Stock Price Prediction with CNN-LSTM Network

Yushi Guan **Peiyao Li** **Cheng Lu**

Abstract

Stock price prediction has always been a rewarding but challenging task since are inherent noises present in the stock time series data. The application of machine learning and deep learning in this domain has attracted many computer science researchers' attention. Most published state of the art machine learning models for stock prediction use a large set of technical and economic factors as feature inputs along with the target stock's historical data to predict future price movements [2][3][9]. Inspired by recent advances in natural language processing and speech synthesis [7][22], we propose a combined CNN-LSTM model that can achieve state of the art performance for stock price prediction without additional data such as technical and economic indicators in the input features. The proposed model outperformed a simpler LSTM model as well as non-neural methods implemented such as RF, KNN, and linear regression. Using a simple trading strategy, we demonstrate that our stock price prediction algorithm outperforms the buy and hold strategy on various index and stocks including SP500 index ETF (SPY), Intel (INTC), Tesla (TSLA). The SPY, INTC, TSLA, and M achieved Sharpe ratios of 0.74, 1.73, and 1.22 respectively.

1 Introduction

Forecasting the stock price has always been a challenging issue [21]. This is because the stock price is an inherently noisy time-series data that is both nonlinear and non-stationary. Many factors can contribute to the stock price fluctuations including macroeconomic events such as economic depression and micro-economic events such as the innovation of the company or merging of companies. Researchers have started to tackle the problem from the machine learning perspective with non-neural tools such as KNN, RF, and Linear Regression (LR), as well as neural methods such as autoencoder and LSTM [3][6]. Deep learning techniques have been proven effective in tackling challenging issues in many areas including image recognition and speech recognition problems [7][22]. Recent achievements in speech recognition with noisy input audio data have inspired using the architecture of a combined CNN-LSTM network for stock price forecasting. Utilizing this architecture along with various SMAs and EMAs for denoising, our model evaluates both the future stock price movement and the magnitude of price change.

We seek to study the performance of the model from different aspects by answering three research questions. Firstly, how well does the CNN-LSTM model compare against existing models? It is hypothesized that the CNN-LSTM model will at least perform as well as the current LSTM model and might even exceed its performance. For comparison against well known existing stock prediction classification models like KNN and RF, the model's regression output is converted to a binary classification to calculate its accuracy and F1 score [6]. Secondly, what features would help produce high-quality results when it comes to financial time series forecasting? We will explore the usefulness of adding technical and economic indicators for the stock price prediction. Lastly, would the model generate profits given a trading strategy? Using a simple trading strategy that buys the stock provided a positive outlook and sells the stock otherwise, we will evaluate whether implementing this model can beat the buy and hold strategy on the SPY, INTC, TSLA, M, and CELG stock tickers.

2 Related Work

Among the deep learning methods, RNN with LSTM units stands out as one of the most popular methods for financial time series forecasting, due to its ability to "memorize" the long-term and short-term trend of time series [1][19]. Strong motivations can be found in stock price prediction [3][5][6][14][20], where researchers implemented LSTM to forecast next day's stock price or return. These models typically took Open-High-Low-Close (OHLC) price and some hand-engineered features along with other economic factors such as interest rates and other stock prices as inputs. However, most of these engineered features are derived either from statistical assumptions of the financial markets or from domain knowledge, which might create bias in the model's prediction.

To address the aforementioned issue in feature engineering, CNN (convolutional neural networks) was introduced in time series prediction. For example, Long et al.[12] leveraged a CNN-RNN architecture to infer future stock price movement, which demonstrated advantages over traditional methods like KNN and SVM. Di Persio et al.[16] conducted a comparison study of CNN, LSTM, and MLP on SP500 index data and argued that CNN could model stock price series better than other architectures. The reason for adopting CNN could be explained in literature that CNN could extract high-quality data and improve the final prediction [9][12].

In light of the strengths of LSTM and CNN, we have decided to implement a CNN-LSTM model for financial series prediction. In this paper, we provide a thorough description of the model and highlight the findings made during the implementation. We conclude this paper by making suggestions for future improvements.

3 Data Preprocessing

We selected SP500 Index ETF (SPY) from Jan 2000 to April 2020 on Yahoo Finance website as our main training data. The data was split into 90% training set and 10% validation set. Fourteen commonly used technical indicators were derived [11][15][17] to investigate the effect of technical indicators on prediction accuracy. A handful of economic indicators including market indices and foreign exchange rates were also incorporated to investigate their effects on the model's performance and trading strategies. For detailed list of technical and economic indicator features investigated, refer to Table 4. This model is also evaluated on other stocks such as Intel, Tesla, Macy's, and Celgene.

4 Methods and Algorithms

In this section, we present the methods and algorithms used in our proposed model.

4.1 Feature Engineering

The objective of feature engineering is to create representations that are invariant to absolute stock price differences and robust to price fluctuations. Exponential Moving Average (EMA) is an exponentially decayed weighted sum of historical prices and has often been used to reduce noise [10]. In our work, instead of using EMA directly, we use the percent difference between EMAs with different window sizes to eliminate the influence of absolute stock prices. We define $EMARatio(m, n)$ as $(EMA(m) - EMA(n))/EMA(n)$ where m is the shorter window and n is the longer window. The performance of the proposed EMA ratio input verus the Rate of Change (ROC) input which simply calculates the percent difference between current price and a single price point in the past are compared. In addition to the EMA ratio or ROC inputs, the effect of introducing economic factors described in Table 4 as inputs was also investigated. The target output of our model is either percent change of next day's price compared to today's price (1 day horizon) or next 10 days' average price compared to today's price (10 day horizon). Please refer to Table 1 for details.

