Convolutional Neural Network-based a novel Deep Trend Following Strategy for Stock Market Trading

Jagdish Chakole[a], Manish Kurhekar[a]

[a]*Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology, Nagpur, India*

Abstract

Extracting or predicting future stock price trends from the current and historical stock trading activity is an open research problem. Convolution Neural Network (CNN) recently shown excellent performance to extract high-level features from raw data in many domains. In this research work, we trained CNN to extract future stock price trends. Trend Following is a trading strategy that does not require prior knowledge of the future stock price trend. In the Trend Following trading strategy, traders buy stock in an uptrend and sell the stock in a downtrend. Trend Following trading strategy is based on the current price trend and the assumption that the current price trend will continue further for some time, but if the current price trend does not proceed then this trading strategy fails to provide the profit. Our objective is to improve the Trend Following trading strategy by combining the advantages of both the CNN-based prediction and the Trend Following trading strategy. In this research work, we proposed a modified CNN-based Trend Following trading strategy named Deep Trend Following (DTF) in which trading decisions are based on the current and predicted future price trend of the stock. Our experimental results are twofold. Firstly our trained CNN-based classifier outperformed the baseline methods. Secondly, the DTF trading strategy outperformed three trading strategies viz the CNN-based, Buy-and-Hold, and the simple Trend Following trading strategies on the American and the Indian index stocks.

Keywords

Algorithmic Trading, Convolutional Neural Network, Trend Following Trading Strategy

1. Introduction

Stock market trading is a fascinating domain in search of lucrative profit. The movement of the stock price is non-linear, non-stationary, and noisy as the stock price is based on various factors, and most of these factors are uncertain. Many factors can move the stock price up or down including demand-and-supply of stock, company's management, competition from competitors, government policy, the policy of the central bank of the country, any news related to the company can influence the stock price. As per the Efficient Market Hypothesis ([1]), the stock market is efficient, and it reflects the effect of information on the stock price.

Traders and investors buy, sell, or hold (do nothing) stocks for profit, but their objectives are different. The perspective of the investor is long term based on the long term trend and the prospect of the company, having a stockholding period ranging from a year to a few years. In contrast, traders buy (sell) and sell (buy) the stock based on the short term trend, having a stockholding period of a few days, hours, minutes, or seconds. The investors predict the long term trend of stock based on fundamental analysis. The trader uses technical analysis to predict the short term trend of the stock. Technical analysis uses historical data to predict the short term trend, and it uses moving averages and other indicators.

As stock price depends on various uncertain events. Long term trend prediction using historical and current stock market data is not reliable as the time window is long, so there is a high probability of occurrence of any such event. In the case of short term trend prediction as the time window is short probability of occurrence of an uncertain event is low. So, in this research work, our focus is on the short term trend prediction.

Algorithm Trading (AT) is nowadays being used heavily in the stock market for trading. In AT, computer programs take the trading decision and execute the trading action, i.e. buying or selling of stock as per the logic written in the computer program. Basically AT is automate the trading process. It can continuously track the stock price activity and can quickly respond to any opportunity in terms of the trading decision. AT is mostly used for two things, first to automate a predefined trading strategy, and second is finding a trading strategy along with its automation.

Trend Following (TF) is a trading strategy within traders who buy stock in an uptrend and sell the shares in the downtrend([2, 3]). So, it does not require prior knowledge of the future stock price trend. Trend Following trading strategy is based on the current price trend and the assumption that the current price trend will continue further for some time. It fails to provide profit if the cur-

MUFin21 (International Workshop on Modelling Uncertainty in the Financial World)
jagdish06@students.vnit.ac.in (J. Chakole); manishkurhekar@cse.vnit.ac.in (M. Kurhekar)
0000-0003-0242-7297 (J. Chakole); 0000-0002-6409-3280 (M. Kurhekar)

rent price trend does not continue after trading action is taken based on the current trend. The success of the TF strategy is based on the behavior of the stock price trend after the trading action is taken. If we predict in advance the future behavior of the stock price trend, then this TF strategy will be more profitable. In that case, the TF strategy will take a trading decision based on current and predicted future stock price trends.

