Paper Replication: (Re-)Imag(in)ing Price Trends

LI Kaican[1], ZHAO Zening[2], LYU Yunhong[2] *
[1]Department of Computer Science and Engineering
[2]Department of Civil and Environmental Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, China
{klibf, zzhaocf, ylyuap}@connect.ust.hk

Abstract

This study aims to replicate and extend the findings of a recent work that proposed the use of convolutional neural networks (CNNs) for predicting future returns based on past price data. We successfully replicated the original results, confirming the efficacy of CNN models in predicting price trends. Additionally, we explored two extensions: testing the model's sensitivity in predicting price trends and providing Grad-CAM visualizations for interpretability. Our findings contribute to the understanding of machine learning applications in financial markets and highlight the potential of CNN models for guiding investment decisions.

1 Introduction

As financial markets continue to grow and investors seek more market information, the role of information in portfolio performance becomes increasingly crucial. Every day, investors are bombarded with various types of information, including historical data and new facts, which they need to process effectively to make better investment decisions. Understanding the predictability of stock returns requires analyzing investor behavior. One key factor in predicting stock prices is the relationship between past stock data and future returns. While traditional methods rely on statistical analysis, they have limitations in terms of time and analysis techniques. This is where machine learning comes in, as it can help overcome these limitations and provide more accurate predictions.

In a recent paper [1], they contribute to the field of predicting stock prices by proposing a new method that utilizes the "image" of stock prices combined with a trained Convolutional Neural Network (CNN). Traditionally, predicting stock prices relied on statistical analysis, but the CNN model offers several advantages over these approaches.

The first advantage is that the CNN model can automatically generate investment signals. By inputting an image of the price plot, the well-trained model can provide prediction results without the need for complex mathematical models. This automation saves time and simplifies the prediction process. Secondly, the image representation of data allows the model to visualize trends and relationships in a way that resembles human perception. This visual understanding can potentially improve the accuracy of predictions, as the model can capture patterns that might be difficult to detect using traditional methods. Lastly, the imaging process helps address the issue of non-stationarity in stock market data. By transforming historical data into a comparable scale, the model can better handle fluctuations and changes in market conditions, enhancing the reliability of predictions.

The goal of this report is to replicate and verify the findings presented by Jiang et al. [1]. We focus on one specific model from their paper, the 20-day model, and conduct our own analysis. Our

*The three authors contributed equally to this project during the whole process of coding, report writing and presentation preparation. Codes to reproduce the results and the video presentation are available at this link—*not* for acknowledging funding agencies.

Course Project, MATH 5470 Statistical Machine Learning (2024 Spring), HKUST.

Figure 1: Apple OHLC Chart from bar chart

replication results confirm their conclusions, and we also explore three additional experiments to further investigate the applicability of the proposed method. These experiments include Gradient-weighted Class Activation Mapping (Grad-CAM), assessing the model's robustness, and regression analysis.

The report is structured into four sections. Section 2 provides the background information, including the problem statement and details about the dataset of price trend images. In Section 3, we replicate the findings of the original paper. Section 4 explores various suggested extensions. Finally, Section 5 concludes the report, summarizing the key findings and suggesting future research directions.

2 Background

2.1 Problem statement

The core focus of this research is to predict future stock returns by utilizing historical price data with the aid of machine learning techniques, particularly Convolutional Neural Networks (CNNs). Given the complexity and noise inherent in financial data, traditional methods may fall short in capturing subtle patterns essential for accurate forecasting. Our study leverages the transformation of price and volume data into two-dimensional images, enabling the application of advanced image processing techniques. This approach not only allows the CNN to extract meaningful patterns from visual data but also harnesses the model's ability to interpret these images to predict whether stock prices will rise or fall over subsequent periods. The ultimate goal is to enhance prediction accuracy and thereby inform better trading strategies.

2.2 Data preprocessing

2.2.1 OHLC Chart Transformation

The dataset used in our models comprises OHLC (Open, High, Low, Close) charts, which are standard graphical representations in financial analysis showing the price movement within a particular day, the basic format of OHLC charts is shown in Figure 1. Each day in the chart is represented by bars and small horizontal lines; the vertical bar shows the high and low of the day, while the horizontal lines display the opening and closing prices. To accommodate the CNN architecture, these one-dimensional time-series data are converted into two-dimensional images.

For this transformation, each day is allocated a three-pixel width within the image: one pixel for the central vertical bar, and one pixel on either side for the opening and closing marks. These images are then scaled such that all have a uniform height, with the vertical axis aligned to the maximum and minimum prices portrayed at the top and bottom of the image, respectively. This consistent scaling is crucial as it ensures that each image reflects relative price movements accurately, making it possible for the CNN to learn from these visual patterns effectively.

