Summary

Computer Vision for Financial Time Series

Mohamad Ahmad

Abstract—Convolutional Neural Networks (CNNs) possess a unique method of *viewing* data. The underlying kernels are able to stride through the input matrix, processing data in many directions. For this reason, the author posits that there could be value in using CNNs to classify financial outcomes using time series data formatted into a matrix.

Keywords—*Computer Vision, CNNs, Deep Learning, Time Series, Monte Carlo Simulation.*

1. Introduction

CNNs are known to stride over an input matrix and perform the convolution operation. This stride mechanism allows them to detect prominent features which have been translated in space. This method of viewing "patches" of data could be utilized to analyze time series data. In this application, the input matricies are of dimension 20×6, representing 20 days of 6 temporally linked financial time series.

2. The Data

2.1. Training Data

Through the yfinance library, 6300 instances of daily AAPL data (1990-01-02 → 2014-12-30) are used for the training set, Tr. The $K = 6$ features of this data are **open, high, low, close, volume, and 14d momentum**. The binary label for each instance is if the open price of the next instance will be lower than the close price of instance 10 market days away. A window of size $H = 120$ is scanned over Tr to yield $|Tr| = 6181$ $H \times K$ matricies and corresponding labels. Each matrix is **min-max scaled** down to yield final training inputs

$$X_{Tr,i} \in [0,1]^{H\times K}, \quad y_{Tr,i} \in \{0,1\} \quad i \in \{1,\dots,|Tr|\}$$

The start and end dates of each X are stored in a data structure for future access.

2.2. Out of Time Data

The test data, Te spans 2033 instances of daily AAPL data (2016-01-04 → 2024-01-31). The same window and scaling operations are performed on Te to yield $|Te| = 1914$ inputs in $[0,1]^{H\times K}$ and outputs in $\{0,1\}$. Hence, we have

$$X_{Te,i} \in [0,1]^{H\times K}, \quad y_{Te,i} \in \{0,1\} \quad i \in \{1,\dots,|Te|\}$$

3. Methodology

3.1. Matching

Given some testing datum $X_{Te,i}$, we have access to the unit-interval-scaled close price represented by the third column of such a matrix, $C_{Te,i} \in [0,1]^H$. For each $C_{Te,i}$, we scan through all $|Tr|$ columns of close price in the training set $C_{Tr,j}$ to find

$$j^* = \underset{j\in\{1,\dots,|Te|\}}{\arg\min} \quad d(C_{Te,i}, C_{Tr,j})$$

where $d(\cdot,\cdot)$ is the standard Euclidian distance. We are able to find the start and end dates encompassing this time of most similar scaled price-action.

3.2. Model Training

Given a historical window of similar price-action, we gather the K scaled features from the start of this period to 50 days following it. Given this matrix, we slide a window of size $W = 20$ days over it to yield instances in $[0,1]^{W\times K}$. Given these instances (approximately 145 of them), we train a CNN with residual skip connections with ADAM.