4.2 Model Architecture

Sequence modelling has been traditionally done with RNNs or LSTMs. In our work, we evaluated the effectiveness of the combination of CNN and LSTM for both dilated and non dialted CNN. Please see Figure 7 and Figure 8 for a visual representation of our architectures, and final proposed model architecture in Table 2.

4.2.1 Long Short Term Memory

LSTM has been widely used for sequence modelling tasks. For our implementation, we use a 2 layer LSTM. We refer readers to [8] for the LSTM cell operation.

The first layer of the 2 layer LSTM performs the following operation:

$$h_t^0 = \mathcal{H}\left(W_{xh}^0 x_t + W_{hh}^0 h_{t-1}^0 + b_h^0\right) \tag{1}$$

where $\mathcal{H}$ is the LSTM cell operation, x_t is the input feature or the output of previous CNN layers is and the second layer uses the output from the first layer,

$$h_t^1 = \mathcal{H}\left(W_{xh}^1 h_t^0 + W_{hh}^1 h_{t-1}^1 + b_h^1\right) \tag{2}$$

A fully connected layer uses the output of the last cell of the second LSTM layer to calculate for the target.

4.2.2 Model Output

The model outputs regression results of the next day's change in price or the next 10 days' average change in price. In order to benchmark that with the existing classification methods, we converted the regression result to a binary classification predicting whether the next price will increase or decrease. We include both the regression performance and classification performance in our results.

4.2.3 CNN and Dilated CNN

Convolution operations have been used widely in computer vision tasks and have been adopted as good feature extractors in natural language tasks. A 1d convolution operation is defined as:

$$o[i] = \sum_{u=-k}^{k} h[u]F[i-u] \tag{3}$$

where o is the output, $2k+1$ is the kernel size, h is the filter and F is the input feature. A dilated convolution is defined as:

$$o[i] = \sum_{u=-k}^{k} h[u]F[i-lu] \tag{4}$$

where l is the dilation.

The input to the CNN is the engineered input feature. The outputs of the CNNs are fed into the LSTM layers. The final proposed model's setting can be found in Table 3.

5 Results and Discussion

In this section, We provide answers to the research questions raised in the introduction and report on interesting findings unraveled during the research process.

5.1 Comparison of Various Model Architectures

We demonstrate our main findings in Table 5. In terms of binary prediction accuracy, our neural models CNN-LSTM (56.5%), D-CNN-LSTM (56.6%) exceeded the performance of the simpler LSTM model as well as other non-neural models. We find our implementation of benchmark models comparable with existing works' reported result [6]. As shown in Table 5, the non-neural method RF and KNN implemented achieved a testing accuracy of 51.9% and 52.0% respectively for next day movement prediction, which is on par with the existing work's reported accuracy[6]. In terms of regression result, CNN-LSTM also outperformed other existing regression models implemented as shown in Table 5.Thus, it can be concluded that the CNN-LSTM can achieve state of the art performance with an accuracy of 56.5% and F1 score of 0.719 [6]. The same conclusion can be reached via analyzing the mean square error and trading performance as shown in Figure 4. This meets our hypothesis that the proposed CNN-LSTM model can achieve equal or better result than just using LSTM alone.

Although the next day prediction's results are not too different amongst various models, the CNN-LSTM model significantly outperforms non-neural models for the 10 day average movement prediction. As shown in Table 5, the CNN-LSTM can capture the underlying future trend much better. With many output channels, CNNs can get activated by a variety of input patterns and pass the processed information to the downstream LSTM layers.

We further explored the effect of dilated CNN vs non-dilated CNN and found the dilated CNN architecture performed slightly better for the prediction of 10 day average future price change Table 5. This is likely because dilated CNN allows the model to include information from a distant past more effectively than a non-dilated CNN, which can be very useful for future trend prediction.

5.2 Most Optimal Features

For next day price prediction,adding technical and economic indicators did not impact the performance of the models, as shown in Table 6. However, an improvement in performance is observed for longer time horizon prediction when these indicators are included which does comply with prior work's findings [9][13].

Different context window were also investigated, and it is found that different context windows did not have significant impacts on the model's performances as they were all able to achieve good performance see Table . We could not lower the window size below 10 due to the model's implementation. Since a window size of 10 performed well consistently, it was decided to be the final context window size. We also found that the difference between EMA ratio and ROC input has no significant impact on our prediction and trading performance, as demonstrated in Figure 2.

5.3 Profitability Analysis

Due to the poor performance of the non-neural models in comparison to the neural methods, we only compared the trading results from neural models to further explore which model performs better in actual trading. To evaluate how the model performs in different industries, we also trained the model on stocks from different sectors: technology (Tesla and Intel), consumer discretionary (Macy's), healthcare (Celgene) (Figure. 5). It can be seen that our model performs better than or similar to the market performance for most of the stocks. However, our model does not perform well on very volatile asset such as Celgene (CELG) from the healthcare industry, it can fail to keep up with sudden price changes in the asset. Healthcare industries are also largely affected by public opinions. We also observe a similar pattern for our trading result during the recent market crash due to COVID-19, shown in Figure 6. Our model fails to predict the trend in highly volatile markets.

Although our model had good performance on next day prediction trading[**?**], its performance suffered when using 10 day horizon prediction to guide today's buy and sell action as shown in Figure 3. It performs significantly worse than the market consistently. The reason is most likely a result of incompatibility between the longer horizon prediction and our very simple trading strategy.

6 Conclusion and Future Work

We proposed a dilated-CNN-LSTM architecture for financial time series forecasting. We demonstrated the strength of our dilated-CNN-LSTM model over existing models when it comes to price prediction and profitability. Additionally, our work verified that including more features such as the economic indicators would slightly improve prediction performance for long horizon predictions.

