Deep Leaming for Stock Market Prediction Using Technical Indicators and Financial News Articles

Conference Paper · July 2018

DOI: 10.1109/IJCNN.2018.8489208

4 authors, including:

Manuel Ramon Vargas
Federal University of Rio de Janeiro

Alexandre Evsukoff
Federal University of Rio de Janeiro

Deep Learning for Stock Market Prediction Using Technical Indicators and Financial News Articles

Manuel R. Vargas
COPPE
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
manuelvargas760@gmail.com

Carlos E. M. dos Anjos
COPPE
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
carlos.menezes@poli.ufrj.br

Gustavo L. G. Bichara
COPPE
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
gustavo.bichara@coc.ufrj.br

Alexandre G. Evsukoff
COPPE
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
alexandre.evsukoff@coc.ufrj.br

***Abstract*—This work uses deep learning models for daily directional movements prediction of a stock price using financial news titles and technical indicators as input. A comparison is made between two different sets of technical indicators, set 1: Stochastic %K, Stochastic %D, Momentum, Rate of change, William's %R, Accumulation/Distribution (A/D) oscillator and Disparity 5; set 2: Exponential Moving Average, Moving Average Convergence-Divergence, Relative Strength Index, On Balance Volume and Bollinger Bands. Deep learning methods can detect and analyze complex patterns and interactions in the data allowing a more precise trading process. Experiments has shown that Convolutional Neural Network (CNN) can be better than Recurrent Neural Networks (RNN) on catching semantic from texts and RNN is better on catching the context information and modeling complex temporal characteristics for stock market forecasting. So, there are two models compared in this paper: a hybrid model composed by a CNN for the financial news and a Long Short-Term Memory (LSTM) for technical indicators, named as SI-RCNN; and a LSTM network only for technical indicators, named as I-RNN. The output of each model is used as input for a trading agent that buys stocks on the current day and sells the next day when the model predicts that the price is going up, otherwise the agent sells stocks on the current day and buys the next day. The proposed method shows a major role of financial news in stabilizing the results and almost no improvement when comparing different sets of technical indicators.**

Keywords — Deep learning, Recurrent neural network, Convolutional neural network, Long Short-Term Memory, Stocks forecasting.

I. Introduction

The aspiration of any investor is to forecast the market behavior aiming the best decision when it comes to buying or selling shares of stocks seeking to maximize his profits. This is a difficult task because market behavior is stochastic, volatile and influenced by many factors such as global economy, politics, investor expectation and others.

The random walk theory [1] introduces a hypothesis that stocks prices are defined randomly and therefore impossible to forecast. However, advances in artificial intelligence and the growth of available data have made it possible to forecast the stock price behavior with better performance than a random process [2]-[8].

There are three approaches to the information required to make a prediction. The first approach, technical analysis, is based on the premise that the future behavior of a financial time series is conditioned to its own past, thus technical indicators could help traders to interpret this behavior in their favor. The second approach, fundamental analysis, is based on company's financial reports and external information as political and economic factors. This information is taken from unstructured data as news articles, financial reports or even publishing in microblogs by analysts. Nofsinger [9] shows that in some cases, investors tend to buy after positive news resulting in a stress of buying and higher stocks prices; and after negative news, they sell, resulting in a decrease of prices. Finally, the third approach considers relevant all information coming from both, financial time series and textual data.

Prior works in this area focused on technical analysis. These works used different statistical techniques and artificial intelligence models to make a prediction based only on technical information [10],[11]. This approach has a limitation since the market reacts to external information that is not contained in the historical data used to extract the technical information.

Inspired by fundamental analysis, many authors propose the use of text mining techniques and machine learning techniques to analyze textual data and extract information that can be relevant to the forecast process [12]-[15]. The most relevant works in the area are reviewed in [16],[17]. Other works [3],[18] use hybrid models that combine text mining techniques with technical information. This approach outperforms other baseline strategies.

Recently, with more computational capabilities and the availability to handle massive databases, it is possible to use more complex machine learning models, such as deep learning models, which presents superior performance in traditional Natural Language Processing (NLP) tasks. The outstanding deep learning models are: CNN [19]-[22], RNN [23],[24], specifically the LSTM architecture [25],[26], and Recurrent Convolutional Neural Network (RCNN) [27],[28].

Some examples of deep learning models for financial time series forecasting are shown in [29],[30]. These authors apply a deep neural network model that uses as input events taken from financial news articles to forecast the direction of prices of a set of stocks and the S&P 500 index. The main characteristics of the work described in [30] are the event representation method and the CNN which models the influence of these events on stock prices behavior in short-term, middle-term and long-term.

