number of tickers in the stock exchange).

Raw data undergoes preprocessing, which includes handling missing values and data standardization. Each dataset covers daily closing prices for publicly traded companies. Each company is uniquely identified by its ticker. For each stock exchange we manually indicate from 20 to 50 points in time in which turning points occurred. In addition, we include periods in the set where there were no turning points, so that the algorithm can distinguish turning points from permanent trends. These manual marks will then serve as the target variable in the training process.

For each of these TP, we then take 100 periods (days) before the target event occurs. During this period, we take time series for index tickers in a given dataset, which reflect the direction of the entire exchange in a representative way. Figure 2 presents the time series of the selected index along with the indication of the turning point (chart on the left). On the right, we present individual time series containing standardized closed values for individual tickers (gray lines) and the entire index (blue line). The right graph covers the period in the left graph, marked with vertical lines. We have indicated in red the period of the TP.

[1] https://github.com/mrafalo/cnn4stock

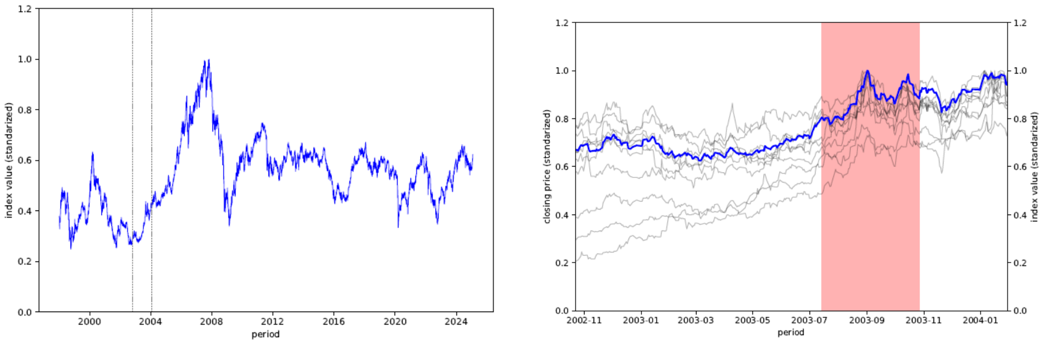

Fig. 2. Sample time series of the index (blue line) with turning point indication (red area). The left graph shows a wide time range, the right is a zoom in on the period containing TP.

The next step is to move from absolute price values to rates of return, for which we then determine histograms. To quantify the changes in stock prices over time, we compute the percentage rate of return between two consecutive days. In this way, for each period we determine the histogram of the rates of return for all tickers registered in that period. We put such histograms, period by period, as columns of the image. For each 100-day sliding window, we compute histograms for all tickers present in the period using 100 static bins and return rates ranging from -15% to 15%. If there is a missing value for a ticker in the rolling window under, then we assign the value from the previous day. Tickers that have more than 20% empty values are removed from the window. Sample histograms, stacked into preliminary heat map are presented on Figure 3, while examples of the final heat maps are presented in Figure 4.

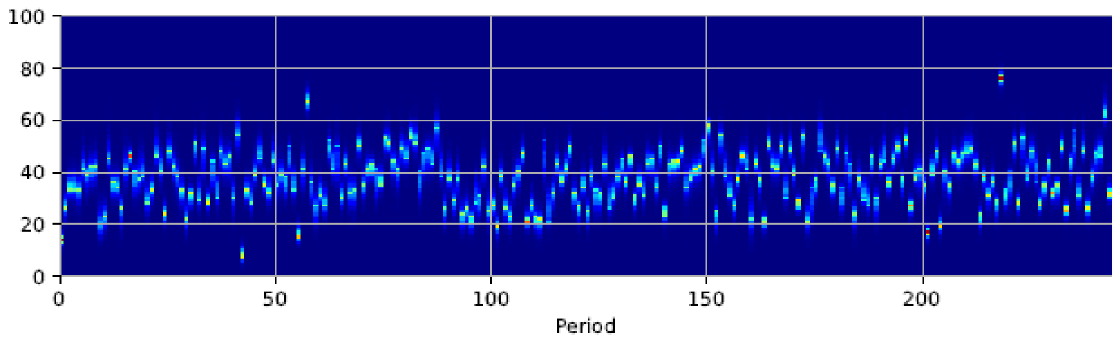

Fig. 3. Sample histograms stacked to initial form of a heat map

Histograms are transformed into images (heat maps) in the following way: a) the x-axis represents period in the window ((100 periods per window), b) the y-axis represents the bins (100 bins ranging from -15% to $+15\%$), c) the color intensity indicates the density of rates within specific bin and specific period. As a result, for a given time window we get a heat map image with dimensions of 100×100.