One weakness of the current trading strategy is that it is only suitable for next day trading and cannot utilize the future average trend well in its strategy. For future work, we plan to develop a more fitting trading strategy for the 10 day average trend prediction. For these explorations, more domain knowledge would be required. Another weakness is that our model does not perform well for sudden volatile movements of prices. These movements are likely caused by factors such as news release that requires the assistance of sentiment analysis [4][18]. In order to respond to these event driven volatility, the next step is to explore incorporating sentiment analysis along with technical analysis to further enhance the model's predictions.

7 References

References

[1] Ayodele Ariyo Adebiyi, Aderemi Oluyinka Adewumi, and Charles Korede Ayo. Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, 2014(March), 2014.

[2] Yujin Baek and Ha Young Kim. ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications*, 113:457–480, 2018.

[3] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long- short term memory. *Journal of Cheminformatics*, 24(4):1–11, 2018.

[4] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[5] Kai Chen, Yi Zhou, and Fangyan Dai. A LSTM-based method for stock returns prediction: A case study of China stock market. *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, pages 2823–2824, 2015.

[6] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

[7] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep Speech: Scaling up end-to-end speech recognition. pages 1–12, 2014.

[8] Sepp Hochreiter. LONG SHORT-TERM MEMORY. 9(8):1–32, 1997.

[9] Ehsan Hoseinzade and Saman Haratizadeh. CNNPred: CNN-based stock market prediction using several data sources. 2018.

[10] J. Stuart Hunter. The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210, 1986.

[11] Yakup Kara, Melek Acar, and Ömer Kaan. Expert Systems with Applications Predicting direction of stock price index movement using artificial neural networks and support vector machines : The sample of the Istanbul Stock Exchange. *Expert Systems With Applications*, 38(5):5311–5319, 2011.

[12] Wen Long, Zhichen Lu, and Lingxiao Cui. Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, 164:163–173, 2019.

[13] Thi Thu Nguyen and Seokhoon Yoon. A novel approach to short-term stock price movement prediction using transfer learning. *Applied Sciences (Switzerland)*, 9(22), 2019.

[14] Adamantios Ntakaris, Giorgio Mirone, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Feature Engineering for Mid-Price Prediction with Deep Learning. *IEEE Access*, 7:82390–82412, 2019.

[15] Jigar Patel, Sahil Shah, Priyank Thakkar, and K Kotecha. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, 42(1):259–268, 2015.

[16] Luca Di Persio. Artificial Neural Networks architectures for stock price prediction : comparisons and applications. 10, 2016.

[17] Yauheniya Shynkevich, T. M. McGinnity, Sonya A. Coleman, Ammar Belatreche, and Yuhua Li. Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, 264:71–88, 2017.

[18] Jianfeng Si, Arjun Mukherjee, Bing Liu, Sinno Jialin, Pan Qing, and Li Huayi. Exploiting Social Relations and Sentiment for Stock Prediction. pages 1139–1145, 2014.

[19] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A Comparison of ARIMA and LSTM in Forecasting Time Series. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pages 1394–1401, 2019.

[20] Van-Dai Ta, CHUAN-MING Liu, and Direselign Addis Tadesse. Portfolio Optimization-Based Stock Prediction Using Long-Short Term Memory Network in Quantitative Trading. *Applied Sciences*, 10(2):437, 2020.

[21] Ming Chi Tsai, Ching Hsue Cheng, Meei Ing Tsai, and Huei Yuan Shiu. Forecasting leading industry stock prices based on a hybrid time-series forecast model. *PLoS ONE*, 13(12):1–24, 2018.

[22] Dong Xiao and Daniel M Ketcha. Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin. *Journal of Heterocyclic Chemistry*, 32(2):499–503, 1995.

8 Appendices

We included details about our model's input definition, and all the tables and figures in the appendices section.

8.1 Model Input

At each time step T, the Exponential Moving Average (EMA) for context window n is defined as:

$$EMA_T(n) = \begin{cases} P_t & t = T - n \\ \alpha P_t + (1-\alpha) EMA_{t-1} & t > T - n \end{cases} \tag{5}$$

where P_t is the price at time t. The coefficient α is derived from the window N:

$$\alpha = \frac{2}{n+1} \tag{6}$$

We define the $EMARatio$ as:

$$EMARatio_T\,(m,n) = \frac{EMARatio_T\,(m) - EMARatio_T\,(n)}{EMARatio_T\,(n)} \tag{7}$$

where n is larger than m.

At each time step T, the Rate of Change (ROC) with context window n is defined as:

$$ROC_T\,(n) = \frac{P_T - P_{T-n}}{P_{T-n}} \tag{8}$$

When T is omitted, it is assumed to be the current day. The table below specifies one of the two possible input sets when economic indicators are not added.