Extracting or predicting future stock price trends from the current and historical stock trading activity is an open research problem for traders and researchers. Many researchers attempted to predict the future stock price trend based on historical stock market data. They used various prediction methods, including machine learning technics like SVM, Random Forest, ANN but input features selection is a critical task for the success of these methods. Convolution Neural Network (CNN) showed the excellent performance to extract high-level features from raw data in many domains, including computer vision. Recently some researchers used ([4, 5]) CNN to predict the future stock price trend as CNN is capable of finding the patterns in the raw data.

The prediction or classification of stock price or trend using machine learning methods is looking promising but is not always reliable. So, taking a trading decision based only on the prediction of future price trends is not a good strategy. The TF strategy based solely on the current stock price trend also does not look promising. The combination of these two strategies will reduce the disadvantages of both approaches. Our objective is to improve the TF trading strategy by combining the advantages of both, i.e. CNN-based future stock price trends prediction and TF trading strategy. In this research work, we proposed a modified CNN-based TF trading strategy, and we name it as Deep Trend Following (DTF) in which trading decisions will base on the current price trend and CNN-based predicted future price trend of the stock.

We experimented with the proposed DTF strategy on the American and the Indian index stocks. As we created the DTF strategy using a CNN-based binary classifier to predict the next session, $t + 1$ close price trend and simple Trend Following trading strategy. Our experimental results are twofold, firstly we evaluated the performance of our CNN-based classifier with the baseline method on the same American index stocks dataset used in the baseline paper ([4]) and the results reported in the Table 2 and Table 3. Secondly, we evaluated the performance of the proposed DTF strategy with the CNN-based, Buy-and-Hold, and simple Trend Following trading strategies using performance evaluation metrics mentioned in Section 4.4 on the American and the Indian index stocks and the results reported in Table 4.

In ([4]), authors use CNN to predict the next-day stock price trend using features from diverse sources. They represented the input data in 2D and 3D forms. In ([6]), authors used Deep Neural Network to predict the next one-minute return prediction and used this prediction for trading strategy for high-frequency trading. Authors in ([7]) used CNN to predict the one-minute ahead trend of the cryptocurrency exchange rates. They experimented on the six popular cryptocurrencies. In ([8]), index stock trend predicted using CNN. They also optimized the hyperparameters of the CNN model using a Genetic Algorithm. In ([9]), the authors proposed an algorithmic trading strategy using CNN. They converted the technical indicator data into a 2-dimensional image to train the CNN model. In ([10]), authors extracted stock price trend from limit order book data using a CNN. They used a Long Short Term Memory method to know past time dependency in the data. In ([11]), the authors used thresholded ensemble CNN for financial data, and this model outperformed the technical methods.

The remainder of the paper is organized as follows. Section 2 describes the methodology used in this work, including Deep Learning and CNN. Section 3 explains the proposed trading system. Section 4 is devoted to experimental results, including data representation and performance evaluation of the proposed trading system. Concluding remarks with suggestions for future research are given in Section 5.

2. Methodology

This section provides a concise presentation to Convolutional neural networks.

2.1. Convolutional Neural Network

The typical architecture of CNN consists of a series of layers, and these layers can classify as the Input layer, Convolutional layer, Pooling layer, Fully connected layer, and Output layer as shown in Figure 1 ([12, 13]) ([14]).

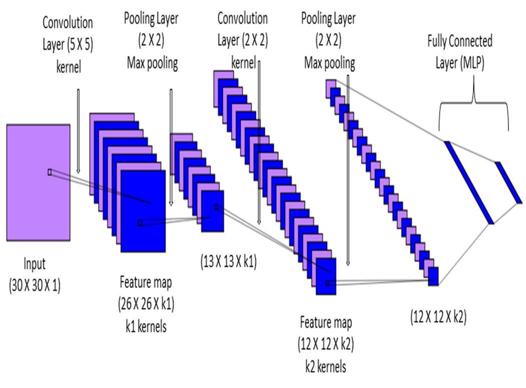

Figure 1: Convolutional Neural Network

2.1.1. Convolutional layer

The first layer in a CNN is a convolutional layer, as shown in Figure 1, this layer extracts features called a feature map from the input data by performing the convolution operation using a filter. In this work, input data represented as a two-dimensional array $A(m,n)$ having m number of rows and n number of columns. A two-dimensional filter $F(p,q)$ having p number of rows and q number of columns used. The convolution operation performed on two functions so the input A and filter F can be thought as functions, the convolution operation on these two functions performed as per the Equation 1 ([15]) to produce another function B which is a 2-D feature map having $m-p+1$ rows and $n-q+1$ columns. The values in the filter learned while training the network.

