Stock Market Prediction Based on Historic Prices and News Titles

Jinqi Tang
Fudan University
Shanghai, China
Tel: (+86)134-8260-8060
jqtang15@fudan.edu.cn

Xiong Chen *
Fudan University
Shanghai, China
Tel: (+86)135-0199-1244
chenxiong@fudan.edu.cn

ABSTRACT
Predicting the trend of stock market prices is a very challenging task, in that stock markets are complicated and can be influenced by a variety of factors. Despite the great difficulty, predicting the trend of stock market prices accurately is very meaningful and can bring a large amount of profit. In the past several decades, a lot of studies have been done on this problem. But most of the methods take only the historic prices data as the input, which is not enough for such a complicated problem. In this paper, a hybrid method taking both historic prices and the news as input is proposed. The hybrid model combines the best of two kinds of networks—— RNN-LSTM for time series data and CNN for abstract high-dimensional data. These two different kinds of networks are combined together to make a prediction. A set of experiments have been carried out to show the performance of the proposed method. The result obtained is promising, and the propose method achieves a great degree of accuracy in prediction and outperforms the baselines a lot.

CCS Concepts
• **Applied computing → Document analysis**

Keywords
Convolution neural network; Combined data; Hybrid algorithm; Long-short term memory; Stock market prediction;

1. INTRODUCTION
The stock market is one of the most important parts of the economy. The stock market influences the national and individual economy to a large extent. Predicting the trend of stock market is the most essential part in the task of finding the right time to buy and sell the share. Once the accuracy of predictions on stock market reaches to a certain level, the profit it brings can be abundant. That's why it has always been object of studies during the past several decades. The trend of Stock market can be affected by many factors, such as, politics, economics, environment, society etc. [1] Due to the complexity and chaos, solving the problem has been proved the to be a very difficult task [2].

ICMLT 2018, May 19--21, 2018, JINAN, China

ACM ISBN 978-1-4503-6432-4/18/05…$15.00
DOI: https://doi.org/10.1145/3231884.3231887

Stock markets Prediction is a long-discussed problem, there has been a constant debate on the predictability of stock markets. there are some researchers who believe the stock market is unpredictable. A theory called Random-walk hypothesis was raised [3], which claimed that a stock's future price is independent of its history. So that the tomorrow's price of a stock has nothing to do with today's price, but tomorrow's information.

On the other hand, Since the stock market was firstly introduced, a lot of researchers from different fields have attempted to predict the stock markets. For many years, traditional statistical prediction methods such as time series analysis, linear regression, chaos theory has been applied to solve the problem. But these methods were failure or partially successful due to the uncertainty of stock market.

To capture the complexity and non-linearity of stock market, various computational tools like neural network(NN), support vector machine(SVM), case-based reasoning(CR), genetic algorithms(GAs) and others [4] [5]. Among all methods used, neural network has been more and more used in recent years, and has shown better performance over other methods in many cases [6].

Stock market prediction using neural network has gone through many years. The first stock market prediction model based was implemented by White. The model he used is feed-forward neural network(FFNN), which can search for and decode non-linear regularities in asset price movements. Since the first attempt, many have taken part in the research of stock market prediction with NN models. Phua et al. used NNs with GA to predict the Singapore Stock Exchange Index. Ken-ichi Kamijo and Tetsuji Tanigawa proposed a recurrent neural network(RNN) approach to recognize stock price pattern [7] [8] [9]. FFNNs have done well in classification tasks, but in dynamical environments, RNNs can account for the history. However, since the units are connected through time steps, RNNs are hard to train. [10]

Most of these methods make a prediction with the historic price of the stock market as the input data. However, for the complexity of the stock market, the trend of future price cannot be predicted accurately based on the price data alone. In this paper, a hybrid algorithm which accepts two different parts of data is proposed. The two parts of data include the historic price data and the news data. The news data contains the politics, economic and society factors which can affect the stock market. The proposed method

shows better result than the methods using the historic price data alone.

The remainder of this paper is organized as follows: Section 2 presents the background and an overview of the theory concept which are base of the proposed method. Section 3 describes the propose method in detail, including the preprocessing step and the structure of the hybrid model. In Section 4, the experimental results and related analysis are presented. A conclusion of the article is included in Section 5.