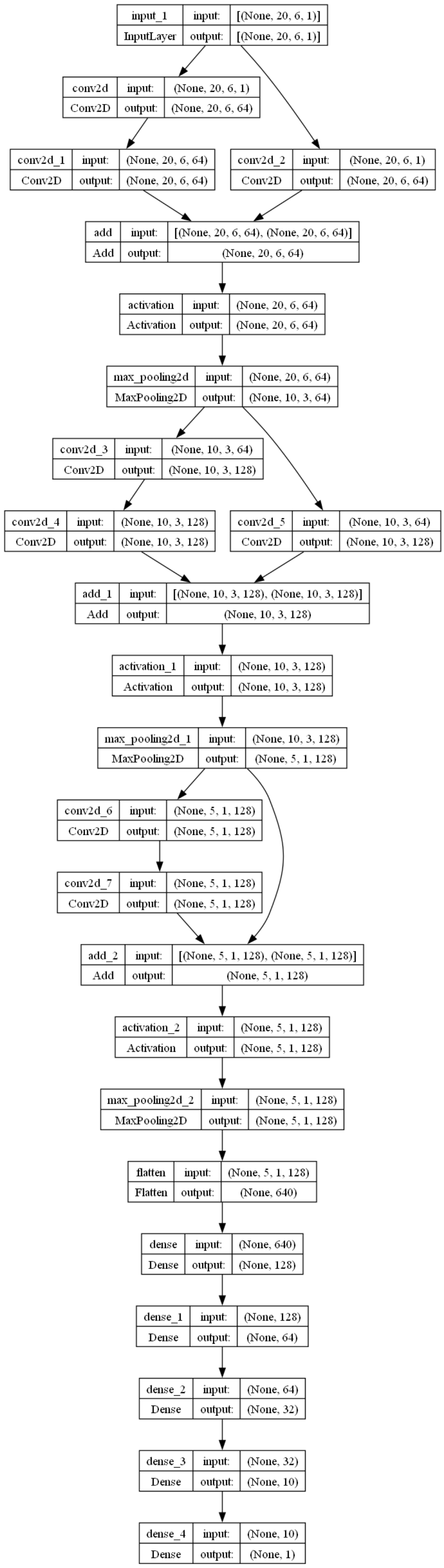

Figure 1. CNN model architecture.

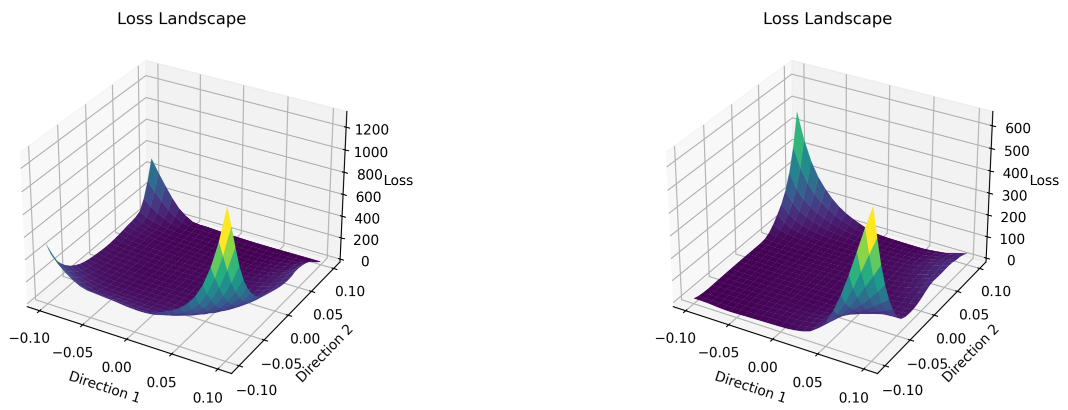

Figure 2. Loss landscapes observed with 2 different CNN models.

The resulting loss landscapes appear to possess a convex surface when directions in weight space are perturbed.

3.3. Model Prediction

The algorithm concludes with inputting the last 20 rows of $X_{Te,i}$ into the model and yielding some $\pi_i = P(y_i = 1|X_{Te,i}) \in [0, 1]$. The π_i is fed into a decision function which depends on some threshold T.

$$D(X_{Te,i}, T) = \begin{cases} 1 & : \pi_i > T \\ 0 & : \pi_i \le T \end{cases}$$

The models in this paper used $T = 0.9$. Such a threshold aims to yield high precision but will undoubtedly have a poor effect on recall. The emphasis on keeping a high precision is to mitigate the potentially lossy (no stop-loss) trades that the model will be tasked with opening.

Assumptions and Concerns

We assume that historically similar times of price action have predictive power in the present.

We assume that the multi-layer breakdown of the market structure will be able to capture information on both market macro-structure $H = 120$ and micro-structure $W = 20$.

It is a concern that $W = 20$ with small training sets ≈ 145 instances will result in **over-fitting.**

It is a concern that $T = 0.9$ will result in this algorithm missing out on many good trades solely due to fear imposed by **uncontrolled risk measures**.