From the works cited above it is possible to identify three key points for the construction of deep learning models. The first one is the definition of the prediction horizon, the second one is the temporal effect of a news document and the third one is the representation type of the information. Regarding the first point, daily prediction (intraday) is the most used. The authors in [29] show that the performance of daily prediction is superior than weekly and monthly prediction. In spite of that, the database used in this work does not have the granularity to allow intraday trading, so a daily approach was used. The second point refers to the time interval that news or events influence the stock prices behavior. If the objective is a daily prediction, it is indicated to use the news published the day before to the prediction day [29]. However, Ding et al. [30] show that even a combination of weekly events and events from the last month can present relevant information to daily prediction. Finally, in relation to the last point, previous works mainly use simple characteristics such as bags-of-words, noun phrases and named entities. Recently, other representation techniques such as word-embedding [31]-[33] and event-embedding [30] are used. They distinguish from previous methods because can represent complex characteristics of words or events with lower-dimensional dense vectors.

In this paper, two models are used to forecast daily directional movements of the stock price of Chevron Corporation (CVX), both receiving two set of indicators in different executions. They are named I-RNN for LSTM model only with the first set of indicators, I-RNN-2 for LSTM model only with the second set of indicators, SI-RCNN for the hybrid model composed by CNN and LSTM with the first set of indicators and SI-RCNN-2 for the hybrid model composed by CNN and LSTM with the second set of indicators.

The hybrid model used was proposed by [34] and has shown good results in this task in [39]. The model uses as input a set of technical indicators extracted from the target series and financial news titles published the day before the prediction day. It is applied a two-step process to represent each news in the data set: first, a word2vec model [32],[33] is used to generate a word representation and later an average of all the word vectors of the same title is performed, addressing sparsity in word-based inputs. The RCNN model aims to obtain advantages from both models: CNN and RNN. CNN has a superior ability to extract semantic information from texts in comparison with RNN and RNN is better to catch the context information and to simulate complex temporal characteristics. Finally, given the directional movements predictions of the stock price a trading agent decides when to buy or sell a stock.

II. Model Design

In this section we introduce the design of the recurrent convolutional neural network model (RCNN) to predict intraday directional-movements in financial time series using financial news articles and technical indicators as input. This model is named SI-RCNN and is shown in Fig. 1. The design has four stages: Input layer, convolutional layer, recurrent layer and output layer. These stages are described below.

A. *Input Layer*

The model uses two types of inputs, the first one is the technical indicator and the second one is the sequence of news titles. With the purpose of differentiating them, the input layers are renamed respectively as technical indicator layer and embedding layer.

The technical indicator layer takes as input two different sets of indicators. The first one, used in I-RNN and SI-RCNN, is a delayed sequence of seven technical indicators in chronological order as described in [3], the set of indicators is [42]:

- Stochastic: a momentum indicator comparing the closing price of a security to the range of its prices over a certain period of time. The %K corresponds to the current market rate for the currency pair and %D is equals to 3-period moving average of %K;
- Momentum: the rate of acceleration of a security's price. It is considered an oscillator and is used to help identify trend lines;
- Rate of Change: the speed at which a variable changes over a specific period of time;
- Williams %R: a momentum indicator that measures overbought and oversold levels. It compares the close of a stock to the high-low range over a period of time, typically 14 days;
- Accumulation/Distribution (A/D) Oscillator: a normalized momentum indicator that attempts to identify supply and demand by determining whether investors are generally buying (accumulating), or selling (distributing) a certain stock;
- Disparity 5: measures the relative position of the most recent closing price to a selected moving average and reports the value as a percentage.

This input is defined as a matrix $I \in \mathbb{R}^{7\times n}$, when n is the length of the delay window.

The second set, used in I-RNN-2 and SI-RCNN-2, uses some of the most common technical indicators on the market[42]. It takes a delayed sequence of five of those indicators as described below [42]:

- EMA (Exponential Moving Average): a moving average that gives more weight to the latest data. The model uses 8-day, 20-day and 200-day EMAs.
- MACD (Moving Average Convergence-Divergence): a trend-following indicator. It is calculated by subtracting the 26-day EMA from the 12-day EMA.

Then a 9-day EMA of the MACD is used as a signal line.

- RSI (Relative Strength Index): a momentum indicator that attempts to identify overbought or oversold conditions. It compares the average gain of up periods against average loss of down periods during a specific timeframe. In this paper we use a 14-day period and a simple moving average of 9 periods of the RSI.
- OBV (On Balance Volume): a momentum indicator that uses volume flow to predict changes in stock price. It is calculated by adding the volume when the closing price is higher than the day before or subtracting when it is lower. We then calculated a 20-day EMA of the OBV.
- BB (Bollinger Bands): a volatility indicator that encapsulates the prices by tracing an upper and a lower band using two standard deviations along a 21-day simple moving average.