2.2.2 Incorporating Additional Data

In addition to price data, our images also incorporate a moving average line and volume bars to provide a richer set of information. The moving average, calculated over the same number of days as the image width (e.g., 20 days for a 20-day image), is plotted as a line connecting dots in the center of each day's column, providing a smoothed reference of price trends over time. The volume data is represented as bars at the bottom fifth of the image, scaled relative to the maximum trading volume observed in the period, which helps in assessing the strength of price movements.

2.2.3 Handling Data Variability

Special care is taken to handle anomalies like missing data due to non-trading days or incomplete records, which are represented by leaving the corresponding pixel columns blank. This method of handling missing data ensures that the integrity of the temporal sequence is maintained without introducing artificial distortions. The background of these images is set to black, maximizing contrast and minimizing data storage requirements, as areas without data consume minimal space. This design decision aids in focusing the CNN's attention on significant data points and trends.

This preprocessing step transforms raw financial data into a structured format that is both efficient for storage and effective for pattern recognition by CNNs, setting the stage for accurate predictive modeling.

3 Paper replication

3.1 CNN architecture and training details

The CNN model used in this study employs a common CNN block that consists of a convolutional layer, an activation function (such as ReLU), and a pooling layer. The overall architecture of the network is depicted in Figure 2.

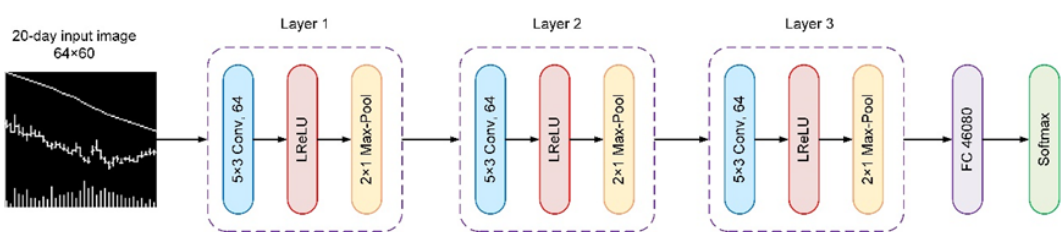

Figure 2: Overall architecture of the baseline CNN model for 20-day price trend images

3.1.1 Convolutional Layers

The CNN model includes a total of three convolution (Conv) layers. Each Conv layer utilizes 5×3 kernels to extract features from the input data. The three Conv layers generate 64, 128, and 256 feature maps, respectively. After the last Conv layer, the resulting feature map is flattened into a vector and passed through a fully connected (FC) layer. The FC layer employs the softmax activation function, which produces a probability distribution over the different classes. It allows the model to make predictions based on the learned features from the previous layers.

To avoid overfitting, the FC layer incorporates a dropout rate of 50%. Dropout randomly sets a proportion of the input units to zero during training, which helps prevent the model from relying too heavily on specific features and improves generalization.

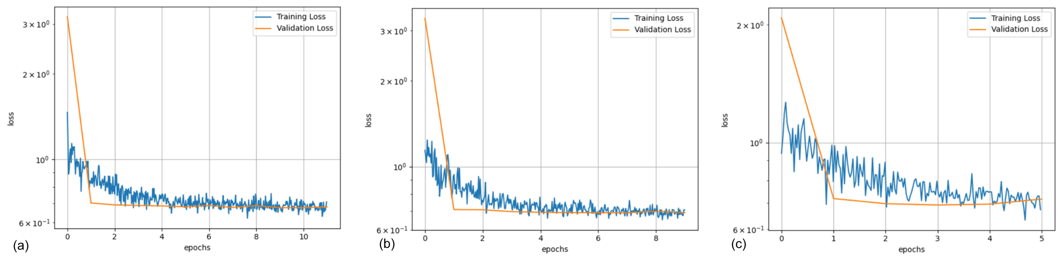

Figure 3: Loss curve of the baseline CNN on the (a) Pre5d task, (b) Pre20d task, (c) Pre60d task

3.1.2 Activation Function

After each Conv layer, a leaky ReLU (LReLU) activation function is applied. The LReLU introduces a small slope for negative input values, which helps prevent the learning process from getting stuck during training.

3.1.3 Pooling Layers

Following each Conv layer, a 2×1 max pooling layer is employed. The pooling operation reduces the spatial dimensions of the feature maps by combining the outputs of neuron clusters into a single neuron. This step effectively reduces the dimensionality of the feature maps.

3.1.4 Model evaluation

Since the goal is predicting the future returns (positive or negative), it is a binary classification problem. Therefore, the performance is assessed based on both the accuracy score and the F1 score. These criteria provide valuable insights into the model's effectiveness in predicting positive and negative returns and allow for a comprehensive evaluation of its performance. The accuracy score is calculated by dividing the sum of true positives (TP) and true negatives (TN) by the total number of predictions made. It represents the proportion of correct predictions out of all predictions and is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

where FP and FN are false positive and false negative, respectively.