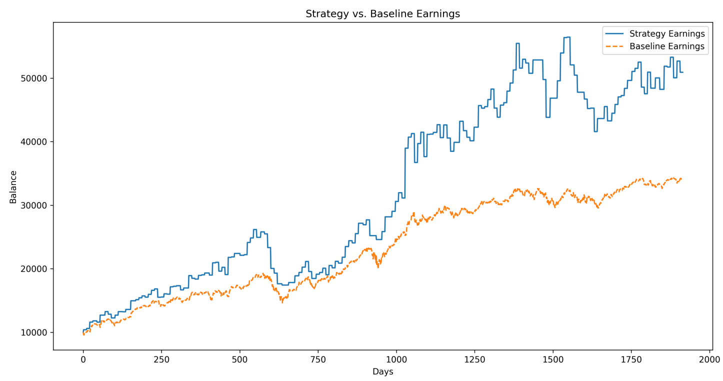

Figure 3. Strategy vs. baseline (buy and hold) earnings.

The backtest results yield a precision of 66% (vs. baseline balance 63%) and an expected value of 1.2% per trade. The resultant performance metrics suggest the possibility that such a model could be profitable.

4.1.1. Monte Carlo Simulation

While backtesting on the real historical market proved to generate some positive alpha, it is of interest to simulate how such a strategy would have performed on different market environments. For this reason, a Monte Carlo simulation was run to generate alternate equity curves and find a distribution for the final balance.

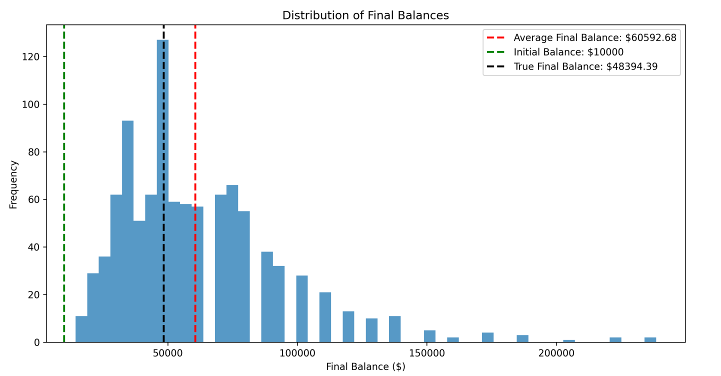

Figure 4. Monte Carlo simulation results.

4. Backtesting

4.1. AAPL Stock Performance

Given some $X_{Te,i}$, if we observe $D(X_{Te,i}, 0.9) = 1$, we open a long position on AAPL at the next day's market open. We close this position 10 market days later. Backtesting this strategy over the out of time sample yields profitable results.

4.2. AAPL Options Performance

It is of interest to apply this strategy to AAPL options. The backtest begins with a balance of $\$10,000$ USD. Rather than opening a long position at the next market open, this backtest will open a number of call contracts at the next market open and sell them 10 market days later. The call contracts are 20 days 'til expiry at the money contracts. Position sizing is done to risk at most 5% of the account.

Assumption and Concern

We assume that there will be sufficient liquidity to purchase 5% of the account portfolio in call options at any given time.

This assumption needs to be checked and is likely not true. It is suggested to have a maximum number of contracts that can be filled. Conservatively, this quantity should not exceed the mean number of available ATM 20 DTE contracts observed in the options dataset.

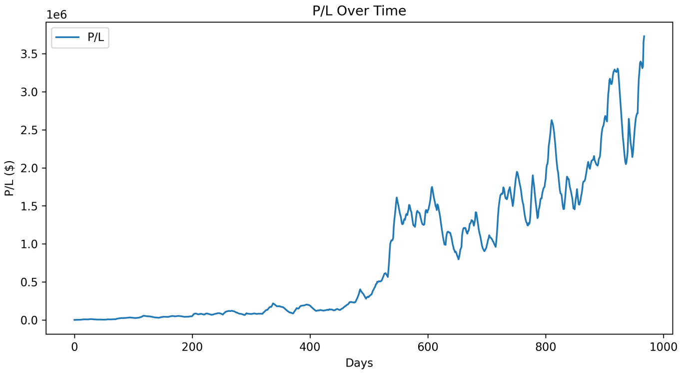

Figure 5. Options strategy P/L results (log scale).