$$B(i,j) = A(i,j) \otimes F(p,q) = \sum_{p=-m/2}^{m/2} \sum_{q=-n/2}^{n/2} A(i+p, j+q) * F(p,q) \tag{1}$$

The activation function is applied to the output of the convolution operation as shown in Equation 2. The nonlinear activation function commonly used is Relu $f(a) = \max(0,a)$.

$$B(i,j) = \delta\Big(\sum_{p=-m/2}^{m/2} \sum_{q=-n/2}^{n/2} A(i+p, j+q) * F(p,q) \Big) \tag{2}$$

2.1.2. Pooling layer

The pooling layer reduces the dimensionality of the feature map by subsampling of the data while retaining the useful features. It reduces the computational cost and also it is a measure to avoid overfitting of the CNN model. In pooling operation, pooling window used, the values under the window reduced to a single value, which reduces the size of the input to the next layer. The commonly used pooling operation is the max pooling, in which among the values, the maximum value selected.

2.1.3. Fully Connected Layer

The convolution operation, activation function, and pooling operation are responsible for extracting features from the input. To map these extracted features in the previous layers into the final output, at the end of CNN, a fully connected network, i.e. multilayer perceptron (MLP) is appended. It converts the last 2-D feature map to the 1-D feature vector. In a classification problem, a single output selected from the feature vector based on the probabilities of all the outputs using a probabilistic function. Softmax is the commonly used probabilistic function for this task.

2.2. Trend Following Strategy

Trend Following is a stock trading strategy that takes trading decisions based on the current stock price trend. It takes a trading decision whenever current stock price trend changes, i.e. from an uptrend to downtrend or downtrend to uptrend for that it continuously observes the current trend. In stock market trading buy first and sell later is called a Long position. Sell first and buy then is called a Short position. Traders take a long position whenever they think the stock price will increase and take a short position whenever they feel the stock price will decrease.

3. Proposed Trading System

Predicting the future stock price trend and optimal utilization of the prediction along with implicit provision for risk management is the fundamental idea of this work. In this section, we propose the complete end-to-end Algorithmic trading strategy using the Trend Following strategy and CNN. Our proposed trading system will work on single security (or stock) for simplicity and will take only one trading decision, i.e. buy, sell, or hold (do nothing) a previous trading position on every trading session. We have included the transaction cost of 0.10% of the trading amount for buy or sell in our study. We have performed experimentation on the daily stock data having open, high, low, and close price information along with the volume of the trading.

The proposed trading system is shown in Figure 2, in which the CNN module is responsible for predicting future stock price trends. It has trained from historical stock market data. It will predict the next trading session $(t+1)$ close price trend from live and historical stock market data. The technical analysis indicators module tells the current (t) stock price trend at the closing time of the current session t from live and historical stock market data.

The Trend Following trading agent will take a trading decision at the open time of the next trading session $t+1$. It is based on the prediction of CNN about the stock close price trend of next trading session $(t+1)$ and current (t) stock price trend provided by technical analysis indicators as shown in the Algorithm 1. In the Algorithm 1, CR_trend denotes the current (t) stock price trend provided by the technical analysis indicators, and NX_trend indicates the prediction of CNN about the next trading session $(t+1)$ stock close price trend. We are taking a trading decision at the opening of the next trading session $t+1$ instead of the close of the current trading session t because non-trading period information also plays a significant role to decide the stock price trend.

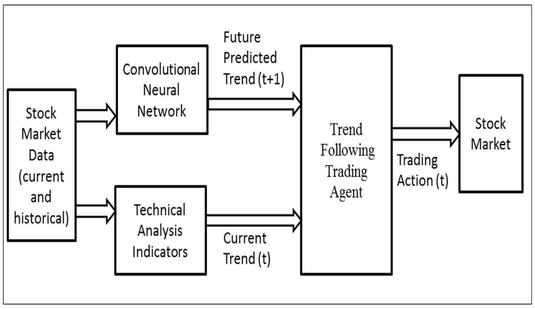

Figure 2: Proposed Trading System

Algorithm 1 Deep Trend Following

if (CR_trend == Uptrend) AND (NX_trend == Uptrend) **then**
    if (CR_position == NIL) **then**
        Open a Long position
    else if (CR_position == Long) **then**
        Hold the previous Long position
    else if (CR_position == Short) **then**
        Close the previous Short position
        Open a new Long position
    end if
else if (CR_trend == Downtrend) AND (NX_trend == Downtrend) **then**
    if (CR_position == NIL) **then**
        Open a Short position
    else if (CR_position == Short) **then**
        Hold the previous Short position
    else if (CR_position == Long) **then**
        Close the previous Long position
        Open a new Short position
    end if
else
    if any previous position then Hold it
    if no previous position then do nothing
end if