The F1 score, on the other hand, is a harmonic mean of precision and recall. Precision is the ratio of true positives to the sum of true positives and false positives, while recall is the ratio of true positives to the sum of true positives and false negatives. They are calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

3.2 Experimental results

3.2.1 Training process

The training process is conducted using an Nvidia 4060 GPU. It takes approximately 25 minutes to complete each training epoch. The loss curves of the baseline CNN on the Pre5d task, Pre20d task, and Pre60d task are all shown in Figure 3. The loss function iteration curves of the CNN models demonstrate gradual convergence over about 6 epochs, eventually stabilizing at a significantly low

Table 1: Performance of the baseline CNN predictor over the whole dataset.

Task	Accuracy	Precision	Recall	F1-score
Pre5d	0.536	0.533	0.720	0.613
Pre20d	0.530	0.537	0.694	0.605
Pre60d	0.522	0.553	0.563	0.558

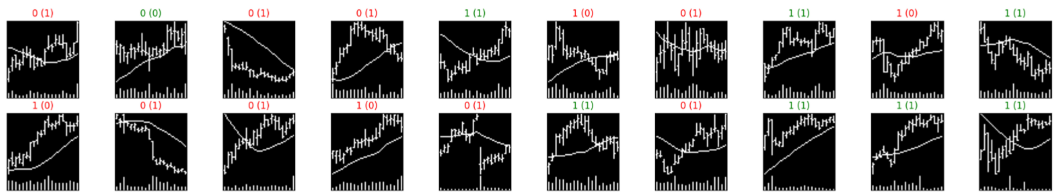

Figure 4: Predictions on the Pre20d task. Green/red indicates correct/wrong predictions respectively. The numbers inside parentheses indicate ground-truth labels.

loss. Throughout the training process, the loss steadily decreases with each epoch, indicating the model's ability to effectively learn from the data.

3.2.2 Performance over the whole data

Initially, we evaluate the outcomes using the entire test dataset spanning 19 years from 2001 to 2019, as presented in the Table 1. The results of Pre20d task are plotted in Figure 4. The accuracy-score and f1-score demonstrate close resemblance to the results reported in the original paper. Notably, even though the accuracy may appear modest, the CNN predictor has the potential to generate substantial profits for investors.

4 Extension

4.1 Sensitivity analysis of the CNN prediction model to other architecture

In this study, we aim to examine the sensitivity of model performance to variations in different aspects of model structure and estimation choices. Table 2 presents the sensitivity analysis of several metrics when different dimensions of the model structure are varied.

The results indicate that the performance remains relatively stable even with slight adjustments to the key parameters. This finding demonstrates the robustness of our replication models in capturing the underlying patterns and signals.

4.2 Interpretation via Grad-CAM visualization

In addition to improving the model's performance in predicting return trends, we are also interested in identifying the key features that contribute to the effectiveness of a machine learning model.

To visualize the model's attention to relevant input features when making "up" or "down" predictions, we utilize Grad-CAM. This technique generates heatmaps for each layer of the CNN, highlighting the regions of the input that are most influential in predicting a particular class. By plotting these activation heatmaps for individual observations, we gain insights into the aspects of the input that significantly influence the CNN's predictions.

Figure 5 displays the activation heatmaps for each of the three CNN blocks in our implemented model when predicting "up" or "down" labels. We found that the shallower layers, such as the first and second CNN blocks, primarily focus on the open and close prices. This is evident from the highlighted bright points in the heatmaps. Additionally, we noticed that the deeper layer, specifically the third CNN block, pays greater attention to the bars indicating price fluctuations.

These findings shed light on the specific input features that capture the model's attention during the prediction process, providing valuable insights into its decision-making mechanisms.

Table 2: Sensitivity analysis of model structure for Pre20d task.