3.2. Cross Validation Loop

The cross-validation loop ensures that the model is trained, validated and tested in a way, that allows generalization and prevents overfitting. It begins with a random split of the dataset, where image representations of stock data are randomly partitioned into three subsets: training, validation and test. The training dataset is used to fit the CNN model, allowing it to learn patterns and extract relevant features. The validation dataset helps fine-tune hyperparameters by

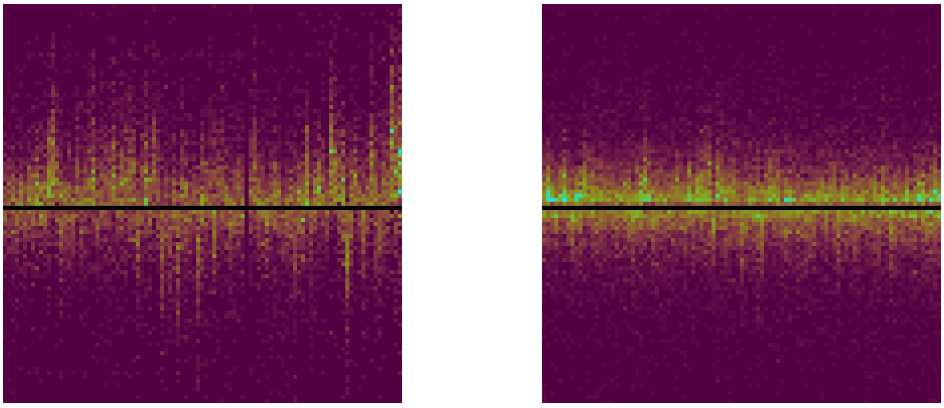

Fig. 4. Sample heat maps images containing target (turning point) sequence within 100 days

providing feedback on model performance without exposing it to the test data. The test dataset is held out until the final evaluation phase, ensuring that the model's predictive capability is assessed on unseen data.

For each dataset we perform 100 iterations of cross validation. In each iteration we randomly select images containing normal (not turning points) events and images for target values. In one dataset we randomly select 500 images, including 90 for target values (18%). The division of the dataset into the indicated subsets is done according to the following proportions: 75% of the data goes to the training set, 15% to the validation set and the remaining 10% to the test set. For images from the target class we apply minority oversampling technique, by image augmentation. Augmentation is done by shifting the target event from -5 to 5 days from the actual TP event date.

3.3. Model Loop

The model loop consists of two main stages: model training and model evaluation, with a feedback mechanism for hyperparameter tuning and optimization. We analyze the current state of common CNN models for time series classification as well as developing our own models from scratch. We compare 3 developed CNN models (cnn1, cnn2, cnn3) with ResNet models (*ResNet101* and *ResNet152*), VGG models (*VGG16* and *VGG19*) and DenseNet models (*DenseNet121* and *DenseNet201*).

The cnn1 model has 7 convolutional layers and is based on 130375 parameters, cnn2 has 6 layers and 183670 parameters, and cnn3 has 5 layers and 19871 parameters.

For each CNN model, we assess the metrics of sensitivity, specificity, precision, accuracy and area under Receiver Operating Characteristic (ROC) curve (AUC) measures.

4. Results

Table 1 presents average accuracy and AUC scores for all CNN models trained in the study, along with dataset indication. ResNet101 and ResNet152 achieve moderate accuracy across all datasets, performing slightly better on TSE and NYSE. VGG16 and VGG19 demonstrate lower classification accuracy, particularly on TSE, suggesting limitations in extracting relevant features from this dataset. Our CNN models exhibit varying performance, with cnn3 achieving the highest accuracy on all datasets. The DenseNet architectures outperform other models, except cnn3. The results indicate that deeper architectures such as DenseNet, VGG or ResNet are not always more effective. The cnn3 model is much simpler and turns out to be more effective, capturing complex patterns better than deeper networks.