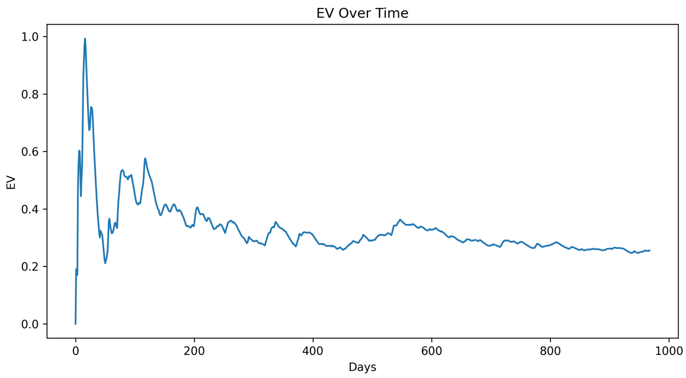

Figure 6. Options strategy average EV per trade over time.

Over the span of the testing set, the strategy was able to grow $10,000 USD to $3.5M USD. The average rolling EV during the backtest seems to stabilize at 26%. A precision of 52% was obtained, with an average return of 102% for correct trades and -55% for incorrect trades.

4.2.1. Confidence Interval Computation

The testing period was partitioned into blocks by month. The strategy was applied to each monthly block to generate samples of EV and precision observations. This methodology was used because the sampling distribution of these metrics is undoubtedly changing by the market structure constantly shifting.

The consequent 95% confidence intervals of this strategy over one month of deployment are as follows:

$$\text{EV per trade:} \quad [9.7\%, 41.1\%] \quad | \quad \text{Precision:} \quad [43.5\%, 57.2\%]$$

4.2.2. Kernel Density Estimation

Given the partitioning of the test data, we are able to produce a kernel density estimate (KDE) for the underlying distribution of the ROI over any given month.

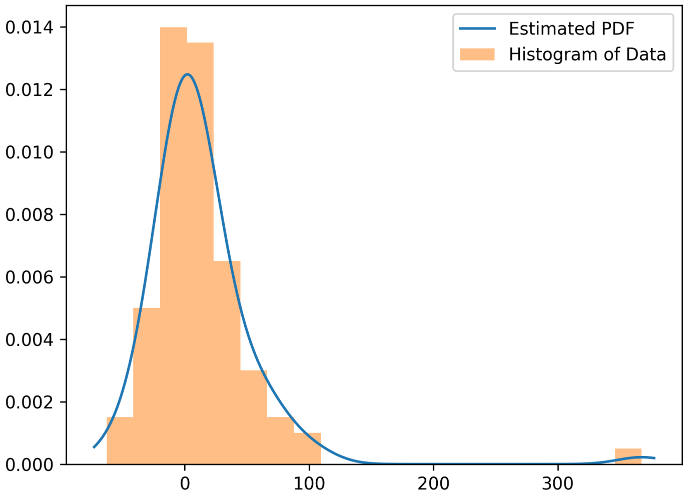

Figure 7. KDE for ROI(%) per month.

Concern

There are outliers observed in the KDE. Furthermore, the P/L over time appears to rapidly pick up at roughly the time of 2020. This abrupt shift in market structure is not stable and it is unwise to assume that this will ever happen again. The testing range should be adjusted to end prior to madness ensuing in the retail options market.

4.2.3. KDE Drawn Simulations

With the KDE, samples for ROI by month are able to be simulated. This allows for the possibility of simulating many different outcomes over any desired timespan.

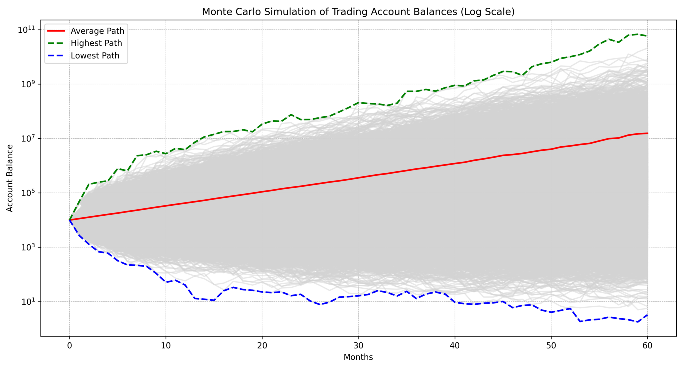

Figure 8. 60 month simulations drawn from KDE.