This input is defined as a matrix $I \in \mathbb{R}^{17\times n}$, where n is the length of the delay window.

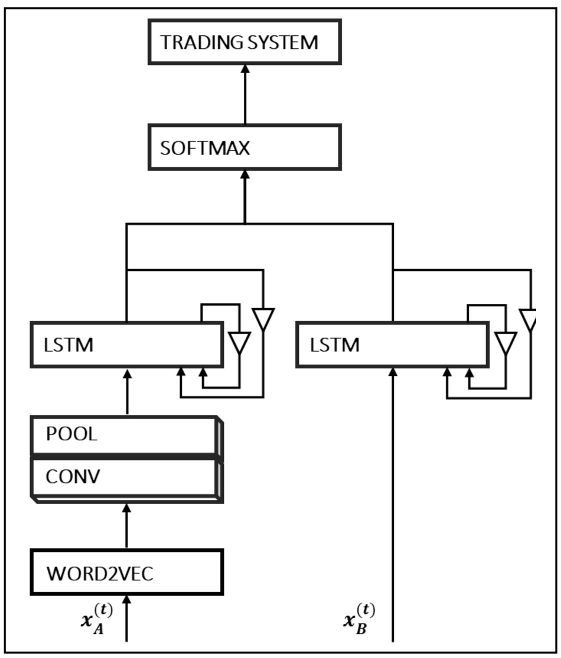

Fig. 1. SI-RCNN model architecture

The embedding layer takes a sequence of encoded sentences as input, this sequence corresponds to a set of titles of news articles from day t arranged in chronological order. The encoding of the sentence is performed in two steps. First, a word2vec model trained on the continuous bag-of-words architecture is used for generating word embedding, a pre-trained embedding vectors trained on part of Google News dataset (about 100 billion words) obtained in [44] was used in this work. This embedding is unique vectors of continuous values with length m for each word in the training corpus. The second step performs an average of all the word vectors in a title, so, a unique vector for the entire title is obtained, called sentence vector. To access each of these sentence vectors, each title in the data set is encoded using a one hot encoding method and finally, it is created an embedding lookup table.

The advantage of using the word2vec model is that the resulting word embedding vectors can capture linguistic regularities such as semantic and syntactic regularities [33],[44], which is a desirable characteristic in NLP tasks.

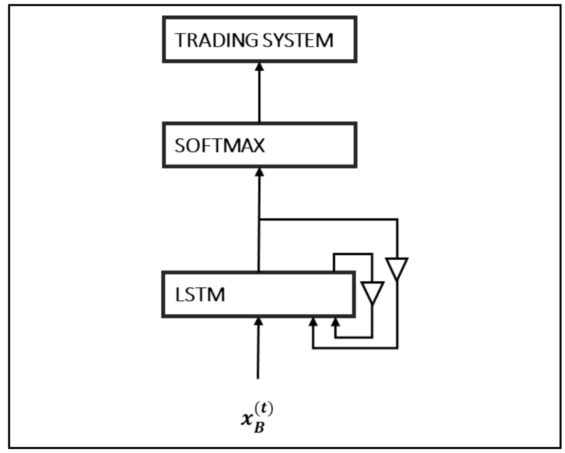

Fig. 2. I-RNN model architecture

B. *Convolutional Layer*

The next stage following the embedding layer is the convolutional layer which is composed by four consecutive operations: convolution, subsampling or pooling, activation and dropout. In this work, the convolutional operator is designed to perform one-dimensional convolution also known as temporal convolution. This operator can capture local information through the combinations of sentence vectors in a window.

Formally, given a sequence of news titles as input represented by their embedding vectors, $X = [x_1, x_2, \dots, x_L]$, and a filter unit, $W = [w_1, w_2, \dots, w_R]$, where $x_l \in \mathbb{R}^m$, $w_r \in \mathbb{R}^m$, L is the number of news titles in a day, m is the length of the embedding vector and R is the lenght of the filter window, the one-dimensional convolution is performed as follows:

$$q_j = W \cdot x_{j:j+R-1} + b \tag{1}$$

where $b \in \mathbb{R}^R$ is a bias term. This operation is applied to each possible window in the sentence to produce a feature map Q.

$$Q = [q_1, \dots, q_j, \dots, q_{L-R+1}] \tag{2}$$

Then, a temporal max-pooling is applied. It is similar as a spatial max-pooling module used in computer vision [31] but applied in one-dimensional input. This operation can capture the most important information in the sequence.

Finally, to introduce a non-linearity into the model we use a rectifier linear unit (ReLU) as activation function. This function is defined in (3).

$$h(q_j) = max\{0, q_j\} \tag{3}$$

Due to the high quantity of parameters in this layer, the model can be susceptible to overfitting, thus we use a regularization technique known as Dropout [35] with probability $p = 0.5$.