2. BACKGROUND AND RELATED WORK

2.1 Recurrent Neural Network

In 1986, Jordan first introduced the modern definition of recurrent neural network [11]. Recurrent neural network(RNN) is a special structure of neural network, whose connections of unit form a directed cycle. This naturally grants it the ability to work with temporal data. The stock market price is a typical kind of time series data, so it's suitable to apply recurrent neural network on stock market prediction.

A simple example of recurrent neural network is shown in Figure 1.

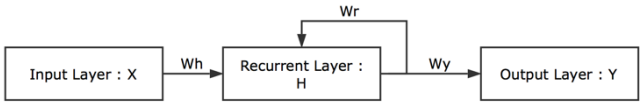

Figure 1. Typical Structure of Recurrent Neural Network

It's composed of three layers: the input layer, the recurrent layer and the output layer. The main difference between recurrent neural network and normal feed-forward neural network is that the output of recurrent layer at timestamp t will affect that at timestamp t+1. Figure 2. shows how recurrent neural network works through the time.

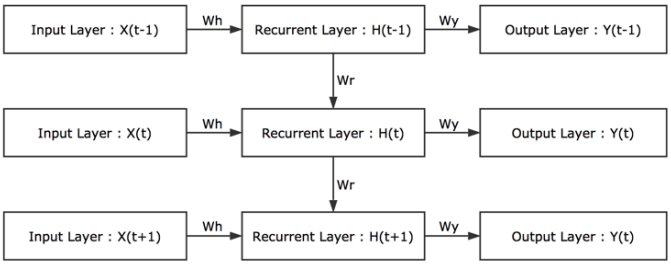

Figure 2. Recurrent Neural Network Structure Through Time

Use $x_i(t)$ to denote the i^{th} component of the input at timestamp t; $h_i(t)$ to denote the output of i^{th} unit of the recurrent layer; and $y_i(t)$ to denote the i^{th} unit of output layer. Then the relationship of the layers can be described by function (1), (2):

$$h_i(t) = \sigma\left(\sum_j w_{ij}^h x_j(t) + \sum_i w_i^r h_i(t-1)\right) \quad (1)$$

$$y_i(t) = \sigma\left(\sum_j w_j^y h_j(t)\right) \quad (2)$$

Here, w^h, w^r, w^y are the weights matrix between the layers, σ () is the activation function, usually tanh or sigmoid function in recurrent neural networks.

Recurrent neural networks are trained based on a comparison of output and the target. Because of the recurrent structure of RNN, gradient vanishing becomes a main problem of the RNN. Insufficient, decaying error backflow makes the training process of RNN take a very long time, so came the long short-term memory(LSTM) recurrent neural network.

2.2 LSTM Recurrent Neural Network

LSTM recurrent neural network is an improved structure of simple recurrent neural network. It was first introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997 [12]. The recurrent hidden layer of the LSTM-RNN is replaced by special units called memory blocks. These memory blocks contain memory cells in addition to special multiplicative unit called gates. Memory cells are self-connected and are used to store the temporal state of the network. Gates can be divided to three kinds: input gate, output gate and forget gate. These gates are introduced to control the flow of information [13].

The LSTM memory block works through following steps. Firstly, get current input state (denoted as $i(t)$) from current input data (denoted as $x(t)$) and previous hidden state (denoted as $h(t-1)$) according to function (3):

$$i(t) = \sigma(W_{ix}x(t) + W_{ih}h(t-1)) \quad (3)$$

Here, σ () is activation function, W is weight. Then, current input state is used to determine current value of input gate (denoted as $g(t)$), which is used to control the flow of input activations into the memory cell. Function (4) describes the process of this step.

$$g(t) = \sigma(W_{gi}i(t)) \quad (4)$$

After then, forget gate and output gate (denoted as $f(t)$ and $o(t)$) can be determined via function (5), (6):

$$f(t) = \sigma(W_{fi}i(t)) \quad (5)$$

$$o(t) = \sigma(W_{oi}i(t)) \quad (6)$$

Finally, we have the last two functions about the state update to complete the formulation of LSTM, here the current state of memory cell is denoted as $m(t)$:

$$m(t) = g(t)i(t) + f(t)m(t-1) \quad (7)$$

$$h(t) = o(t)m(t) \quad (8)$$

The structure of a memory block is shown in Figure 3.