	Variation	Accuracy	Precision	Recall	F1-score
Baseline	N/A	0.530	0.538	0.669	0.597
Filters (64)	32	0.527	0.540	0.601	0.569
	128	0.524	0.542	0.537	0.540
Layers (3)	2	0.523	0.539	0.566	0.552
	4	0.521	0.540	0.521	0.530
Dropout (0.5)	0	0.519	0.537	0.539	0.538
	0.25	0.508	0.544	0.329	0.410
	0.75	0.532	0.540	0.675	0.600
BN (yes)	no	0.524	0.541	0.548	0.545
Xavier (yes)	no	0.526	0.543	0.547	0.545
Activation (LReLU)	ReLU	0.530	0.539	0.665	0.595
Max-Pool Size (2×1)	(2×2)	0.530	0.544	0.580	0.562
Filter Size (5×3)	(3×3)	0.528	0.543	0.571	0.557
	(7×3)	0.515	0.547	0.390	0.455
Dilation/Stride (2,1)/(3,1)	(2,1)/(1,1)	0.526	0.543	0.546	0.545
	(1,1)/(3,1)	0.517	0.542	0.449	0.491
	(1,1)/(1,1)	0.525	0.543	0.539	0.541

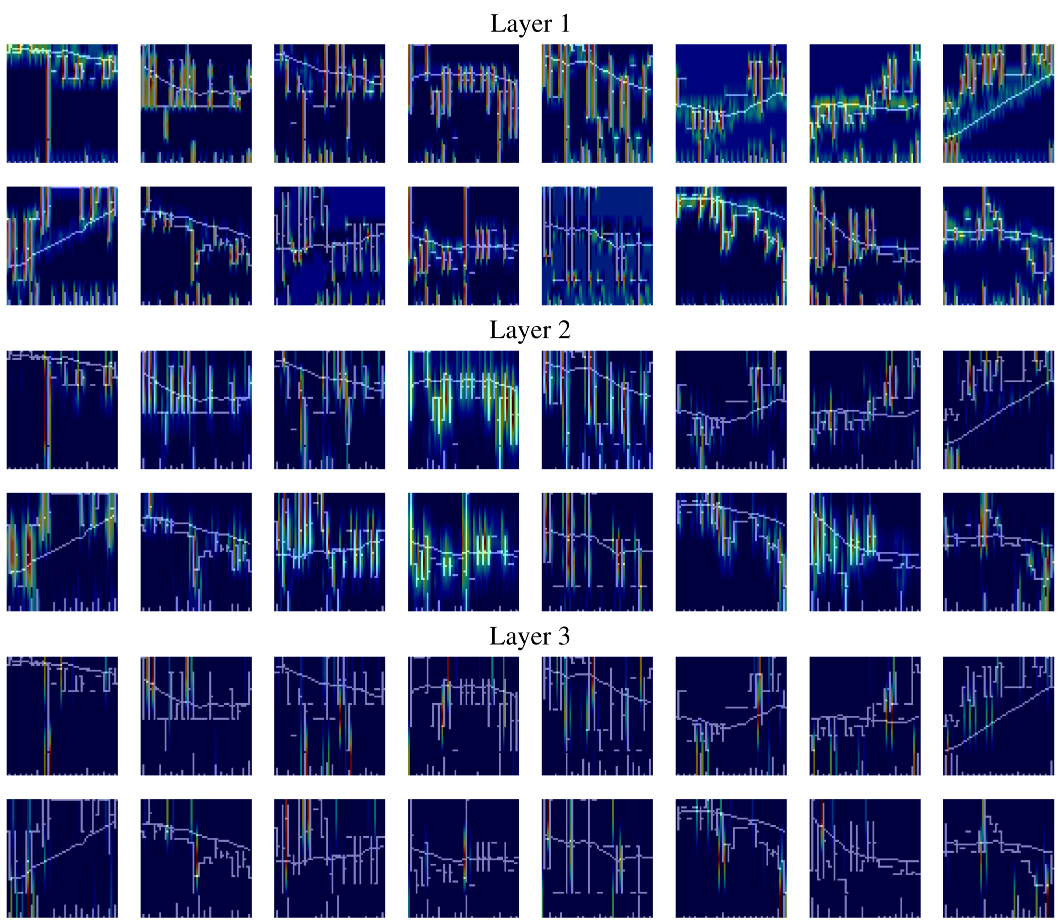

Figure 5: Grad-CAM visualization of model attention at different layers. The task is Pre20d and the predictions in the above examples are all correct.

5 Conclusions

This study aimed to replicate the findings of [1], which proposed the use of CNN models for financial market prediction. Our replication results align with the original findings, verifying the efficacy of CNN models in this domain. Additionally, we explored two extensions to their work: Grad-CAM and Robustness.

Our findings demonstrate that CNN models are not only capable of predicting the direction of market movements but also the magnitude of corresponding changes. By introducing additional structure to the final fully connected layer, we achieved improved model performance during testing. This is a significant contribution to our understanding of Machine Learning applications in the financial market, as it suggests that CNN-generated signals can guide valid investment decisions.

While our analysis focused on individual stocks, utilizing the CNN architecture to construct a diversified portfolio in the future could yield valuable insights and benefits.

References

[1] Jingwen Jiang, Bryan Kelly, and Dacheng Xiu. (re-) imag (in) ing price trends. *The Journal of Finance*, 78(6):3193–3249, 2023.