Equations (1) to (3) describe the process through which one feature is extracted from one filter. The model uses multiple filters with varying window sizes to obtain multiple features, therefore it is possible to explore different combinations of news titles. The output of this layer is defined as $h \in \mathbb{R}^{L-R+1 \times K}$, where K is the number of filters.

C. *Recurrent Layer*

Two separate recurrent layers are used: one following the convolutional layer, to be possible to interpret the output of the convolutional layer as a sequence of $L - R + 1$ time steps; and the second following the technical indicator layer. In both cases the RNN has the same purpose, to model the temporal characteristics in the input sequence. The LSTM model only is named as I-RNN and is shown in Fig. 2.

In this stage, a special RNN architecture named Long Short-Term Memory (LSTM) is used to introduce a new structure called memory cell. The key element of the LSTM is the cell state, C_t, which is controlled by 3 different gates, forget gate, input gate and output gate. The forget gate, f_t, decides which information of the previous cell state is remembered or forgotten. The input gate, i_t, decides which values of the cell state are updated by an input signal. Finally, the output gate, o_t, allows the cell state to have or not an effect on other neurons. One advantage of this structure is that it allows modeling long-term dependences in sequence data and preventing the vanishing gradient problem [43].

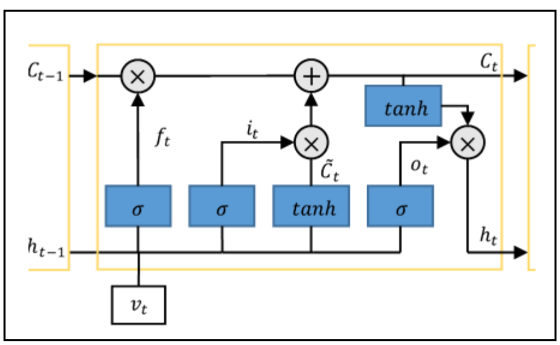

Fig. 3. LSTM architecture

The LTSM equations are shown from (4) to (9), where v_t is the input of the recurrent layer, h_t is the output of the recurrent unit and *W* are the weights matrices. Fig. 3 shows a detailed scheme of this architecture.

$$f_t = \sigma(W_{vf} v_t + W_{hf} h_{t-1} + b_f) \tag{4}$$

$$i_t = \sigma(W_{vi} v_t + W_{hi} h_{t-1} + b_i) \tag{5}$$

$$\tilde{C}_t = tanh(W_{vC} v_t + W_{hC} h_{t-1} + b_C) \tag{6}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{7}$$

$$o_t = \sigma(W_{vo} v_t + W_{ho} h_{t-1} + b_o) \tag{8}$$

$$h_t = o_t * tanh(C_t) \tag{9}$$

D. *Output Layer*

The last stage of the RCNN model is a traditional fully connected layer with softmax as activation function whose output is the probability distribution over labels. In this work, the objective is to forecast the direction of intra-day stock price movements of the Chevron Company, this direction is used to create a binary class label where a label [1,0] represents that the stock price will increase and label [0,1] represents that the stock price will decrease.

III. Trading Agent

Given the directional movements predictions of the stock price made by either the I-RNN or the SI-RCNN model, a trading agent decides about when to buy or to sell a stock. There are two ways in which the agent can make profit. The first one is when the stock price is going up and the agent buys a stock at the current price and sells when the stock price grow (long operation). The second one consists in selling a stock that is not currently owned (usually borrowed) at the current price and subsequently repurchasing it when the price is lower, this type of operation is known as Short-selling and is used for making profit when the price of the stock is going down.

The agent in this paper performs daily trading operations, which consists in make a set of two operations, one at the current day and another 24 hours later, both using the closing price. For example, if tomorrow the model predicts that the stock price is going up, the trading agent buys a stock on the current day at the closing price and sells this stock at the closing value the next day.

Those operations of buying or selling stocks have a cost (taxes) that can be fixed or variable, and may depend on the trading volume, investment volume, country of origin of the operation, among other factors. In this paper is assumed that the operation is executed in the USA due to the lower taxes. We have performed experiments using two different brokers: one with fixed cost per operation and another with a fixed cost for each trading stock. The first broker is US Fidelity®, which has a fixed cost of $4,95 for each operation and the second one is Interactive Brokers®, which has a cost of $0,005 for each stock bought or sold, with a minimum value of $1 for each operation and a maximum of $50. For simulation purposes, the investment was fixed at $10.000, which led to a mean cost of $2 for each set of operations, operating with Interactive Brokers® and $9,90 operating with US Fidelity®.