The accuracy measure reflects the quality of the classifier well, but only when the data set

Table 1. Accuracy and AUC measures results for each CNN architecture and dataset

Model Name	Accuracy				AUC			
	WSE	LSE	TSE	NYSE	WSE	LSE	TSE	NYSE
ResNet101	0.61	0.71	0.83	0.83	0.56	0.69	0.84	0.85
ResNet152	0.61	0.59	0.81	0.83	0.56	0.53	0.79	0.81
VGG16	0.54	0.62	0.58	0.62	0.49	0.50	0.52	0.49
VGG19	0.57	0.56	0.64	0.61	0.51	0.50	0.50	0.50
cnn1	0.62	0.63	0.84	0.88	0.56	0.56	0.81	0.88
cnn2	0.56	0.62	0.82	0.86	0.47	0.55	0.79	0.83
cnn3	0.87	0.91	0.97	0.96	0.89	0.95	0.98	0.98
denseNet121	0.67	0.86	0.96	0.91	0.67	0.86	0.97	0.93
denseNet201	0.75	0.83	0.95	0.93	0.70	0.85	0.96	0.95

Source: own study

is balanced. If the number of target events is relatively small in the data set, accuracy will not reflect the quality of the model correctly. In such a case, a good approach is to base on the confusion matrix and determine the sensitivity and specificity measures and AUC curve. ResNet101 and ResNet152 exhibit moderate AUC values across all exchanges, with slightly better performance on TSE and NYSE. VGG16 and VGG19 show lower AUC values, indicating weaker classification performance. Custom CNN models demonstrate varied performance, with cnn3 achieving the highest scores. DenseNet architectures perform well on TSE and NYSE datasets.

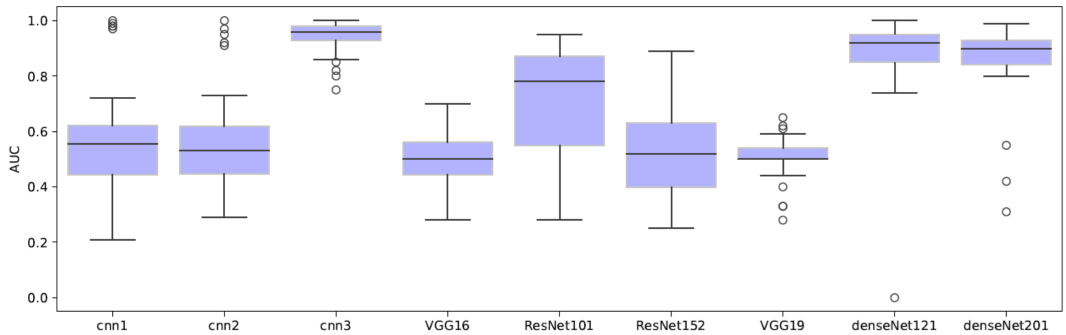

Fig. 5. CNN models results (AUC) for LSE dataset

The box plot 5 illustrates the variability of AUC scores for different models for LSE dataset. DenseNet models (DenseNet121 and DenseNet201) consistently exhibit high median AUC values with relatively low variance, indicating stable and reliable classification performance. cnn3 also shows strong performance, with consistently high AUC values and minimal dispersion. Conversely, VGG16 and VGG19 demonstrate the lowest AUC scores across all datasets, accompanied by higher variance and frequent outliers, suggesting less robust classification capability.

Relating our research to other studies in this domain, it can be indicated that our approach is novel and reports promising results. Early applications of CNN in time series classification, such as [20], demonstrate effectiveness in extracting features from raw data without manual preprocessing, outperforming traditional methods on various datasets. In finance, CNN models have been used for stock movement prediction [8] and trading signal classification [17], achieving up to 75% accuracy. Hybrid models combining CNN with LSTM have shown promising results in Forex prediction (accuracy: 75%) [11], while turning point classification using fuzzy segmentation [12] reports accuracy reaching 71%. However, the transformation of time series into images limits interpretability. To address this, [7] proposes an explainable dual-mode CNN (XDM-CNN) combining $1D$ and $2D$ CNN to enhance model transparency while maintaining accuracy.

5. Summary

Our method involves transforming stock price data into heat maps that represents return rate distributions. These visual representations serve as input for CNN models, which learn relevant features through convolutional operations. The study evaluates multiple CNN architectures, including ResNet, VGG, DenseNet and custom-designed CNN models. The model-building process follows a structured framework.

Experimental results show that cnn3 model achieves classification performance reaching average (for 100 iterations) prediction accuracy between 87% and 97%, with average AUC up to 0.98. The heat map representation provides an effective way to encode stock price movements, enhancing model performance compared to traditional feature-based approaches. The study also highlights that the effectiveness of CNN varies across stock exchanges, with certain datasets exhibiting higher classification variability.