4. EXPERIMENTAL RESULTS

This section describes the data representation to train the CNN model. The detail of the baseline algorithm to compare the performance of the proposed classifier. The experimental setup of the proposed DTF trading strategy, also about performance evaluation metrics used. Detail of the CNN-based, Buy-and-Hold, and simple Trend Following trading strategies used to compare the performance of the proposed DTF strategy and also discusses the experimental results.

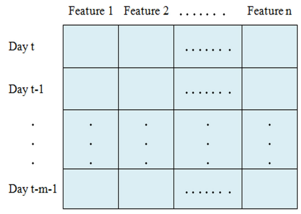

Figure 3: Input features in matrix form

4.1. Data Representation

Our proposed Deep Trend Following trading strategy uses the stock price trend predicted by the two-dimensional CNN, to make a trading decision. CNN predicts the next session stock price trend based on the current and historical stock trading data, so input data to CNN is represented in the two-dimensional matrix form, as shown in Figure 3. We have derived n features from the stock trading data, and these total n features form the columns of the matrix. The entire m rows of the matrix are the current and historical days, i.e. trading sessions. The optimal value of m is ten, i.e. ten recent days' data to predict the next trading session close price selected by experimentation on different values of m. The features used and the calculations are described in Table 5.

Total features are thirty, which can group as technical indicators, simple moving averages, exponential moving averages, day of the week. Day of the week is a very significant feature because the global stock markets correlated up to a certain extend. The non-trading hours period is also a significant factor in the stock price trend. The experimental data is divided into training and testing, as shown in Table 1.

The training data is very crucial for the performance of the machine learning model. We identified the outlier in our dataset based on the observation of the daily % changes in the close price, as shown in Figure 4. We removed the outlier using Z-statistics.

4.2. Baseline algorithm

Our baseline algorithm is the one reported in "Ehsan Hoseinzade and Saman Haratizadeh (2019)". We compare the performance of the proposed method with the baseline algorithm ([4]). We have experimented with the proposed method with the same dataset mentioned in the baseline algorithm. Compared the results of the proposed method with the results reported in the baseline algorithm.

Table 1
Experimental Data of the American index stocks DJIA, NASDAQ, S&P 500, RUSSEL and Indian index stock NIFTY

Index Stock	Time	Total Span	Training Span	Testing Span
NASDAQ	Start	1990-01-02	1990-01-02	2018-07-10
	End	2019-12-31	2018-07-09	2019-12-31
DJIA	Start	1990-01-02	1990-01-02	2018-07-06
	End	2019-12-31	2018-07-05	2019-12-31
S&P 500	Start	1990-01-02	1990-01-02	2018-07-09
	End	2019-12-31	2018-07-06	2019-12-31
RUSSEL	Start	1990-01-02	1990-01-02	2018-05-29
	End	2019-12-31	2018-05-25	2019-12-31
NIFTY	Start	2000-01-03	2000-01-03	2018-01-03
	End	2019-12-31	2018-01-02	2019-12-31

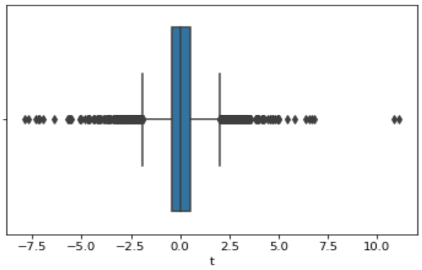

Figure 4: Outlier in the DJIA dataset

In the baseline algorithm, They predicted the next day binary direction of the index stock using the CNN-based framework. They used data from a diverse set of variables. Derived 82 features from data to train the CNN model. They used the past 60 days' data. So, their input matrix dimension is 60X82.