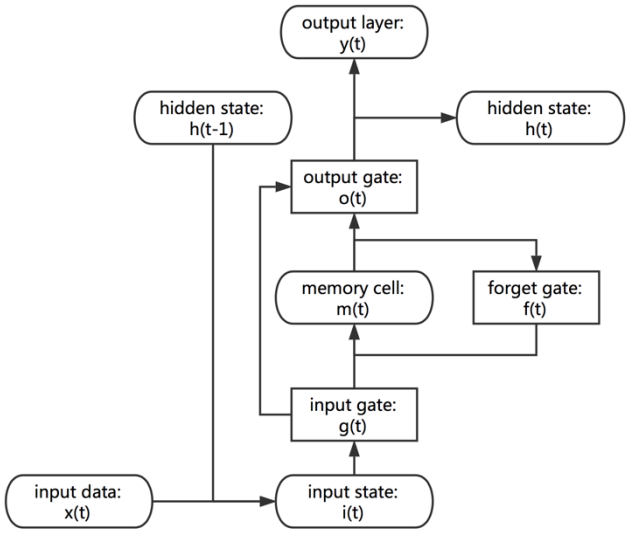

Figure 3. Structure of Memory Block

2.3 Convolution Neural Network

Convolution Neural Network, generally known as CNN, is another improved form of neural network. In ordinary feed-forward neural network, the adjacent layers are fully-connected, which will lead to a huge weight matrix if the input vector has a high dimension and cause the over-fit problem. CNN uses a special unit called filter or kernel to replace this part. Each unit of the output layer will only be connected to a small part of the input within the filter. The filters are shared by all the units in the same layer.

With such an architecture, convolution neural network requires less memory and is easier to train. Another advantage of this architecture is that it can focus on a small region of the input, thus find some local features. CNN has shown its strong ability in image classification field. Yoon Kim implemented convolution neural network in a sentence classification task and achieved excellent results on multiple benchmarks with little hyper parameter tuning. [14]

3. METHODOLOGH AND DEVELOPMENT

3.1 Data Collection

The stock market index we choose is the Dow Jones Industrial Average (DJIA). The whole dataset consists of two parts. The first part contains price data of DJIA from 2008/08/08 to 2016/07/01, including opening price, highest price, lowest price, closing price, trade volume and adjusted closing price(ACP). This part of dataset is downloaded directly from Yahoo Finance.

The other part of the dataset is every day's world news headlines crawled from Reddit World News Channel, which is relatively easier to acquire. They are ranked by reddit user's votes, and only the top 25 headlines are considered for a single date. The dates range from 2008/08/07 to 2016/06/30.

The purpose of the experiment is to predict whether the adjusted closing price will go up or not the next day.

3.2 Data Preprocessing

The training and testing samples are generated from the dataset.

The two parts of data constitute the input vector of each sample. The first part, historic price data, is a matrix which is every day's price data stacked in a certain number of time steps. That means, the price data of a certain number of days before the predicted day is considered. This part of the input can be defined as $(X_{i-k+1}, X_{i-k+2}, \dots, X_{i-1}, X_i)$, where X_i is a vector containing all price data of time step i, and k is the chosen window width. Since the Dow Jones Industrial Average price range from about 6500 to about 18000, which is difficult to processed directly with neural network, proper normalization is necessary before sent into the model. Another reason for the normalization is that if the stock market price keeps rising or dropping, and it may pose the problem that most values in the test set are out of the scale of the training set. Thus, the model has to predict some numbers which is never seen before. The method used here is linear normalization. Let x be a single feature value in the dataset, x_{mean} be the average value of all samples on this feature, x_{max} and x_{min} accordingly be the max and min value of all samples on the feature. Then the normalized value $\hat{x}$ can be computed via function (9):

$$\hat{x} = \frac{0.8*(x - x_{mean})}{x_{max} - x_{min}} \tag{9}$$

The normalization is added to every feature of the price data. The normalized data range from -0.8 to 0.8.