The simulation results show that 88% of paths end up profitable, while 12% do not. The variance in final balances, $\sigma^2 \to +\infty$, with a 95% confidence interval for the mean final balance having a very large range

$$\text{Mean Final Balance:} \quad [9.9 \times 10^6, 2.2 \times 10^7]$$

Concerns

The confidence intervals for simulated results are being biased upwards. Consider providing confidence intervals for the median instead. Furthermore, these simulations fall prey to the risky assumption that there are enough call contracts to purchase.

5. Reduced Volatility Backtesting

By bounding our training set to end before 2020, we hope to sample less volatile results.

5.1. AAPL Stock Performance

While we still observe a higher end balance than baseline, there is a marked reduction in alpha.

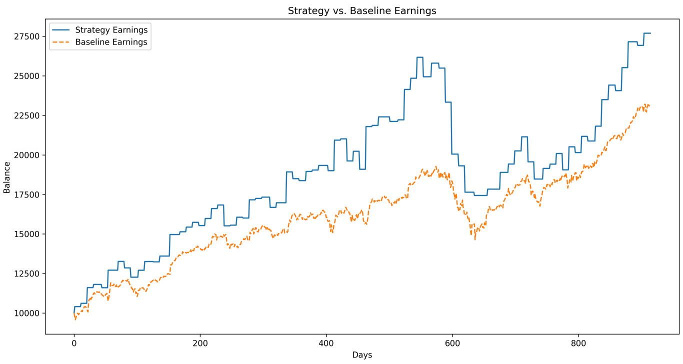

Figure 9. Strategy vs. baseline (buy and hold) earnings.

5.2. AAPL Options Performance

We again see promising results that are able to scale $10,000 USD into $300,000 USD over the span of 4 years. Again, the liquidity assumption is imposing some unwise optimism on the results and hence needs to be checked or discarded.

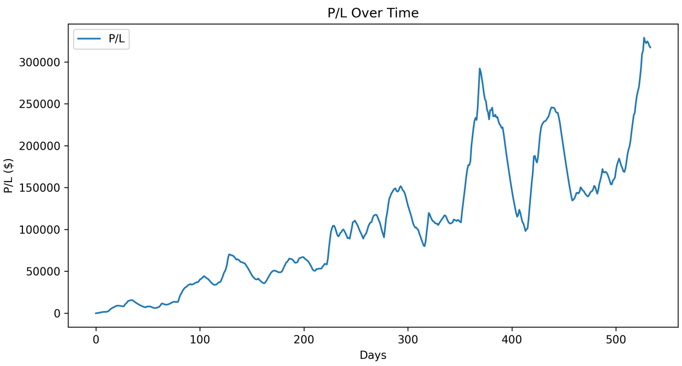

Figure 10. Options strategy P/L results.

The rolling average EV plot again shows promising results.

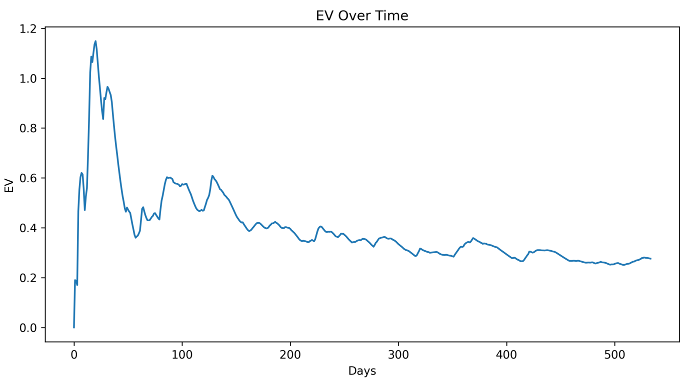

Figure 11. Options strategy average EV per trade over time.