4.3. Experimental Setup

Our experimentation grouped into two tasks. First, we predicted the next day's stock price trend at the close price of the next day trading session, this is a classification problem, as shown in the equation below.

$$class = \begin{cases} 1, & \text{if} \quad Close\_Price_{t+1} > Close\_Price_t \\ 0, & \text{else} \end{cases}$$

We compare the classification results with the classification result of the baseline method and the other methods reported in the baseline paper, as shown in Table 2 and Table 3 ([4]).

Second, as per our proposed method, we created the DTF trading strategy using classification result of the proposed classifier and Trend Following trading strategy. We compared the result of the DTF trading strategy with the buy-and-hold, CNN-based trading strategy, and simple Trend Following trading strategy. All four trading strategy experimented on DJIA, NASDAQ, and S&P 500 American index stocks; also on Indian index stock NIFTY as shown in Figure 5, Figure 6, Figure 7 and Figure 8 respectively.

4.4. Performance evaluation metrics for trading strategy

We have used % Accumulated Return, % Maximum Drawdown, % Average Daily Return, % Average Annual Return, Sharpe Ratio, Standard Deviation, Skewness, and Kurtosis[3] to compare the proposed DTF trading strategy with the CNN-based, simple Trend Following and buy-and-hold trading strategies.

4.5. Trading Strategies

We compare the performance of our proposed DTF trading strategy with the Simple Trend Following, Buy-and-Hold, and CNN-based trading strategy in terms of % accumulated return. In the CNN-based trading strategy buy action is taken when CNN predicts, the price will go up at the close of the next trading session $t + 1$, and it will take sell action when CNN predicts the price will go down. Buy action suggestion ignored if an already long position is open similarly sell action ignored if an already short position is open.

4.6. Experimental Results

Experimentation in this work is grouped into two tasks. In the first task, we proposed CNN based binary classifier and compared the results of the classification with the classifiers in the baseline paper. In the second task, we introduced the DTF trading strategy using CNN based binary classifier and simple Trend Following trading strategy. Also, compared the result of the proposed DTF

Table 2
Average F-measure of all four trading strategy

	1	2	3	4	5	6
Market	PCA+ANN	CNN-cor	Technical	3D-CNNpred	2D-CNNpred	DTF
DJI	0.4283	0.39	0.415	0.4979	0.4975	0.53
NASDAQ	0.4136	0.3796	0.4199	0.4931	0.4944	0.55
S&P 500	0.4237	0.3928	0.4469	0.4837	0.4914	0.56
RUSSELL	0.4279	0.3924	0.4525	0.4846	0.5002	0.53

Table 3
Best F-measure of all four trading strategy

	1	2	3	4	5	6
Market	PCA+ANN	CNN-cor	Technical	3D-CNNpred	2D-CNNpred	DTF
DJI	0.5392	0.5253	0.5518	0.5612	0.5562	0.61
NASDAQ	0.5312	0.5498	0.5487	0.5576	0.5521	0.61
S&P 500	0.5408	0.5723	0.5627	0.5165	0.5532	0.58
RUSSELL	0.5438	0.5602	0.5665	0.5787	0.5463	0.58

trading strategy with the buy-and-hold, CNN-based, and simple Trend Following trading strategy.

In the first task, as mentioned in Section 5.3, we compare the classification result of the proposed CNN-based binary classifier with the baseline classifiers in terms of the macro-average and macro-best F-measure scores. We used the macro-average-F-measure comparison metric instead of accuracy because the financial data has an imbalanced class distribution [5]. The macro-average F-measure score, and macro-best F-measure score of all methods, including the proposed classifier mentioned in Table 2, and Table 3 respectively. In both the Tables column numbers 4, 5 are the two baseline classifiers. Column number 6 is the proposed classifier, and column numbers 1, 2, 3 are the classifiers reported in the baseline paper.

The proposed classifier outperformed the baseline classifiers and other classifiers reported in the baseline paper in terms of the average and best F-measure score. Our classifier outperformed the baseline classifier because a deep learning method required extensive training data, and we used large training data compared to the baseline method. Also, Their input matrix has 60 rows, i.e. 60 previous days, and we have used ten past days. Only recently trading days influence the stock price trend, and they used too many last days' data in the input matrix. We also used the feature to capture non-trading time information.

