IMPERIAL

Innovative Approaches to Asset Prediction: Combining Deep Learning with Financial Modelling

Final Report

Author
C. Ioannidis
CID: 02533490

Supervised by
Prof. Pasquale Della Corte
Prof. Walter Distaso
Dr Lluis Guasch

A Thesis submitted in fulfillment of requirements for the degree of
Master of Science in Applied Computational Science and Engineering

Department of Earth Science and Engineering
Imperial College London
2024

Abstract

This project addresses the increasing complexity and volatility in financial markets through the development of advanced analytical tools. Leveraging the theoretical foundations established by Bryan Kelly and Kusuma, we propose refining Convolutional Neural Networks (CNNs) to enhance the prediction of financial asset behaviors.

Declaration of Originality

I hereby declare that the work presented in this thesis is my own unless otherwise stated. To the best of my knowledge the work is original and ideas developed in collaboration with others have been appropriately referenced.

Copyright Declaration

Contents

1 Introduction

Contents

1.1 Introduction

The prediction of financial markets has undergone significant evolution in recent years, driven largely by advancements in machine learning techniques. This study explores the use of Convolutional Neural Networks (CNNs) as advanced predictive models for analyzing time-series data across various asset classes. The primary objective of this research is to enhance the accuracy of financial market predictions by employing CNN architectures capable of capturing the complex patterns inherent in financial data. This project involves a comprehensive approach, beginning with the collection and preprocessing of extensive time-series datasets, followed by the development and iterative refinement of CNN-based models that address the unique challenges posed by financial market forecasting.

1.2 Background and Motivation

Traditional methods for predicting financial markets, such as the Auto Regressive Integrated Moving Average (ARIMA) model, have been widely applied but often struggle to capture the nonlinear and intricate dynamics of financial data. The advent of machine learning, and specifically CNNs, has introduced more sophisticated techniques capable of

addressing these limitations. Recent research has demonstrated the efficacy of CNNs in financial market prediction by transforming time-series data into visual formats that can better capture underlying patterns.

For instance, Kusuma et al. (2019) utilized CNNs to analyze historical stock data converted into candlestick chart images, achieving high prediction accuracy for stock markets in Taiwan and Indonesia [1]. Sezer and Ozbayoglu (2019) further advanced this approach by transforming time-series stock data into 2-D bar chart images and applying CNNs to identify trading signals, demonstrating superior performance to traditional methods, especially during bearish market conditions [2]. Zeng et al. (2021) expanded on these concepts by introducing a video prediction model for economic time series that leveraged CNNs' ability to detect spatial patterns in image sequences, outperforming traditional techniques like ARIMA and Prophet [3]. Jiang (2023) also employed CNNs with OHLC (Open, High, Low, Close) charts to forecast stock returns, highlighting the model's capacity to recognize intricate patterns and adapt across different geographical and temporal scales [4]. Collectively, these studies underscore the potential of CNNs to significantly improve asset price prediction accuracy, offering a marked advantage over traditional forecasting methods.

1.3 Research Objectives

The primary goal of this research is to enhance the predictive capabilities of financial market models through the development and refinement of CNN architectures. By focusing on the analysis of time-series data, this study aims to uncover complex patterns that drive market behavior, thus facilitating more accurate and reliable predictions. Achieving higher predictive accuracy is expected to support more effective risk management strategies, particularly in volatile market environments. Additionally, this research seeks to provide insights that can assist investors and financial analysts in making more informed decisions regarding investment strategies, thereby contributing valuable knowledge to the field of financial analytics. The overarching aim is to advance the adoption of more sophisticated machine learning techniques in financial forecasting, promoting a deeper understanding of global market dynamics.

1.4 Methodology Overview

The methodological approach of this study encompasses several stages. Initially, extensive time-series data will be collected and preprocessed to ensure it is suitable for deep learning applications. The data will then be transformed into image formats, such as candlestick charts, to facilitate the training of CNN models. Following data preparation, various CNN architectures will be developed and iteratively refined to optimize their predictive performance. The models will be trained using historical market data to learn patterns and trends that may inform future market movements.