IV. Experiments

A. *Data Description*

The database used in this work consists of 106.494 news articles from Reuter's website corresponding to the period from October 20, 2006 to November 21, 2013. The main topic of all these articles is financial news. This dataset was created and released by [29]. Each news article consists of its title, content and publishing date. The publishing date is employed for news alignment with a corresponding financial time series. Ding et al. [29] carried out a set of experiments and showed that news titles are more useful to forecast than news contents. For this reason, the proposed model uses only news titles as input.

The CVX stock price series is selected for the experiments. This series was obtained through Yahoo Finance in the same period used to take the news articles. This series was selected with as most of news were directly related to it. The target output and the technical indicators were calculated based on the information of this series and used as input of the models.

The target output consists in a binary variable where a value $[1,0]$ represents that the close price in the day $t+1$ will increase compared with the closing price in the day t and a value $[0,1]$ represents that the close price in the day $t+1$ will decrease compared with the previous day.

Since the approach in this work is daily prediction, all the news titles of the same day are aligned in a unique instance, therefore each instance represents one single day. Then, it is applied a filter step, that consists in selecting from the news just the titles directly related to a CVX stock, for example, news that refers to the CVX company, oil and natural gas production, among others can be selected. The reason of using this filter step is that financial news of the general market can contain much irrelevant information, as confirmed in [29]. Finally, the set of news, technical indicators and outputs are aligned creating the pair input-output. The days without released news are ignored. Detailed statistics of this data set are shown in Table I.

TABLE I. STATISTICS OF DATASET

Dataset	Training	Validation	Test
Time interval	03/11/2006 29/06/2012	02/07/2012 11/03/2013	12/03/2013 19/11/2013
# Instances	1112	134	139
# Total Docs	13149	1976	2046

B. *Details of Implementation*

To train the word embedding we use the Word2vec model. The selected length of these word vectors is 300 and it is initialized with publicly available vectors that were trained on 100 billion words from Google News using the continuous bag-of-words architecture. Words not present in the set of pre-trained words are initialized randomly. Thus, the input of the embedding layer has dimension $[\mathrm{L} \times 300]$ when L is the number of news in a day t. It is important to note that the model uses only news from the present day to make a prediction of the next day.

In the case of the technical indicators, one model (I-RNN) uses a set of seven indicators arranged in a delayed sequence, the length of the delay window is defined as 5. The other model (I-RNN-2) uses a different set of five indicators with variation of the parameters that generates 17 features and the same delay window was used.

Regarding to the model parameters, we used three different filter windows: $[3 \times 300]$, $[4 \times 300]$ and $[5 \times 300]$. For each one of these types of filters we used 64 filter units, stride 1 was applied and a padding convolution performed. The window of the temporal pooling layer was set to 2. In this way, the output of the convolutional layer, h, has dimension $[((\mathrm{L}-2)/2)+1 \times 192]$. The recurrent layer on top of the output h, has 128 LSTM units. Finally, for modeling the sequence of technical indicators a one layer recurrent network with 128 LSTM units is used.

The algorithm used for training the model is the stochastic gradient descent (SGD), using momentum 0.9 and initial step size 0.1, implemented using Tensorflow.

V. Results and Discussion

The focus of this paper is to evaluate the capacity of the SI-RCNN and I-RNN models to execute trading operations in a simulated environment. After measuring the success of the model by validation accuracy, its profit is obtained according to the trading operations.

If the trading agent knows the correct price movements all the time (100% of accuracy) and using the trading strategy explained in the section III, the maximum possible profit is $153,003.70 in the training set, $10,416.98 in the validation set and $9,567.52 in the test set. This profit is calculated through 1112 days for the training set, 134 days for the validation set and 139 days for the test set.

TABLE II. COMPARISON OF ACCURACY RESULTS

Model	Training ACC (%)	Validation ACC (%)	Test ACC (%)
I-RNN	55.22	55.97	52.52
SI-RCNN	84.08	60.45	56.84
I-RNN-2	59.08	50.74	48.92
SI-RCNN-2	88.31	61.19	51.08

It is not possible to precisely predict every day movement correctly. In this paper, the accuracy of the best SI-RCNN in predicting the correct price movements is 56.84%, as shown in Table II, with this information as input of the trading agent and considering the lower brokerage fee, the agent is able to increase the initial investment by the factor of 13.94% in the test set. Even though it seems way lower than the best possible scenario, if compared to the buy-and-hold strategy (3.22%) it has an amazing performance.