In the second task, the performance of the proposed DTF trading strategy compared with the performance of the CNN-based, simple Trend Following, and buy-and-hold trading strategies on the American index stock DJIA, NASDAQ and S&P 500; the Indian index stock NIFTY using performance evaluation metrics mentioned in Section 5.4. This comparison is shown in Table 4. The % Accumulated return of the proposed DTF trading strategy is considerably better than the other trading strategies mentioned in Table 4. The comparison of all four trading strategies including proposed DTF trading strategy in terms of % Accumulated return for index stocks DJIA, NASDAQ, S&P 500 and NIFTY is shown in Figure 5, Figure 6, Figure 7 and Figure 8 respectively. Figure 9, summarizes the performance of these four trading strategies on the same index stocks in terms of % accumulated returns.

The maximum drawdown of the trading strategy represents the historical risk associated with the trading strategy, so it should be minimum. The maximum drawdown of the proposed DTF trading strategy is considerably minimum than the maximum drawdown of the Buy-and-Hold and CNN based trading strategies on all datasets. It is better than simple Trend Following trading strategy on two datasets and approximately equal for the other two datasets. The proposed DTF method's % Daily and % Annual Returns is better than all different trading strategies, as shown in Table 4. Similarly, the Sharpe ratio of the trading strategy should be maximum, and the Sharpe ratio of the proposed DTF method is better than all other trading strategies, as shown in Table 4. The skewness and kurtosis of the proposed DTF method on all datasets are reasonably acceptable. The volatility, i.e. standard deviation of the all four trading strategy, is approximately the same.

Table 4
The performance comparison of the all four trading strategy on the experimental test dataset of four index stocks

Index stock	Performance evaluation metrics	Proposed Model	Buy-and-Hold	CNN	Trend Following
S&P 500	% Accumulated Return	26.88	15.70	-15.57	24.01
	% Average Annual Return	18.56	10.84	-10.75	16.58
	% Maximum Drawdown	14.10	17.54	17.31	14.12
	Standard deviation	0.835	0.855	0.737	0.842
	% Average Daily Return	0.074	0.043	-0.043	0.066
	Sharpe Ratio	1.581	0.946	-1.159	1.426
	Skewness	-0.188	-0.882	-0.358	-0.241
	Kurtosis	2.650	2.434	9.904	2.369
DJIA	% Accumulated Return	31.10	16.37	5.19	28.11
	% Average Annual Return	21.07	11.09	3.51	19.04
	% Maximum Drawdown	14.28	16.33	12.67	14.18
	Standard deviation	0.809	0.847	0.707	0.812
	% Average Daily Return	0.084	0.044	0.014	0.076
	Sharpe Ratio	1.825	0.975	0.458	1.669
	Skewness	-0.114	-0.802	0.713	-0.161
	Kurtosis	2.636	2.180	3.966	2.469
NASDAQ	% Accumulated Return	33.93	15.29	-11.67	32.96
	% Average Annual Return	23.11	10.41	-7.94	22.45
	% Maximum Drawdown	17.12	21.90	22.77	17.12
	Standard deviation	1.107	1.143	0.994	1.113
	% Average Daily Return	0.092	0.041	-0.031	0.089
	Sharpe Ratio	1.487	0.754	-0.642	1.446
	Skewness	-0.296	-0.583	-0.493	-0.219
	Kurtosis	2.707	2.357	6.092	2.604
NIFTY	% Accumulated Return	23.74	17.35	-5.41	12.29
	% Average Annual Return	12.28	8.98	-2.80	6.36
	% Maximum Drawdown	14.28	14.55	20.08	15.14
	Standard deviation	0.810	0.826	0.790	0.834
	% Average Daily Return	0.049	0.035	-0.011	0.025
	Sharpe Ratio	1.288	0.874	-0.149	0.725
	Skewness	0.201	0.264	1.359	0.161
	Kurtosis	1.982	1.860	15.23	1.804

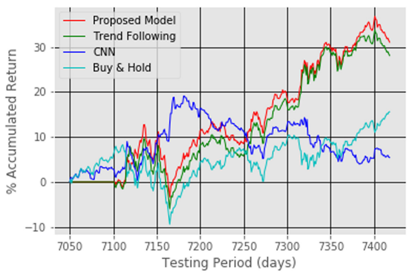

Figure 5: The comparison of the performance of all four trading strategy on the test data of DJIA index stock using % accumulated return