Table 1: Input Sets

EMARatio Input	ROC Input
EMARatio(2,3)	ROC(1)
EMARatio(2,5)	ROC(2)
EMARatio(3,5)	ROC(5)

8.2 Model Architecture

Table 2: Best Performing Dilated CNN-LSTM Architecture

Field	Setting
Conv1d	output_channel=20, kernel_size=3, dilation=2
Conv1d	output_channel=20, kernel_size=3, dilation=2
LSTM	hidden_size=20, num_layers=2, dropout=0.1
Linear	out_features=1

8.3 Figures

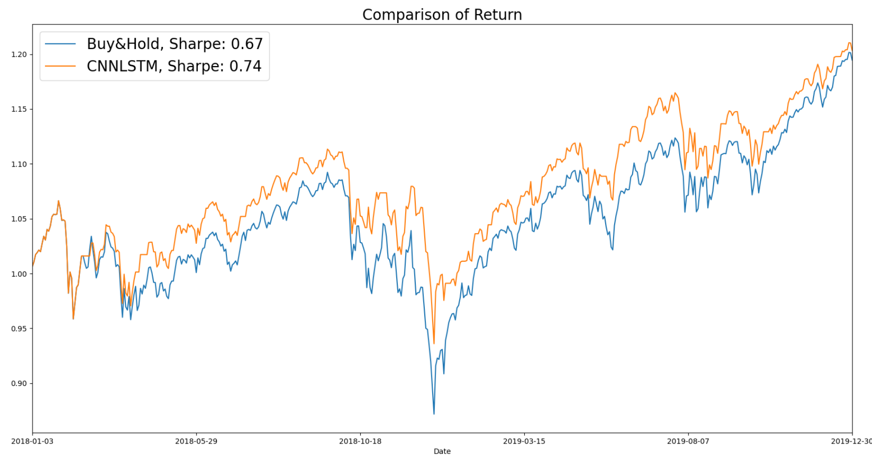
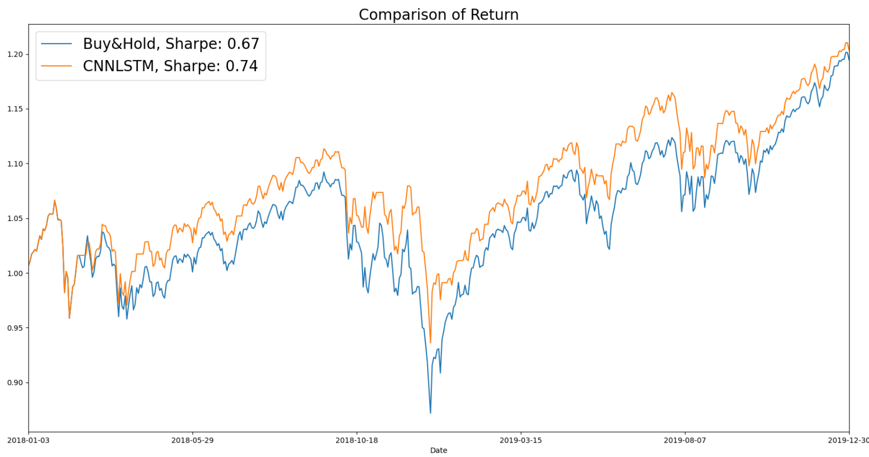


Figure 1: Performance of Final Proposed Model as Defined in Table 3

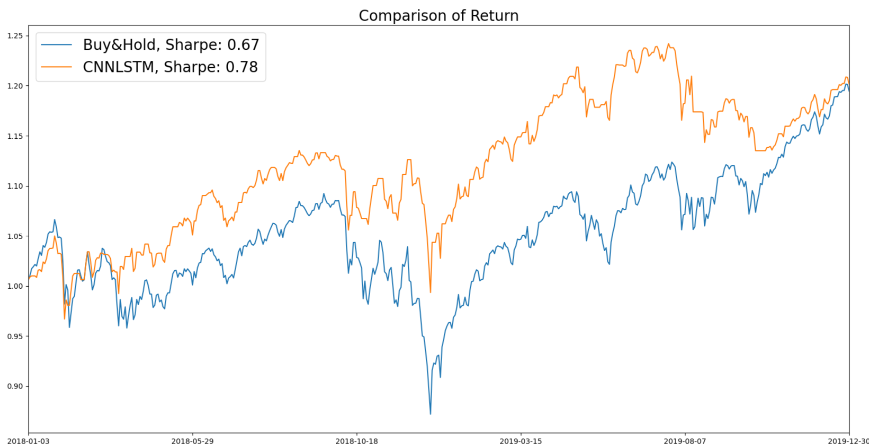
All results below follow the default setting listed below unless otherwise stated.

Table 3: Final Proposed Setting

Field	Setting
Architecture	Dilated CNN-LSTM
Context Window	10
Input	EMA Ratio Input Set
Target	1 Day Rate of Change
Ticker	SPY

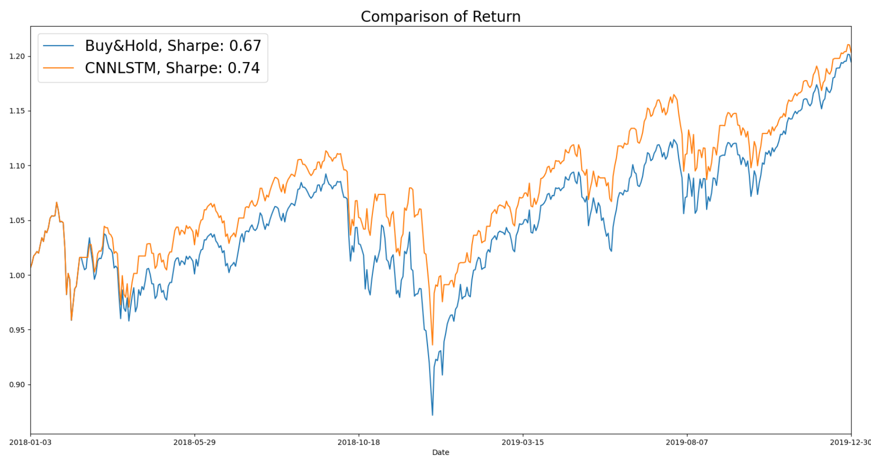


(a) SPY Trading - EMA ratio input - Dilated CNN-LSTM.