In the final stages of the project, the developed models will be tested within simulated environments to assess their accuracy and practical applicability in real-world scenarios. This evaluation will be conducted through rigorous backtesting and benchmarking against

standard market indices, such as the S&P 500, to measure their performance relative to traditional investment strategies.

1.5 Expected Contributions

The anticipated outcomes of this research are expected to provide significant insights into market dynamics, thereby enhancing the decision-making processes in financial investments. By demonstrating the practical utility of CNNs in financial forecasting, this study aims to bridge the gap between theoretical advancements and their real-world applications in financial markets. The findings are expected to contribute substantially to the field of financial analytics, promoting the integration of advanced machine learning techniques into market prediction and portfolio management strategies. This research represents a meaningful contribution to the ongoing discourse on the application of deep learning in financial markets, with implications for both academic research and practical investment decision-making.

2

2 Methodology

Contents

This section outlines the technical approach used in developing and testing the Convolutional Neural Network (CNN) model for financial market prediction. The methodology is divided into several components: data acquisition and preprocessing, model development, evaluation and backtesting, and visualization and reporting. Each component is described in detail, highlighting the tools, libraries, and techniques used throughout the development process.

2.1 Data Acquisition and Preprocessing

2.1.1 Data Acquisition and Scope

The initial phase of the project involved the acquisition of high-quality financial time-series data from multiple sources, including the Center for Research in Security Prices (CRSP), Kaggle, and Yahoo! Finance. These datasets provided comprehensive OHLC

(Open, High, Low, Close) data across a wide range of securities listed on major U.S. stock exchanges, covering a period from the 1990s to 2017. The choice of this timeframe was deliberate to ensure that the training data was unrelated to the backtesting period (June 2019 to June 2024), thereby minimizing the risk of overfitting.

2.1.2 Data Preprocessing and Integration

Once acquired, the data underwent a rigorous preprocessing phase to ensure quality and suitability for deep learning applications. This phase involved several steps:

- **Data Cleaning**: Handling missing values using forward and backward filling techniques, and removing outliers using statistical methods such as Z-score analysis and interquartile range (IQR) filtering.
- **Normalization**: Standardizing the scale of OHLC data to ensure consistency across the dataset, typically scaling values between 0 and 1.
- **Transformation to Image Format**: The normalized OHLC data was converted into 64x64 pixel candlestick chart images using libraries such as Pandas, PIL, and Plotly. These images were then stored as .npy files for efficient loading and processing during the model training phase.

Simplified Pseudocode for Data Preparation:

```
Read CSV data files
For each data point:
    Clean and normalize data
    Convert OHLC data to candlestick chart image
    Save image as .npy file
```

2.2 Model Development

2.2.1 CNN Model Development

The core of the methodology focused on developing a robust CNN model using PyTorch, designed to predict financial market conditions based on visual representations of OHLC data.

- **Model Architecture**: The CNN architecture was constructed with multiple layers, including convolutional layers, pooling layers, dropout layers, and fully connected layers. The architecture was optimized to capture spatial patterns in the candlestick chart images.
- **Training Process**: The model was trained using GPU acceleration (CUDA) to handle large datasets efficiently. Hyperparameters such as learning rates, batch sizes, and epochs were carefully tuned to optimize model performance.

Simplified Pseudocode for CNN Model Development:

```
Define CNN architecture with multiple layers

Initialize model parameters
For each epoch:
    Load image data
    Forward pass through the network
    Compute loss and gradients
    Update model parameters
Save trained model
```

2.3 Evaluation and Backtesting

2.3.1 Model Evaluation

The CNN model, developed as a classifier, was evaluated using standard classification metrics to determine its performance in distinguishing between various market conditions:

- **Accuracy**: Proportion of correct predictions out of the total number of predictions.
- **Precision**: Proportion of true positive predictions out of all positive predictions made by the model.
- **Recall**: Model's sensitivity to correctly identifying all actual instances of a specific market condition.
- **F1 Score**: Harmonic mean of precision and recall, balancing the trade-off between these metrics.

2.3.2 Backtesting Strategy

Following model evaluation, a comprehensive backtesting strategy was employed to assess the CNN model's performance in a simulated trading environment. The strategy was implemented over a period from June 2019 to June 2024, aligning with the model's lookback periods and following a weekly rebalancing approach.