And the second part, news data, is the top 25 news headlines of the day before the target day. Some strange words and punctuation is removed ahead, and all letters are lower case. A dictionary is generated according to the headlines, each word that appears in any of the headlines is covered. Then the text data can be converted to vector of numbers which represents the index of each word in the dictionary [15]. Later the vectors are padded to the same length. Zeros are appended to vectors with length less than the padding length l. Those who with length longer than l are truncated. [16] [17]

The label, namely, the expected output represents whether the ACP of the target day is greater than that of the day before. A binary class is assigned to each entry of the dataset. '1' indicates that the price will go up on the following time step, and '0' means that it won't. Suppose the target day is time step i+1, and let y be the label of the samples, and ACP_i be the adjusted closing price of time step i. y is defined by function (10):

$$y = \begin{cases} 1 & \text{if } ACP_{i+1} > ACP_i \\ 0 & \text{if } ACP_{i+1} \le ACP_i \end{cases} \tag{10}$$

3.3 Classification Model

The structure of classification model is shown in Figure 4.

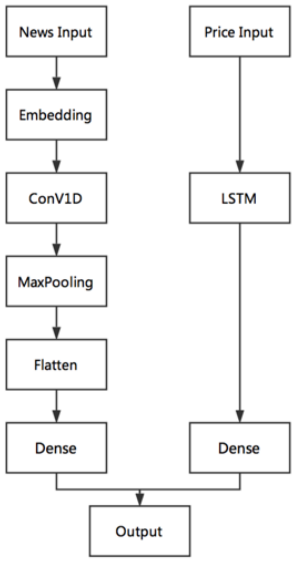

Figure 4. Proposed Model Structure

The two parts of inputs are processed separately, and then concatenated together to predict the result. For the price data input, a LSTM layer is added after the input layer. Since the LSTM network is suitable with the time series, it can capture the temporal feature of the input series, and output the features in a vector type. Following is a full-connected layer.

For the news data input, it will first go through an embedding layer, which can map the index of words in the dictionary to word vectors. Since the training text is not abundant, the embedding layer is loaded from a pre-trained network called" Global Vectors for Word Representation (GloVe)". GloVe is a word embedding method proposed by Stanford University, and each of the word vectors has a dimension of 50. Following the embedding layer, a convolution layer is added. This layer extracts the locality information of the word vectors, followed by a max-pooling layer

and a flatten layer. Then, the flattened vector is linked to a full-connected layer.

At last, the two full-connected layers are concatenated together and form a new mixed layer, which is directly connected to the output layer. The activation function of the output layer is sigmoid function, in which the output value ranges from 0 to 1. The values equal or greater than 0.5 is considered as 1, which means the price will rise the next day. The else are considered as 0, that is, the price will drop the next day.

In order to avoid over-fitting, dropout is introduced in some of the hidden layers with the drop rate being 20%, which implies that 20% of the units in hidden layer will be ignored in a single epoch and the weights related will not be updated. This kind of operation has been proved to be effective against over fitting. The drop rate is picked from a large range of numbers, through a large amount of experiments, which produces the highest mean accuracy.

4. EXPERIMENTAL RESULT

The Adjusted Close Price of DJIA Index between 2008/08/08 and 2016/07/01 is shown in Figure 5.

The price of every trading day is included in this dataset. There are altogether 1989 samples of price data. The former 1600 samples are split out as the training set, and the rest act as the validation set.

Table 1. Presents some characteristics of the dataset, including the value of the index on the first day and the last day of the period, the min and max value and the percentage of times the value goes up.

Table 1. Data Characteristics

Index	Start	End	Min	Max	Times goes up
DJIA	11734.32	17949.37	6547.05	18312.39	53.54%

Experiments has been carried out using the aforementioned preprocessing and model structure on the dataset. Simple long-short term memory(LSTM) model with the price data input, feed-forward neural network(FFNN) with the price data input and convolution neural network(CNN) based on news data are compared as the baselines. Figure 6. compares the result of proposed method to the baselines in terms of accuracy. The accuracy is defined as follows. If the predicted result is identical to the actual situation, it's a true prediction. Among them, if the result is rising, it's noted as true positive(TP). Otherwise, it's true negative(TN). And among the false prediction, if the prediction result is rising, it's false positive(FP), or else, it's false negative(FN). Then the accuracy is defined by function (11):

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

For each method, 40 times of experiments are carried out for each of the methods. All experimental results are presented in Figure 6.