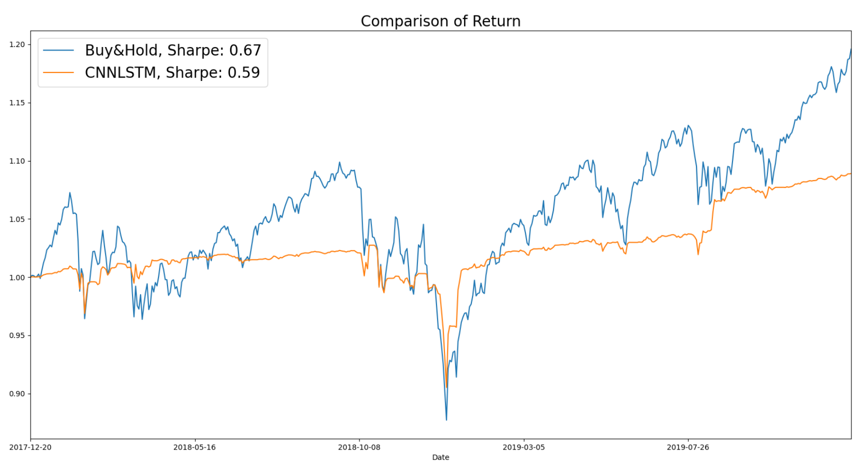


(b) SPY Trading - ROC Input - Dilated CNN-LSTM.

Figure 2: EMA ratio input vs ROC input

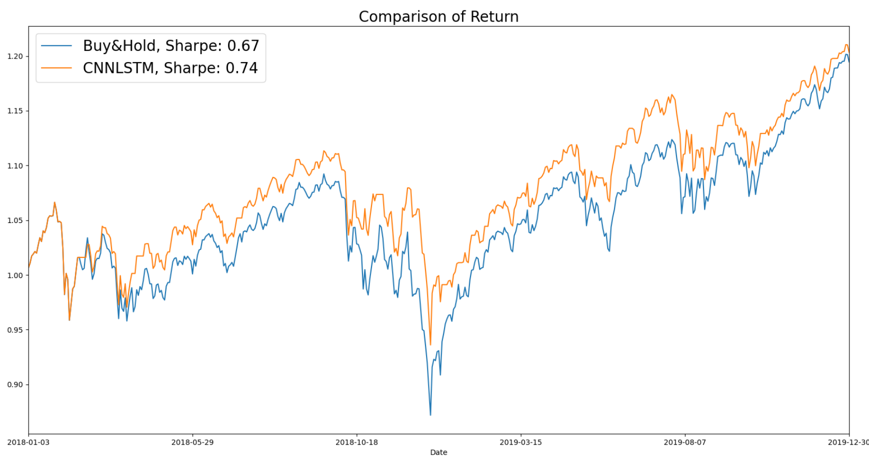


(a) SPY Trading - 1 day horizon - Dilated CNN-LSTM.

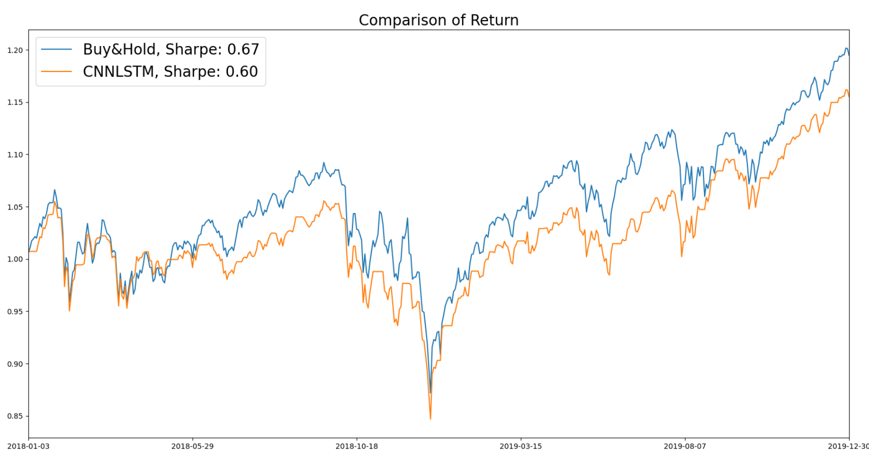


(b) SPY Trading - 10 day horizon - Dilated CNN-LSTM.

Figure 3: Next Day Prediction vs 10 Day Average Change Prediction

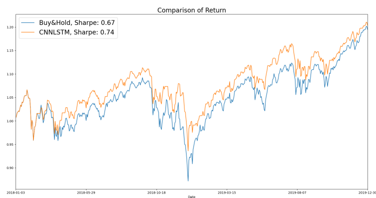


(a) SPY Trading - Dilated CNN-LSTM.

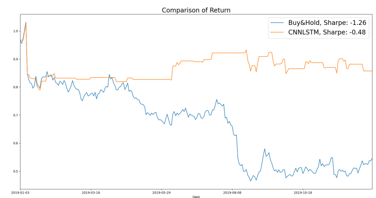


(b) SPY Trading - LSTM.

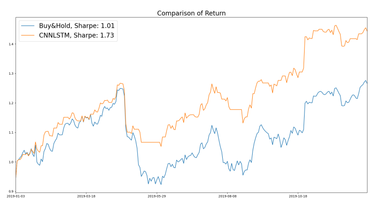
Figure 4: Dilated CNN-LSTM vs LSTM



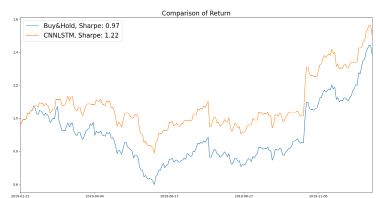
(a) SPY Trading - Dilated CNN-LSTM.