The proposed method presents a novel approach to stock turning point prediction but also has limitations. The transformation of numerical data into heat maps introduces a degree of abstraction that may impact interpretability. Additionally, while CNN demonstrate strong classification capabilities, they do not provide explicit economic reasoning for detected patterns. Future research should explore hybrid models that integrate CNN with sequence-based architectures such as LSTM or Transformers to capture both spatial and temporal dependencies. Further work is also needed to enhance model explainability, potentially through attention mechanisms or post-hoc interpretability techniques. The study demonstrates that CNN, when applied to financial data in an image-based format, offer a promising direction for improving market prediction models.

References

[1] Ahelegbey, D. F., Billio, M., and Casarin, R.: Modeling turning points in the global equity market. In: *Econometrics and Statistics* 30.1 (2024), pp. 60–75.

[2] al., D. S. et: Stock market analysis: A review and taxonomy of prediction techniques. In: *International Journal of Financial Studies* (2019).

[3] Amini, S., Hudson, R., Urquhart, A., and Wang, J.: Nonlinearity everywhere: implications for empirical finance, technical analysis and value at risk. In: *The European Journal of Finance* 27.13 (2021), pp. 1326–1349.

[4] Bai, S., Kolter, J. Z., and Koltun, V.: An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. In: *arXiv:1803.01271* (2018).

[5] Bardaji, J., Clavel, L., and Tallet, F.: Constructing a Markov-switching turning point index using mixed frequencies with an application to French business survey data. In: *OECD Journal: Journal of Business Cycle Measurement and Analysis* 2009.2 (2010), pp. 111–132.

[6] Barry, D. and Hartigan, J. A.: A Bayesian Analysis for Change Point Problems. In: *Journal of American Statistical Association* 88.421 (1993), pp. 309–319.

[7] Cai, W., Zhu, X., Bai, K., Ye, A., and Zhang, R.: An explainable dual-mode convolutional neural network for multivariate time series classification. In: *Knowledge-Based Systems* 299.May (2024), p. 112015.

[8] Chen, S. and He, H.: Stock Prediction Using Convolutional Neural Network. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 435. 1. Chicago: Purpose-Led, 2018.

[9] Cho, H. and Fryzlewicz, P.: Multiple Change-point Detection for High-dimensional Time Series via Sparsified Binary Segmentation. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 77.2 (2015), pp. 475–507.

[10] Ghaderpour, E., Antonielli, B., Bozzano, F., Mugnozza, G. S., and Mazzanti, P.: A fast and robust method for detecting trend turning points in InSAR displacement time series. In: *Computers and Geosciences* 185.March (2024), p. 105546.

[11] Hadizadeh Moghaddam, A. and Momtazi, S.: Image processing meets time series analysis: Predicting Forex profitable technical pattern positions. In: *Applied Soft Computing* 108.September (2021), p. 107460.

[12] Khodaee, P., Esfahanipour, A., and Mehtari Taheri, H.: Forecasting turning points in stock price by applying a novel hybrid CNN-LSTM-ResNet model fed by 2D segmented images. In: *Engineering Applications ofArtificial Intelligence* 116.September (2022), p. 105464.

[13] Lim, M.: The Handbook of Technical Analysis. Wiley, 2015.

[14] Luo, L. and Chen, X.: Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction. In: *Applied Soft Computing* 13.2 (2013), pp. 806–816.

[15] Mallat, S.: Understanding deep convolutional networks. In: *Philosophical Transactions of the Royal Society A* 374.2065 (2016), p. 20150203.

[16] Sayavong, L., Wu, Z., and Chalita, S.: Research on Stock Price Prediction Method Based on Convolutional Neural Network. In: *2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*. Jishou, China, 2019, pp. 173–176.

[17] Sezer, O. B. and Ozbayoglu, A. M.: Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. In: *Applied Soft Computing Journal* 70.April (2018), pp. 525–538.

[18] Tang, H., Dong, P., and Shi, Y.: A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points. In: *Applied Soft Computing* 78 (2019), pp. 685–696.

[19] Tatane, K., Sahib, M. R., and Zaki, T.: From Time Series to Images: Revolutionizing Stock Market Predictions with Convolutional Deep Neural Networks. In: *International Journal of Advanced Computer Science and Applications* 15.1 (2024).

[20] Wang, Z., Yan, W., and Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: *2017 IJCNN*. Vol. 2017-May. IEEE, pp. 1578–1585.

[21] Yazdani, F., Khashei, M., and Hejazi, S. R.: A binary integer programming (BIP) model for optimal financial turning points detection. In: *Journal of Modelling in Management* 18.5 (2023), pp. 1313–1332.