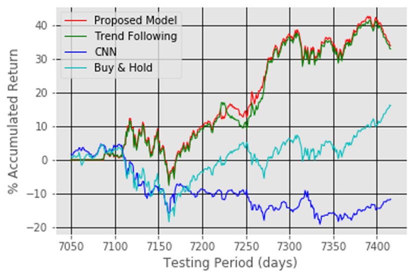

Figure 6: The comparison of the performance of all four trading strategy on the test data of NASDAQ index stock using % accumulated return

Table 5
Description of features

S.N.	Feature	Description	Source/Formula
1.	Close price	Daily close price of the stock	Yahoo Finance
2.	Open price	Daily open price of the stock	Yahoo Finance
3.	Week day	Day of the week	Datetime Library
4.	RSI	Relative Strength Index	TA Library
5.	R	Williams Percent Range (%R)	TALibrary
6.	CCI	Commodity Channel Index	TA Library
7.	RSI trend	if RSI>70 then 2, if RSI<30 then 1 else 0,	
8.	R trend	if R>-100 then 2, if R>100 then 1 else 0,	
9.	CCI	if CCI<70 then 2, if CCI<30 then 1 else 0,	
10.	MOM1	One day % return	$[(\frac{Close_t - Close_{t-1}}{Close_{t-1}}) * 100]$
11.	MOM2	Two days % return	formula is similar to S.N. 6
12.	MOM3	Three days % return	formula is similar to S.N. 6
13.	VC1	% change in volume compare to volume one day before	$[(\frac{Volume_t - Volume_{t-1}}{Volume_{t-1}}) * 100]$
14.	VC2	% change in volume compare to volume two days before	formula is similar to S.N. 9
15.	VC3	% change in volume compare to volume three days before	formula is similar to S.N. 9
16.	OHC	% change between open and high price	$[(\frac{High_t - Open_t}{Open_t}) * 100]$
17.	OLC	% change between open and low price	$[(\frac{Open_t - Low_t}{Open_t}) * 100]$
18.	OCC	% change between open and close price	$[(\frac{Open_t - Close_t}{Open_t}) * 100]$
19.	LHC	% change between low and high price	$[(\frac{High_t - Low_t}{Low_t}) * 100]$
20.	SMA5	Simple moving average n=5 days close price	$[\frac{1}{n} \sum_{i=t-n}^{t} Close_i]$
21.	SMA10	Simple moving average n=10 days close price	formula is similar to S.N. 16
22.	SMA15	Simple moving average n=15 days close price	formula is similar to S.N. 16
23.	SMA20	Simple moving average n=20 days close price	formula is similar to S.N. 16
24.	SMA25	Simple moving average n=25 days close price	formula is similar to S.N. 16
25.	EMA5	Exponential Moving Average n=5 days close price	EMA
26.	EMA10	Exponential Moving Average n=10 days close price	EMA
27.	EMA15	Exponential Moving Average n=15 days close price	EMA
28.	EMA20	Exponential Moving Average n=20 days close price	EMA
29.	EMA25	Exponential Moving Average n=25 days close price	EMA
30.	NT	Non trading period information	$[(\frac{Open_{t+1} - Close_t}{Close_t}) * 100]$

5. Conclusion

Owing to the noisy and non-stationary behavior of the stock price trend, the trading strategy based only on the prediction is risky. The performance of the Trend Following trading strategy depends on the stock price trend after trading action is taken but the Trend Following strategy can not predict the future stock price trend. In this paper, we proposed a modified Trend Following trading strategy named Deep Trend Following (DTF) trading strategy, which is a combination of CNN-based prediction with Trend Following trading strategy.

The main contribution of this research work is that it combines the CNN-based prediction with the Trend Following strategy in which the Trend Following strategy will take a trading decision based on the current trend and future trend predicted by the CNN-based classifier. This combination reduces the risk based on the only prediction. We experimented with the proposed DTF trading strategy on the American and the Indian index stocks and compared the results with the CNN-Based, simple Trend Following, and Buy-and-Hold trading strategy. The proposed DTF model outperformed the other three trading strategies in terms of percentage Accumulated Return, Maximum Drawdown, Average return, and the Sharpe ratio.