It is important to note that the brokerage firm that has the best return, for this case, is Interactive Brokers® which was around \$2 for each set of operations while the second broker (US Fidelity®) was around \$9 for each set. Using the US Fidelity® brokerage firm there is a loss of US\$1,098.10 higher than Interactive Brokers®' in the test set, due to the 4.5 times higher brokerage. This loss occurs because the daily variations of the stock price are low, the variation mean is 0.6%, the max is 2.4% and the minimum is 0.008, as can be seen in Fig. 4 which shows the CVX prices time series during the test period. This makes some operations unfeasible to execute due to the high brokerage fee. The Table III summarizes the results for each model using the first set of technical indicators given by both brokerage firms.

Fig. 4. CVX Prices from the test set

TABLE III. PROFIT RESULTS FOR EACH FIRM

Model	Brokerage firm	Training US\$ (%)	Validation US\$ (%)	Test US\$ (%)
I-RNN	Interactive Brokers®	15,086.71 (150.87%)	1,634.66 (16.35%)	467.81 (4.67%)
I-RNN	US Fidelity®	6,301.91 (63.02%)	576.06 (5.76%)	-630.29 (-6.30%)
SI-RCNN	Interactive Brokers®	105,957.96 (1,059.58%)	2,013.03 (20.13%)	1,394.80 (13.94%)
SI-RCNN	US Fidelity®	97,173.16 (971.73%)	954.43 (9.54%)	296.70 (2.97%)

TABLE IV. COMPARISON OF PROFIT RESULTS

Model	Brokerage firm	Training US\$ (%)	Validation US\$ (%)	Test US\$ (%)
I-RNN	Interactive Brokers ®	15,086.71 (150.87%)	1,634.66 (16.35%)	467.81 (4.67%)
SI-RCNN	Interactive Brokers ®	105,957.96 (1,059.58%)	2,013.03 (20.13%)	1,394.80 (13.94%)
I-RNN-2	Interactive Brokers ®	19,262.61 (192.63%)	764.94 (7.65%)	-668.93 (-6.69%)
SI-RCNN-2	Interactive Brokers ®	120,879.42 (1,208.79%)	1,706.88 (17.07%)	-935.03 (-9.03%)

The Table IV shown in a comparison of profit results among the proposed models using the lower cost brokerage firm. The columns in this table consist in profits in US dollar and the percentage in relation to the investment over the training, validation and test set. It is worth mentioning that only the models using the first set of technical indicators are able to make profit in test set. This might be occurring given the lower accuracy obtained by the second set of indicators and also the possibility that they are mispredicting major movements.

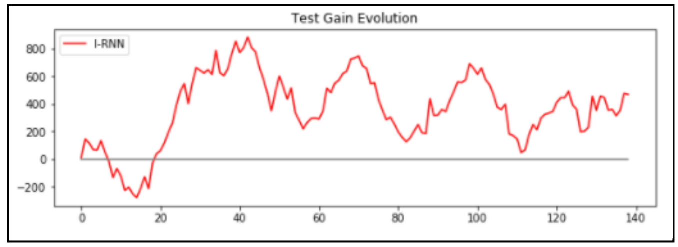

Fig. 5. I-RNN cumulative profits over the test set

The results shown on the last column of Table IV are obtained through the cumulative profits over the test set period, as can be seen in Fig. 5 to Fig. 8.

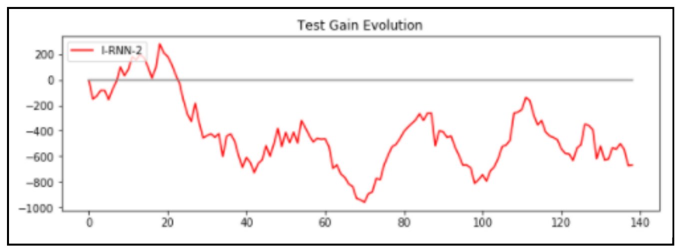

Fig. 6. I-RNN-2 cumulative profits over the test set

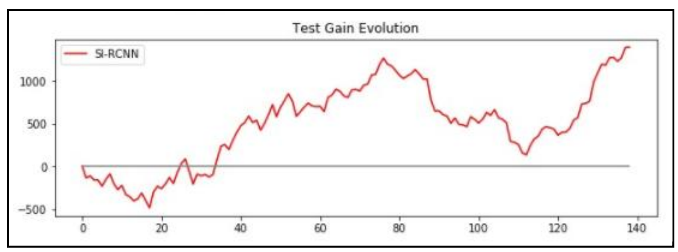

Fig. 7. SI-RCNN cumulative profits over the test set

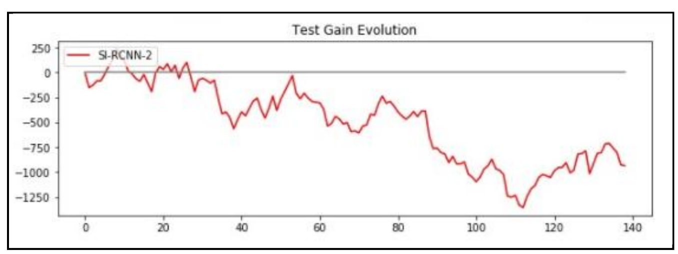

Fig. 8. SI-RCNN-2 cumulative profits over the test set

VI. CONCLUSIONS AND FUTURE WORKS

This work presented a deep learning model that combines a convolutional layer with a recurrent layer for daily stock price movement prediction and uses as input a combination of technical indicators and news titles. The RCNN architecture can model the local relation of a sequence of news titles and their temporal features. The output of this model is used by a trading agent with the purpose of performing trading operations.