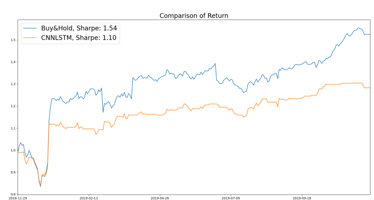
(b) Macy's (M) Trading - Dilated CNN-LSTM.



(c) Intel (INTC) Trading - Dilated CNN-LSTM.

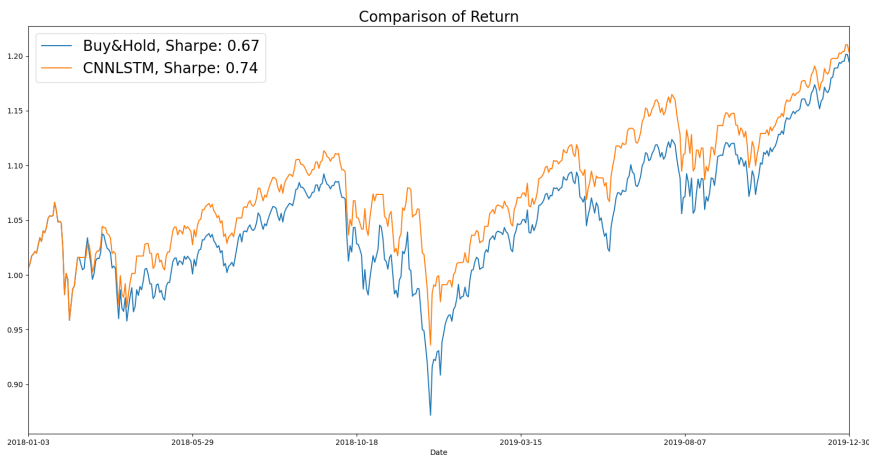


(d) Tesla (TSLA) Trading - Dilated CNN-LSTM.

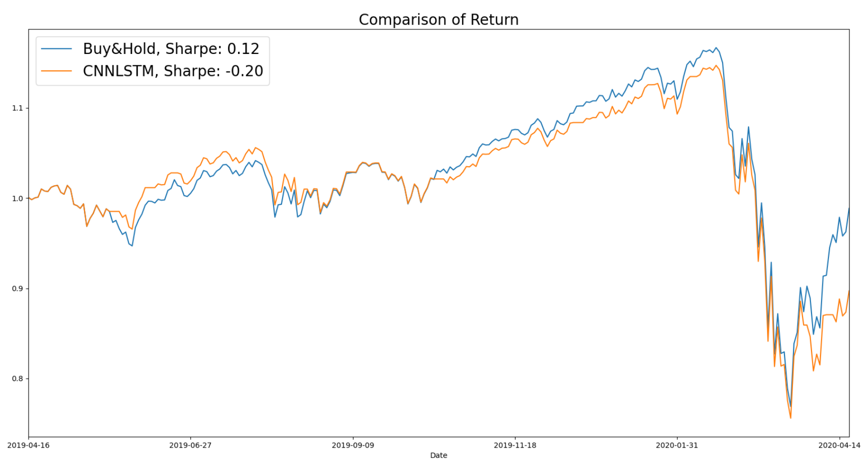


(e) Celgene Corp (CELG) Trading - Dilated CNN-LSTM.

Figure 5: Next Day Prediction Trading Performance on Different Stocks



(a) SPY Trading - 2018-01 to 2019-12 - Dilated CNN-LSTM.



(b) SPY Trading - 2019-04 to 2020-04 - Dilated CNN-LSTM.

Figure 6: SPY Trading Performance Variation on Different Time Periods.