We also compared our CNN-based binary classifier with the baseline CNN-based binary classifier, and it outperformed the baseline classifier in terms of macro-average and macro-best F-measure scores. It performed better because we have considered the non-trading period information, and we have taken more training data

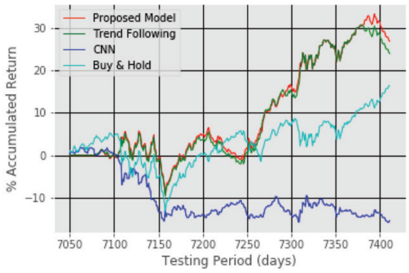

Figure 7: The comparison of the performance of all four trading strategy on the test data of S&P 500 index stock using % accumulated return

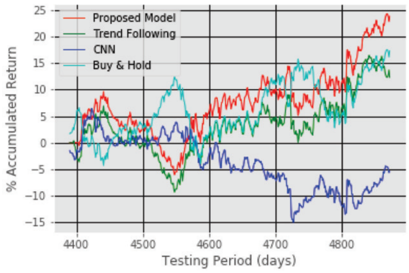

Figure 8: The comparison of the performance of all four trading strategy on the test data of NIFTY index stock using % accumulated return

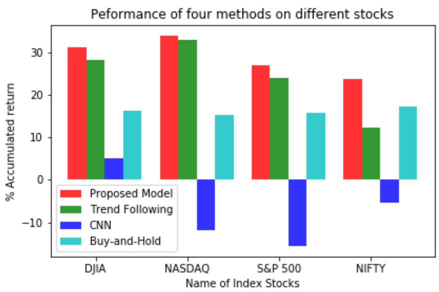

Figure 9: Performance of all four trading strategy in terms of % accumulated return of experimental test dataset of all four index stocks

compare to the baseline classifier. We experimented with the proposed classifier and DTF trading strategy on daily stock market data, in the future its performance on the small frequency data such as hourly, 15-minutes, 5-minute data should have been experimented with. We have selected the proposed CNN architecture by experimenting with different architectures, need to choose the CNN architecture by the more robust method in the future.

References

[1] E. F. Fama, L. Fisher, M. C. Jensen, R. Roll, The adjustment of stock prices to new information, International economic review 10 (1969) 1–21.

[2] S. Fong, Y.-W. Si, J. Tai, Trend following algorithms in automated derivatives market trading, Expert Systems with Applications 39 (2012) 11378–11390.

[3] J. Chakole, M. Kurhekar, Trend following deep q-learning strategy for stock trading, Expert Systems (2019) e12514.

[4] E. Hoseinzade, S. Haratizadeh, Cnnpred: Cnn-based stock market prediction using a diverse set of variables, Expert Systems with Applications 129 (2019) 273–285.

[5] H. Gunduz, Y. Yaslan, Z. Cataltepe, Intraday prediction of borsa istanbul using convolutional neural networks and feature correlations, Knowledge-Based Systems 137 (2017) 138–148.

[6] A. Arévalo, J. Niño, G. Hernández, J. Sandoval, High-frequency trading strategy based on deep neural networks, in: International conference on intelligent computing, Springer, 2016, pp. 424–436.

[7] S. Alonso-Monsalve, A. L. Suárez-Cetrulo, A. Cervantes, D. Quintana, Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators, Expert Systems with Applications 149 (2020) 113250.

[8] H. Chung, K.-s. Shin, Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction, Neural Computing and Applications 32 (2020) 7897–7914.

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

[10] Z. Zhang, S. Zohren, S. Roberts, Deeplob: Deep convolutional neural networks for limit order books, IEEE Transactions on Signal Processing 67 (2019) 3001–3012.

[11] S. Ghoshal, S. Roberts, Thresholded convnet ensembles: neural networks for technical forecasting, Neural Computing and Applications (2020) 1–14.

[12] Y. LeCun, Y. Bengio, et al., Convolutional networks for images, speech, and time series, The handbook of brain theory and neural networks 3361 (1995) 1995.

[13] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.

[14] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.

[15] G. S. Tandel, M. Biswas, O. G. Kakde, A. Tiwari, H. S. Suri, M. Turk, J. R. Laird, C. K. Asare, A. A. Ankrah, N. Khanna, et al., A review on a deep learning perspective in brain cancer classification, Cancers 11 (2019) 111.