As we can see in Figure 6., the accuracy of the proposed method outperforms that of the baselines. Mean accuracy of all methods are presented in Table 2.

Table 2. Mean accuracy of compared methods

Model	Hybrid Algorithm	LSTM	FFNN	CNN
Mean Accuracy	54.45%	52.64%	50.33%	51.38%

Through these experimental result, it can be observed that the addition of the news data and the model proposed is effective in the problem of predicting the move of stock market Index. The world news can certainly affect the trend of stock markets. Influential news, especially those related to the world politics and world economy, may cause the index to go up or go down dramatically.

Comparing the result of the proposed method and the single LSTM model with the price data input alone, the proposed hybrid model incorporated with the additional news data increases the mean accuracy by nearly 2%, which is a significant improvement. Also, we can observe from the results that the mean accuracy of the LSTM model using only the price data outperform that of the FFNN model. This somehow proves that the LSTM model is more suitable to handle the sequential data. On the other hand, the CNN model with the news data input alone achieves a result better than the FFNN model with the price data, which indicates that the news data can be used to predict the trend of the stock market.

The proposed method combines the LSTM model and the CNN model, in order to extract the information in both the price data and the news data, and achieves a satisfactory result.

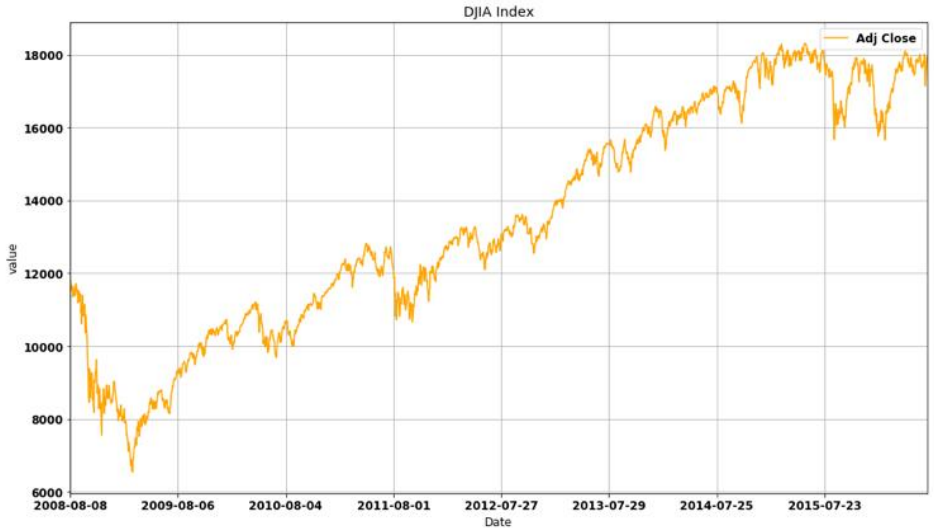

Figure 5. Adjusted Close Price of DJIA Index

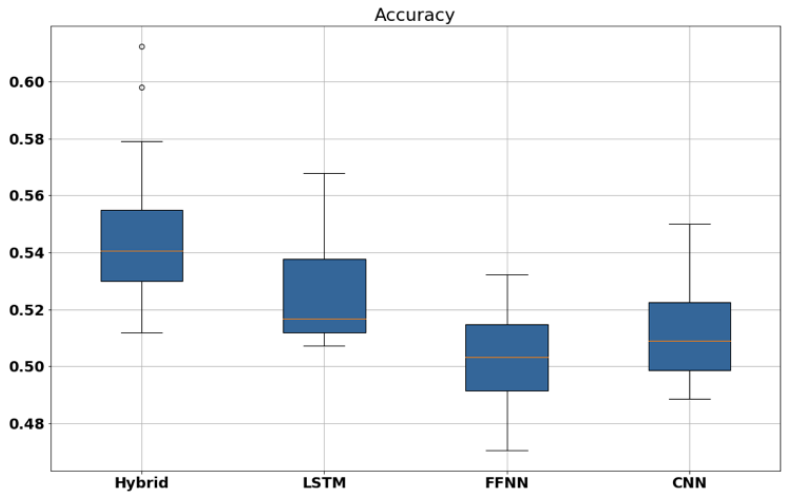

Figure 6. Accuracy compared to baselines

5. CONCLUSION

In this paper, a hybrid algorithm which can process the combined data of the historic price data and the news data is proposed. The historic price data goes through a LSTM network after a simple preprocessing step. And the news data, on the other hand, need some more complicated treatments before sent to a CNN network. At last, the two parts of the network are merged together to make a prediction. As the experimental result shows, the accuracy of the predictions made by the proposed method outperforms the baselines, which take the historic price data or the news data alone as the input.

The news data contain politics, economy, environment and society news, which are part of the factors that influence stock market. The proposed method is able to extract these factors from the text, converting these factors into features which can be used to predict the trend of stock market.

Through the experimental result, the LSTM network is proved to have advantage on processing temporal data. The LSTM network can find the pattern of the data even if it goes through a long time. Thus, this kind of neural network is quite suitable to handle stock market price. Along with the news data, the regular pattern of the mystery stock market can be quite clear.