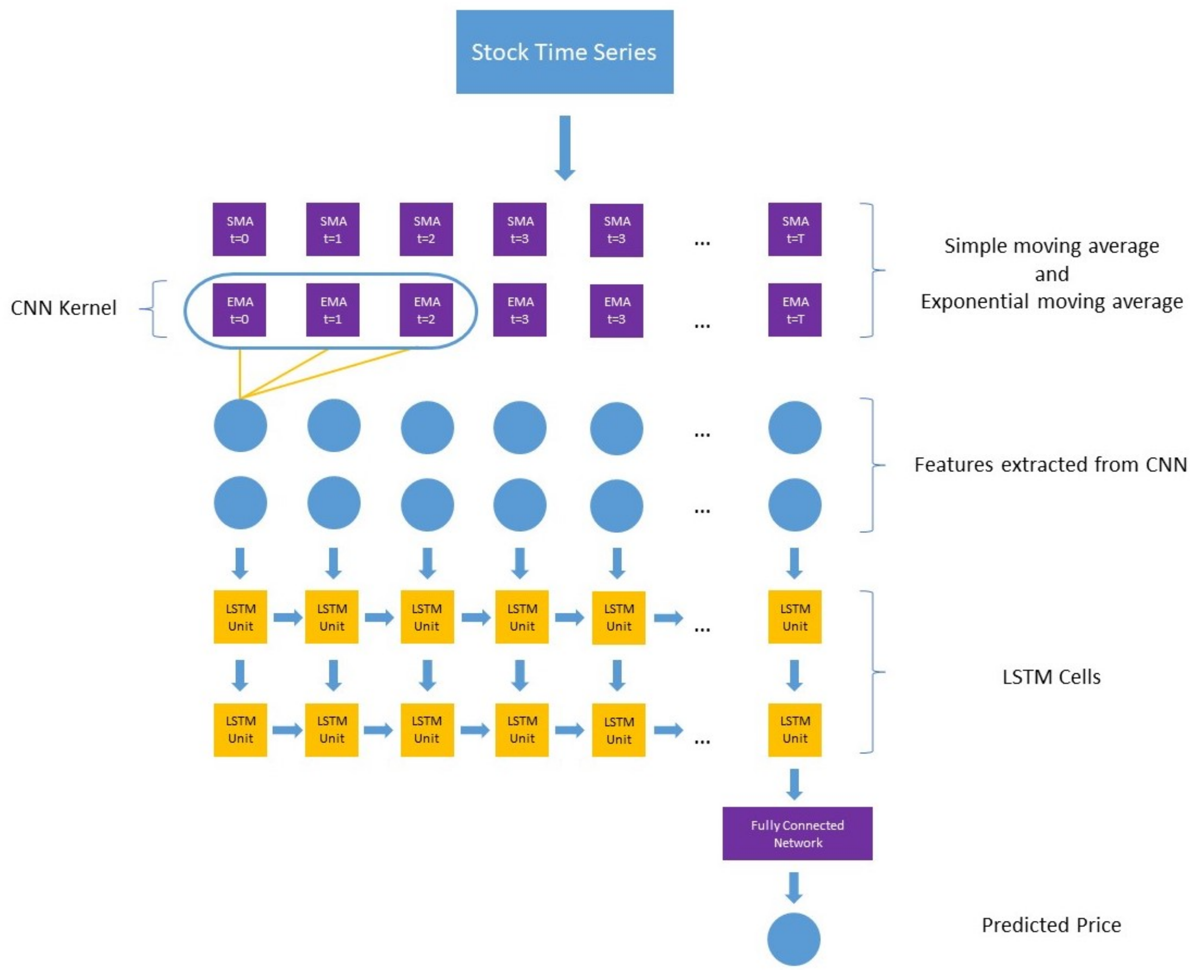


Figure 7: CNN-LSTM Model.

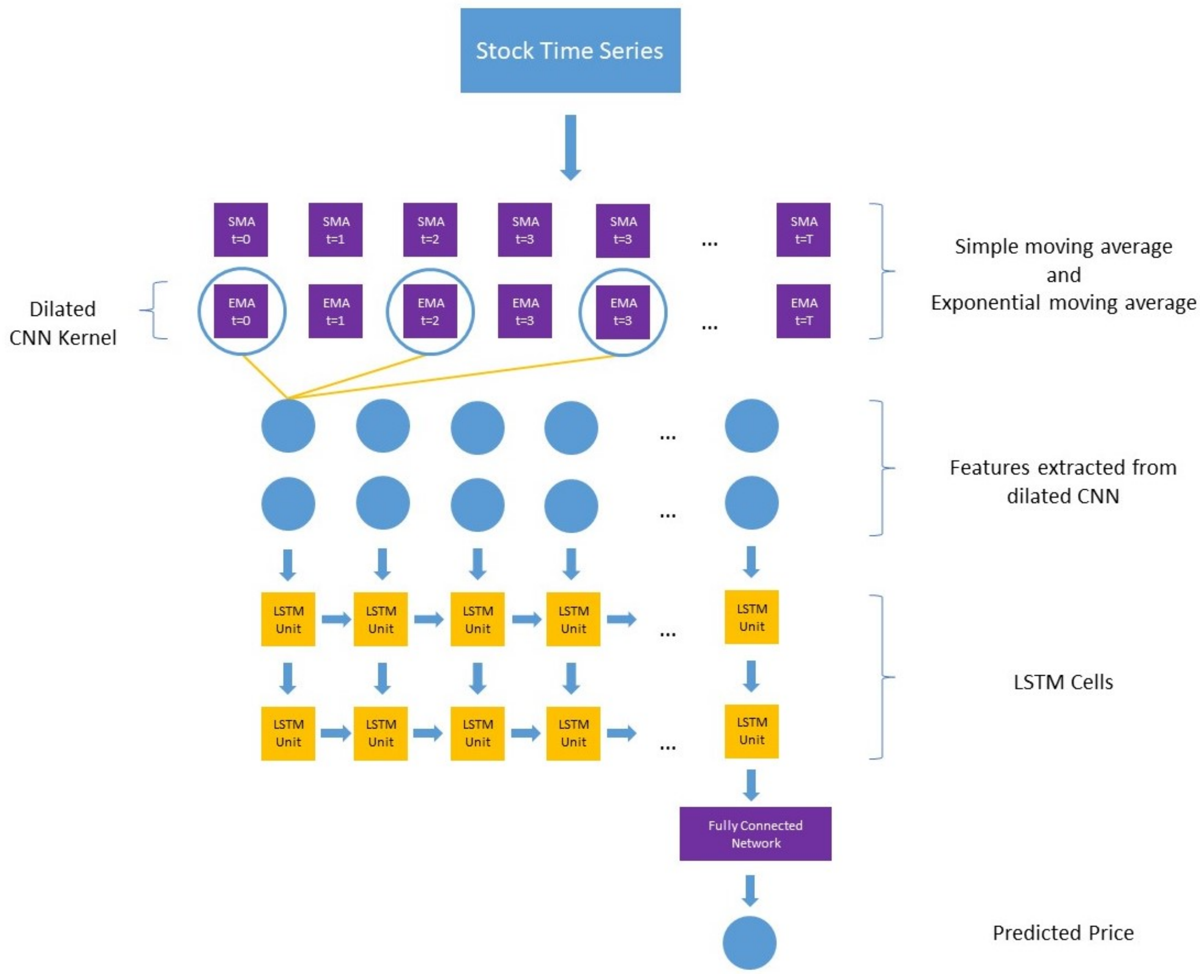


Figure 8: Dilated CNN-LSTM Model.