The updated KDE shows reduced outliers and a more feasible distribution estimate.

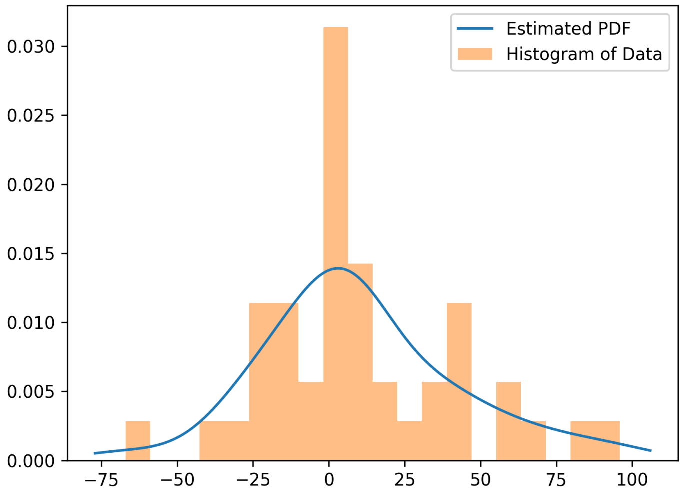

Figure 12. KDE for ROI(%) per month.

Lastly, for the 60 month simulations drawn from the KDE, we observe significantly lower variance across simulated curves.

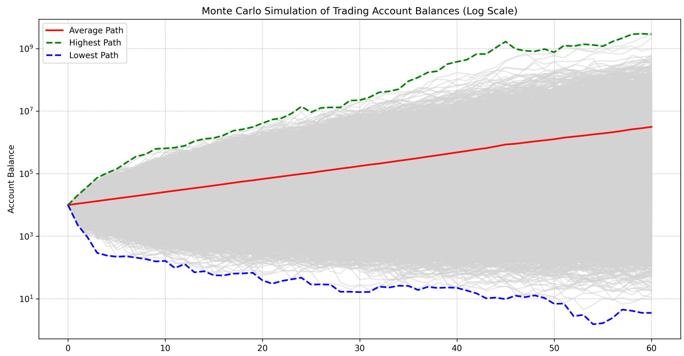

Figure 13. KDE for ROI(%) per month.

6. Addressing the Concerns

6.1. Conservative Liquidity Assumptions

We make the assumption that at most 30 ATM 20 DTE call options can be purchased. This will impact the compounding of the strategy but will bring the backtesting results closer to reality.

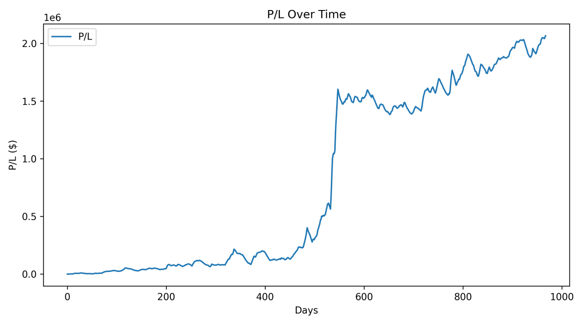

Figure 14. 2016 → 2024 backtest results with conservative liquidity.

Applying this modified strategy over the full test set (2016 → 2024) we observe a final balance of $\$2.1M$ USD – a 40% decrease. Then for the backtest ending before 2020, we observe the following:

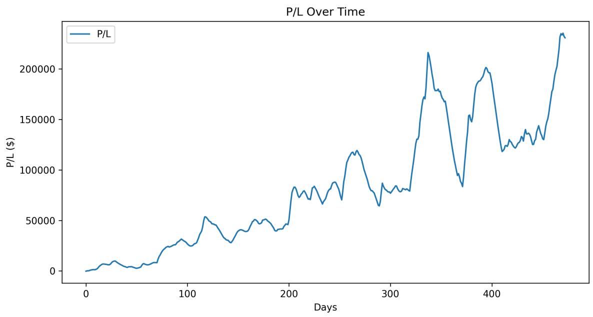

Figure 15. 2016 → 2020 backtest results with conservative liquidity.

a 20% decrease in final balance, from $300,000 USD down to $240,000USD.