Results presented in Table IV show that the SI-RCNN architecture could make a reasonable profit (13.94% in 8 months) when compared with buy-and-hold strategy which were 3.22% over the period. Finally, with the results shown in the Table II it is possible to confirm the positive influence of a hybrid input (news titles and technical indicators). This is shown by the lower performance of the I-RNN model, which uses only a set of technical inputs, compared to SI-RCNN model, that uses a hybrid input.

Furthermore, the Table II shows that it is relevant which set of technical indicators is chosen as input of the models, since the SI-RCNN model is better than SI-RCNN-2 that uses the first and the second set of indicators, respectively, and the I-RNN model is better than I-RNN-2.

Besides, results have shown that an accuracy above 50% does not necessarily leads to a profitable set of predictions, once SI-RCNN-2 achieves an accuracy of 51.08% and it has a loss of $935.03. This may be occurring due to the difference among daily price variations, brokerage cost and also because our models do not weight volatility, so both 0.1% up and 3.5% up are labeled as [1,0], possibly generating mispredictions on major movements.

It is important to note that deep learning models require a large amount of data, but when dealing with stock market prices predictions past occurrences does not necessarily have correlation with future behavior.

This work has been motivated by the successes of Deep Learning methods in Natural Language Processing task. The future work will include the use of test methods such as reported in [37] and [38] for making better embedding vectors for the news titles. Other research direction is the use of a reinforcement learning algorithm to train the proposed model on market simulation (trading simulation). These algorithms can train a model to create their own trading strategy. Preliminarily tests using Q-Learning have shown unsatisfactory results. And also another branch of evolution would be portfolio optimization.

Moreover, the financial time series is known by its volatility, in many cases occurs small changes in the series that can be interpreted as noise. Thus, some kind of trading strategy should be included such as stop gain and stop loss and also the elimination of small variations in order to make the model to focus only in events with significant variation on prices.

ACKNOWLEDGMENT

The authors are grateful to the Brazilian Research Council - CNPq for the financial support for this research. The authors are also grateful to NVIDIA Research Program that provided the hardware to run the experiments.

REFERENCES

[1] B. G. Malkiel, A Random Walk Down Wall Street, W. W. Norton, New York, (1973).

[2] J. Bollen, H. Mao, and X. Zeng, Twitter mood predicts the stock market, Journal of Computational Science, 2(1), 1-8 (2011).

[3] Y. Zhai, A. Hsu and S. Halgamuge, Combining news and technical indicators in Daily stock price trends prediction, Advances in Neural Networks, 1087-1096 (2007).

[4] G. Rachlin, M. Last, D. Alberg and A. Kandel, ADMIRAL: A data mining based financial trading system, IEEE Symposium on Computational Intelligence and Data Mining, 720-725 (2007).

[5] J. J. Wang, J. Z. Wang, Z. G. Zhang and S. P. Guo, Stock index forecasting based on a hybrid model, Omega, 40(6), 758-766 (2012).

[6] R. P. Schumaker and H. Chen, A quantitative stock prediction system based on financial news, Information Processing and Management, 45(5), 571-583 (2009).

[7] M. Khashei and M. Bijari, A novel hybridization of artificial neural networks and ARIMA models for time series forecasting, Applied Soft Computing, 11(2), 2664-2675 (2011).

[8] A. K. Nassirtoussi, S. Aghabozorgi, T. Y Wah and D. C. L. Ngo, Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment, Expert Systems with Applications, 42(1), 306-324 (2015).

[9] J. R. Nofsinger, The impact of public information on investors. Journal of Banking & Finance, 25(7), 1339-1366 (2001).

[10] H. Mizuno, M.Kosaka, H. Yajima and N. Komoda, Application of neural network to technical analysis of stock market prediction, Studies in Informatic and control, 7(3), 111-120 (1998).

[11] W. Leigh, R. Purvis and J. M. Ragusa, Forecasting the NYSE Composite Index with Technical Analysis, Pattern Recognizer, Neural Network, and Genetic Algorithm: a Case Studying Decision Support, Decision Support Systems, 1(32), 361–377 (2002).