8.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 5.

Table 4: Input Features

Feature	Description	Type	Time Frame	
SMA	Simple moving average	Technical indicators	Jan,1 2020	Apr,20 2020
EMA	Exponential moving average	Technical indicators	Jan,1 2020	Apr,20 2020
F-SMA	Future Simple moving average	Technical indicators	Jan,1 2020	Apr,20 2020
F-EMA	Future Exponential moving average	Technical indicators	Jan,1 2020	Apr,20 2020
MOM	Momentum	Technical indicators	Jan,1 2020	Apr,20 2020
ROC	Rate of change	Technical indicators	Jan,1 2020	Apr,20 2020
ATR	Average true range	Technical indicators	Jan,1 2020	Apr,20 2020
BB	Bollinger bands	Technical indicators	Jan,1 2020	Apr,20 2020
BB	Relative strength index	Technical indicators	Jan,1 2020	Apr,20 2020
MACD	Moving Average Convergence Divergence	Technical indicators	Jan,1 2020	Apr,20 2020
%R	Larry Williams %R	Technical indicators	Jan,1 2020	Apr,20 2020
%K	Stochastic %K	Technical indicators	Jan,1 2020	Apr,20 2020
%D	Stochastic %D	Technical indicators	Jan,1 2020	Apr,20 2020
A\D	(Accumulation\Distribution) Oscillator	Technical indicators	Jan,1 2020	Apr,20 2020
IXIC	NASDAQ	Market Indices	Jan,1 2020	Apr,20 2020
GSPC	S&P 500 index	Market Indices	Jan,1 2020	Apr,20 2020
DJI	Dow Jones Industrial Average	Market Indices	Jan,1 2020	Apr,20 2020
NYSE	NY stock exchange index	Market Indices	Jan,1 2020	Apr,20 2020
RUSSELL	RUSSELL 2000 index	Market Indices	Jan,1 2020	Apr,20 2020
HSI	Hang Seng index	Market Indices	Jan,1 2020	Apr,20 2020
SSE	Shang Hai Stock Exchange Composite index	Market Indices	Jan,1 2020	Apr,20 2020
FCHI	CAC 40	Market Indices	Jan,1 2020	Apr,20 2020
FTSE	FTSE 100s	Market Indices	Jan,1 2020	Apr,20 2020
GDAXI	DAX	Market Indices	Jan,1 2020	Apr,20 2020
JPY=X	US to Japanese yen	Foreign exchange rates	Jan,1 2020	Apr,20 2020
AUD=X	US to Australian dollar	Foreign exchange rates	Jan,1 2020	Apr,20 2020
CAD=X	US to Canadian dollar	Foreign exchange rates	Jan,1 2020	Apr,20 2020
CHF=X	US to Swiss franc	Foreign exchange rates	Jan,1 2020	Apr,20 2020
CNY=X	US to Chinese yuan	Foreign exchange rates	Jan,1 2020	Apr,20 2020
EUR=X	US to Euro	Foreign exchange rates	Jan,1 2020	Apr,20 2020
GBP=X	US to British pound	Foreign exchange rates	Jan,1 2020	Apr,20 2020
NZD=X	US to New Zealand dollar	Foreign exchange rates	Jan,1 2020	Apr,20 2020
AAPL	Apple Inc.	US Company stocks	Jan,1 2020	Apr,20 2020
AMZN	Amazon.com Inc.	US Company stocks	Jan,1 2020	Apr,20 2020
GE	General Electric Company	US Company stocks	Jan,1 2020	Apr,20 2020
JNJ	Johnson & Johnson.	US Company stocks	Jan,1 2020	Apr,20 2020
JPM	JPMorgan Chase & Co.	US Company stocks	Jan,1 2020	Apr,20 2020
MSFT	Microsoft Corporation	US Company stocks	Jan,1 2020	Apr,20 2020
WFC	Wells Fargo & Company.	US Company stocks	Jan,1 2020	Apr,20 2020
XOM	Exon Mobil Corporation.	US Company stocks	Jan,1 2020	Apr,20 2020

Table 5: Stock Prediction Performance of Different Models without Economic Factors

Metrics	CNN + LSTM	D-CNN-LSTM	LSTM	LR	RF	KNN
Time horizon = 1						
Mean Square Error	0.616	0.615	0.618	0.275	N/A	N/A
Accuracy(%)	56.5	56.6	55.4	54.5	51.9	52.0
F1 score	0.719	0.719	0.707	0.705	0.510	0.641
Time horizon = 10						
Mean Square Error	0.717	0.724	0.722	0.260	N/A	N/A
Accuracy(%)	63.9	64.1	63.9	61.1	45.4	59.0
F1 score	0.775	0.775	0.775	0.758	0.372	0.739

Table 6: Stock Prediction Performance of Different Neural Models with Economic Factors

Metrics	CNN + LSTM	D-CNN-LSTM	LSTM
Time horizon = 1			
Mean Square Error	0.620	0.618	0.622
Accuracy	55.4	55.8	55.3
F1 score	0.719	0.722	0.718
Time horizon = 10			
Mean Square Error	0.716	0.716	0.714
Accuracy	65.2	67.3	66.1
F1 score	0.804	0.804	0.807

Table 7: Comparison of Different Context Window Size

Metrics	10	20	30	40
Time horizon = 1				
Mean Square Error	0.616	0.615	0.617	0.618
Accuracy	55.3	55.3	55.2	55.2
F1 score	0.721	0.721	0.720	0.720
Time horizon = 10				
Mean Square Error	0.718	0.719	0.716	0.716
Accuracy	69.7	67.5	69.9	65.6
F1 score	0.813	0.811	0.822	0.807