- **Trading Signals**: The CNN model generated signals to go long, short, or hold based on its predictions.
- **Backtesting Framework**: QSTrader was used to simulate trades and evaluate the strategy's performance against a benchmark buy-and-hold strategy of the S&P 500 index.

- **Performance Metrics**: Key metrics such as cumulative return, Sharpe ratio, maximum drawdown, and volatility were calculated to assess the effectiveness of the CNN-driven strategy.

Simplified Pseudocode for Backtesting Strategy:

```
Load trained model
For each backtesting period:
    Generate trading signals (long, short, hold)
    Execute trades in simulated environment
    Calculate performance metrics (return, Sharpe ratio, drawdown)
Compare performance to buy-and-hold benchmark
```

2.4 Visualization and Reporting

- **Rationale**: To provide a clear and comprehensive visualization of the model's performance and the results of the backtesting strategy.
- **Implementation**: Utilized Matplotlib to generate plots and charts illustrating key performance metrics and outcomes.

Simplified Pseudocode for Visualization:

```
Collect performance metrics
Generate visual plots for cumulative return, Sharpe ratio, drawdown
Display comparison graphs
Export visual reports
```

2.5 Technical Platform and Implementation Details

The implementation of the solution was conducted entirely using Python, leveraging a range of open-source libraries and frameworks to facilitate various aspects of data processing, model development, and evaluation. The development environment primarily utilized Visual Studio Code (VSCode), which provided a robust platform for writing, testing, and organizing the Python scripts into modular files. This modular approach allowed for the separation of different functional components, such as data preparation, model training, and backtesting, thereby enhancing maintainability and facilitating iterative development.

For data processing and visualization, libraries such as Pandas, PIL, Plotly, and Matplotlib were employed. Pandas was utilized for data manipulation tasks, including reading CSV files and handling missing values, while PIL and Plotly were used to generate candlestick chart images from the processed OHLC data. Matplotlib was also employed to create additional visualizations, both during the exploratory data analysis phase and for presenting the final results.

The core of the model development was implemented using PyTorch, a popular deep learning framework known for its flexibility and support for dynamic computation graphs. PyTorch, in combination with CUDA, enabled efficient training of the CNN model on GPUs, significantly accelerating the computation and allowing for more extensive hyperparameter tuning. NumPy was also utilized for numerical computations, providing efficient array operations and integration with other libraries.

For the evaluation and backtesting of the CNN model, QSTrader, an open-source framework for systematic trading strategies, was used. QSTrader facilitated the simulation of trading strategies based on the model's outputs and enabled a comprehensive analysis of the model's performance against a benchmark buy-and-hold strategy of the S&P 500 index. This integration allowed for a rigorous assessment of the model's predictive capabilities in a controlled environment, replicating real-world trading scenarios.

The development process was initially carried out within Jupyter Notebooks to allow for interactive experimentation and visualization of results. Upon achieving satisfactory performance, the code was then refactored into standalone Python scripts for a more production-ready deployment. This transition ensured that the final implementation was optimized for operational use while retaining the flexibility and modularity required for further enhancements and testing.

Overall, the choice of tools and frameworks was guided by their suitability for handling large-scale financial data and their ability to support the iterative development of deep learning models. The combination of Python's ecosystem of libraries with a modular development strategy provided a robust and scalable solution, capable of addressing the complex challenges associated with financial market prediction using CNN architectures.

3 Results

4 Discussion

5 Conclusion

Bibliography

[1] R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua, *Using deep learning neural networks and candlestick chart representation to predict stock market*, 2019. arXiv: 1903.12258 [q-fin.GN].

[2] O. B. Sezer and A. M. Ozbayoglu, *Financial trading model with stock bar chart image time series with deep convolutional neural networks*, 2019. arXiv: 1903.04610 [cs.LG].

[3] Z. Zeng, T. Balch, and M. Veloso, *Deep video prediction for time series forecasting*, 2021. arXiv: 2102.12061 [cs.CV].

[4] J. Jiang, B. Kelly, and D. Xiu, "(re-)imag(in)ing price trends," *The Journal of Finance*, vol. 78, no. 6, pp. 3193–3249, 2023. DOI: 10.1111/jofi.13268. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jofi.13268. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jofi.13268.