In future, we will try to get a larger variety of data related to the problem and find a more accurate model to make the prediction. And the concept can be transferred to solve other complex problems.

6. ACKNOWLEDGMENTS

This paper is supported by Shanghai science and Technology Committee, No.17DZ1201605.

7. REFERENCES

[1] Billah, M., & Waheed, S. 2017. Stock market prediction using an improved training algorithm of neural network. *International Conference on Electrical, Computer & Telecommunication Engineering*. 1-4. IEEE.

[2] Nelson, D. M. Q., Pereira, A. C. M., & Oliveira, R. A. D. 2017. Stock market's price movement prediction with LSTM neural networks. *International Joint Conference on Neural Networks*. 1419-1426. IEEE.

[3] Musgrave, G. L. 1997. A random walk down wall street. *A Random Walk Down Wall Street*. 40(1), 18-23.

[4] Chen, K., Zhou, Y., & Dai, F. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market. *IEEE International Conference on Big Data*. 2823-2824. IEEE.

[5] Mahalakshmi, G., Sridevi, S., & Rajaram, S. 2016. A survey on forecasting of time series data. *International Conference on Computing Technologies and Intelligent Data Engineering*. 1-8. IEEE.

[6] Arasu, B. S., Jeevananthan, M., Thamaraiselvan, N., & Janarthanan, B. 2014. Performances of data mining techniques in forecasting stock index – evidence from India and US. *Journal of the National Science Foundation of Sri Lanka*. 42(2).

[7] Yoo, P. D., Kim, M. H., & Jan, T. 2005. Machine Learning Techniques and Use of Event Information for Stock Market Prediction: A Survey and Evaluation. *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*. 2, 835-841. IEEE.

[8] Qiu, M., Li, C., & Song, Y. 2016. Application of the Artifical Neural Network in Predicting the Direction of Stock Market Index. *International Conference on Complex, Intelligent, and Software Intensive Systems*. 219-223. IEEE.

[9] Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K., & Funaya, K. 2016. Robust Online Time Series Prediction with Recurrent Neural Networks. *IEEE International Conference on Data Science and Advanced Analytics*. 816-825. IEEE.

[10] O, Bernal. S, Fok. R, Pidaparthi. 2012. Financial Market Time Series Prediction with Recurrent Neural Networks. Citeseer.

[11] Jordan, M. I. 1997. Serial order: a parallel distributed processing approach. *Advances in psychology*. 121, 417-495.

[12] Hochreiter, S., & Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*. 9(8), 1735.

[13] Zaremba, W., Sutskever, I., & Vinyals, O. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:*1409.2329

[14] Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv*:1408.5882

[15] Le, Q. V., & Mikolov, T. 2014. Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 1188-1196.

[16] Hassan, A., & Mahmood, A. 2017. Deep learning for sentence classification. *IEEE Long Island Systems, Applications and Technology Conference*. 1-5. IEEE.

[17] Graves, A. 2012. Supervised Sequence Labelling with Recurrent Neural Networks. *Springer Berlin Heidelber*.