[12] G. Gidofalvi and C. Elkan, Using news articles to predict stock price movements. Department of Computer Science and Engineering, University of California, San Diego, (2001).

[13] R. P. Schumaker, H. Chen, Textual analysis of stock market prediction using breaking financial news: The AZFin text system, ACM Transactions on Information Systems (TOIS), 27(2), 12 (2009).

[14] B. Wang, H. HUANG and X. WANG, A novel text mining approach to financial time series forecasting, Neurocomputing, 83, 136-145 (2012).

[15] H. Gunduz and Z. Cataltepe, Borsa Istanbul (BIST) daily prediction using financial news and balanced feature selection, Expert Systems with Applications, 42(22), 9001-9011 (2015).

[16] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah and D. C. L. Ngo, Text mining for market prediction: A systematic review, Expert Systems with Applications, 41(16), 7653-7670 (2014).

[17] R. C. Cavalcante, R. C. Brasileiro, V. L. Souzam J. P. Nobrega and A. L. Oliveira, Computational intelligence and financial markets: A survey and future directions, Expert Systems with Applications, 55, 194-211 (2016).

[18] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis and A. Jaimes, Correlating financial time series with micro-blogging activity, Proceedings of the fifth ACM international conference on Web search and data mining, 513-522 (2012).

[19] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882, (2014).

[20] X. Zhang and Y. LeCun, Text understanding from scratch, arXiv preprint arXiv:1502.01710, (2015).

[21] P. Blunsom, E. Grefenstette and N. Kalchbrenner, A convolutional neural network for modelling sentences, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, (2014).

[22] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning, Proceedings of the 25th International Conference on Machine Learning, ICML, New York, NY, USA, 160–167 (2008).

[23] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký and S. Khudanpur, Recurrent neural network based language model, INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 1045–1048 (2010).

[24] O. Irsoy and C. Cardie, Opinion Mining with Deep Recurrent Neural Networks, EMNLP, 720-728 (2014).

[25] X. Wang, Y. Liu, C. Sun, B. Wang and X. Wang, Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. ACL (1), 1343-1353 (2015).

[26] J. Xu, D. Chen, X. Qiu and X. Huang, Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification, arXiv preprint arXiv:1610.04989, (2016).

[27] C. Zhou, C. Sun, Z. Liu and F. Lau, A C-LSTM neural network for text classification, arXiv preprint arXiv:1511.08630, (2015).

[28] S. Lai, L. Xu, K. Liu and J. Zhao, Recurrent Convolutional Neural Networks for Text Classification, AAAI, 2267-2273 (2015).

[29] X. Ding, Y. Zhang, T. Liu and J. Duan, Using Structured Events to Predict Stock Price Movement: An Empirical Investigation, EMNLP, 1415-1425 (2014).

[30] X. Ding, Y. Zhang, T. Liu and J. Duan, Deep Learning for Event-Driven Stock Prediction, IJCAI, 2327-2333 (2015).

[31] Y. Bengio, R. Ducharme, P. Vincent and C. Jauvin, A neural probabilistic language model, *Journal of machine learning research*, 3(Feb), 1137-1155 (2003).

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, Proceedings of ICLR Workshop, (2013).

[33] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, 3111-3119 (2013).

[34] Y. L. Boureau, F. Bach, Y. LeCun and J. Ponce, Learning mid-level features for recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2559-2566 (2010).

[35] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, Journal of Machine Learning Research, 15(1), 1929-1958 (2014).

[36] R. Luss and A. d'Aspremont, Predicting abnormal returns from news using text classification, Quantitative Finance, 15(6), 999-1012 (2015).

[37] Q. V. Le and T. Mikolov, Distributed Representations of Sentences and Documents, ICML, 14, 1188-1196 (2014).

[38] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen and R. Ward, Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval, IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 24(4), 694-707 (2016).

[39] Vargas, M. R., de Lima, B. S., and Evsukoff, A. G. Deep learning for stock market prediction from financial news articles. IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 60-65 (2017).

[40] Sutton, R. and Barto, A. (1998) - Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA.

[41] Watkins, C. Learning from Delayed Rewards. Ph.D. Thesis, University of Cambridge, England (1989).

[42] Kirkpatrick, C. D. and Dahlquist, J.R. - Technical Analysis: The Complete Resource for Financial Market Technicians, FT Press, New Jersey (2010).

[43] Hochreiter, S.; Schmidhuber, J.; Long short-term memory. Neural Computation. 9 (8): 1735–1780 (1997).

[44] Google word2vec. Tool for computing continuous distributed representations of words, July 29, 2013. Internet: https://code.google.com/archive/p/word2vec/ [Accessed Dec 20, 2017].