6.2. Inference on the Median of Simulated Results

By means of **bootstrapping**, we are able to provide confidence intervals for the median simulated final balance over a period of 60 months. Using monthly strategy ROI data from the whole testing set, we observe a median final balance of $244,000 USD with a 95% confidence interval as follows:

$$\text{Median Final Balance:} \quad [2.31 \times 10^5, 2.56 \times 10^5]$$

a significant decrease in expected final balance by multiple orders of magnitude.

For monthly ROIs sampled from a reduced volatility timeframe (before 2020), we apply the same bootstrapping techniques to conclude with a median final balance after 60 months of $135,000 USD with a 95% confidence interval as follows:

$$\text{Median Final Balance:} \quad [1.3 \times 10^5, 1.4 \times 10^5]$$

7. Conclusion

While the results may initially seem promising, the strategy is subject to assumptions that remain to be tested. It is apparent that a positive EV is probable on a month-by-month basis, however the simulations conducted show between a 12% and 20% chance of the strategy not being profitable over the span of 5 years. Coupled with the mere assumptions regarding generalizability and the poor precision estimates, it is reasonable to head back to the drawing board.

The backtest results that sample monthly ROIs from periods of reduced volatility display significantly lower returns – indicative of some volatile market trends skewing the simulation balances upwards. Furthermore, inference on the median as opposed to the mean shows final balances which are orders of magnitude smaller than the naïve backtests.

The feature engineering portion of this model is not extensive. Different financial features with significant predictive power need to be used. Furthermore, different parameter choices for H and W need to be grid searched and cross validated.

The assumption that the matching procedure will yield a positive effect on performance needs to be checked. It is quite feasible that this matching procedure leads to poor generalization and overfitting.

This model should also be deployed to other ticker symbols. While it is known that this model performs poorly on TSLA and SPY, the author has designed a ranking algorithm by means of a custom kernel to find stocks with the most similar characteristics to AAPL. Once option chain data for these tickers is found, the true generalizability of the model can be tested.

The spatial recognition abilities of the CNN are well suited to tasks such as facial recognition, but perhaps kernel activation for uniform squares (only differing in grayscale intensity) is minimal. This simplicity which initially appeared to be a blessing is likely limiting the CNN from utilizing its full power of feature recognition. It is of interest to the author to explore other graphical representations of temporally linked time series to perhaps give the CNN inputs that are able to give rise to higher kernel activations.

Despite the poor results seen within the summary, this model is still able to generate significant alpha on AAPL stock and far exceeds the precision and backtest quality of other models which the author has built. These include, but are not limited to the following:

→ Recurrent Neural Networks
→ Logistic Regression
→ Transformer Models
→ Ensemble models
→ AdaBoost
→ GARCH Models
→ ARIMA Models
→ Support Vector Machines
→ Discriminant Analysis
→ LSTM Models
→ CNN-LSTM Hybrid Models

8. About the Author

Mohamad Ahmad is a mathematics and statistics graduate from the University of Ottawa. He has been interested in financial ML models since 2020 where he started trying many simple models for financial market regression and classification.

To no surprise, he has yet to “crack the secret" and still continuously engages in quantitative analysis primarily as a hobby. He has tried over 100+ supervised learning models over the span of 4+ years and thoroughly enjoys the hunt for alpha as a regular hobby.

Mohamad is currently a cofounder at BlueGem.ai, a data literacy education and consulting firm. He mostly spends his days preparing and teaching asynchronous content on mathematics, statistics, machine learning, deep learning, and artifical intelligence.

Through BlueGem, he aims to decentralize ML/AI education in aim to turn the industry into a more equal-equity landscape, free of financial and geographical boundaries.

Mohamad's interests span mathematics, statistics, deep learning, computer vision and business. He has ran startups prior to BlueGem and will continue to love the entrepreneurial lifestyle. In 2022, he won 2 entrepreneurial scholarships from the University of Ottawa for his efforts in the garment manufacturing company he was an owner of.

8.1. Contact Information

Email: mohamad@bluegem.ai
Telephone: +1 (343)-576-0150