Forecasting Financial Markets using Deep Learning

Razvan Zanc, Tudor Cioara, Ionut Anghel

Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
zanc.razvan@gmail.com, {tudor.cioara, ionut.anghel}@cs.utcluj.ro

***Abstract* – Forecasting the behavior of financial markets represents an area of interest for many traders and investors due to the potential increase of capital which an accurate forecast can provide. The main objective of this paper is to predict the market behavior using Deep Learning techniques. We propose a stacked LSTM (Long Short-Term Memory) architecture on which we conducted several experiments on cryptocurrency and forex datasets. Our study reveals that in the context of financial markets, a high accuracy of a forecasted asset does not imply that the forecasted value will contribute positively to a profitable trading system.**

***Keywords* – financial markets, forecasting, deep learning, LSTM, cryptocurrency, forex.**

I. INTRODUCTION

The financial market is the market where financial securities are traded with low transaction costs at prices that correspond to demand and supply.

The currency market, Forex, in its modern form, resembles a lot with the other financial markets, but unlike the markets for other securities, foreign exchange trading is geographically decentralized. Forex [40] is for foreign exchange and simply refers to the conversion of one currency to another. From all financial markets in the world, the foreign exchange market has the most liquidities. The main actors on the foreign exchange market stage are the traders, which in turn are central and commercial banks, governments, financial institutions, or commercial companies [1]. The Bank for International Settlements, published in the 2016 the Triennial Central Bank Survey, where it was reported that the average daily turnover of the foreign exchange market reached $5.1 trillion in April 2016 [2]. For individuals who want to trade in the forex market, there is a segment dedicated for them called retail foreign exchange, which is a small fragment of the much larger market, where participants anticipate the exchange rate between two different currencies [3]. Due to the emergence of electronic trading platforms that offer access to individual traders to the global currency market, the retail foreign exchange segment has evolved, in 2016 reaching an average daily turnover of $282 billion, representing 5.5% of the entire foreign exchange market [2].

A cryptocurrency is a digital currency which was created using encryption techniques in order to secure and verify transactions, control the creation of units, having the goal of serving as a decentralized medium of exchange [4, 5, and 6]. Unlike centralized digital cash systems and central bank systems, cryptocurrencies use decentralized control which works empowering a distributed ledger technology, most common a blockchain, that serves as a public database of financial transactions [6, 7]. The first release of a decentralized cryptocurrency it is considered to be Bitcoin in the year 2009 [8]. Since the first release of Bitcoin, more than 4000 other cryptocurrencies have been created, reaching a daily turnover of approximatively $265 billion.

Algorithmic trading is the use of an automated system to perform transactions executed in a predetermined manner by a specific algorithm without human intervention. Algorithmic strategies are designed before the start of trading and are executed without the discretion of human traders [9]. The fact that its performance can be found on historical market data that is representative of future market data is the most important asset in creating an automated system.

The main objective of our work is the study a potential implementation of a financial prediction module, which can be integrated into a smart financial trading system which has the primary goal of trading different parities within forex and crypto markets in order to maximize profits.

The rest of the paper has the following structure: Section II presents related work, Section III presents a conceptual architecture of a smart trading system, Section IV presents the proposed financial forecast module implemented using a stacked LSTM architecture, Section V presents the experiments conducted on the proposed financial forecast model, Section VI presents a discussion based on the results of the experiments, and finally Section VII presents the conclusions of the present paper.

II. RELATED WORK

Forecasting variables using advanced modeling techniques represent one of the most in-depth topics studied in academic literature, placing themselves in the top ten most difficult data mining problems [11]. Finding an efficient forecasting method for time series-based datasets continues to be a difficult problem with many potential implementations across multiple industries [10].

Numerous approaches have been developed in the scientific literature to predict the behavior of financial markets. The Autoregressive Integrated Moving Average (ARIMA) [12] is one of the most common statistical technique used for forecasting challenges. The paper [27] analyzed the

978-1-7281-4914-1/19/$31.00 ©2019 IEEE

Autoregressive Integrated Moving Average to predict the evolution pf the Turkish Lira against the American dollar, showing that ARIMA's performance is not as satisfying as the performance offered by other techniques. The differentiation steps in ARIMA makes the data stationary which is a limitation in time series forecasting related challenges and also the differentiation operation amplifies the high frequency noise, which has a direct impact on the prediction's precision.

Other regression based automated techniques were applied to time series forecasting challenges, such as Support Vector Machine (SVM) [13, 14]. Support Vector Machine builds a decision function achieving linear regression, and it can perform better than the ARIMA model in those cases where the time series are very complex. The study presented in [28] using Australian forex data shows that the Support Vector Machine model's performance is better than other proposed machine learning techniques.

Recently, Artificial Neural Networks (ANN) became a popular approach when it comes to proposing solutions for forecasting time series. Given the high volume of algorithms based on ANN available in the academic literature, the identification process of a precise ANN algorithm that should be used for a forecasting task, should be based on an analysis on three aspects: the level of the complexity of the solution, the expected accuracy of the prediction, and the data features [17]. Given complexity and accuracy, the Feed Forward Neural Network predictor, in which the information is passed through the network only in the forward direction, is reported to offer the best results. The authors in [29] present a stock market forecasting method based on Artificial Neural Networks. Their study compares the performance of a neural network model against the performance offered by the ARIMA model, predicting the trend of future prices using the S&P 500 Index dataset. The results of the presented experiments show that the neural network system performs better than ARIMA only when the market conditions are stable. The neural network model achieved an accuracy of 23%, and the ARIMA model achieved 42% in more volatile scenarios. In [30], ARIMA's performance is compared against multiple neural network models for forecasting a currency pair's exchange rate in the foreign exchange market. The performance of the presented models was evaluated using the prediction's accuracy, the normalized mean square error and the mean absolute error. The results show that the neural networks perform significantly better than the ARIMA model, the neural network model reporting an accuracy of 80%.

Taking into consideration that the data features are time series, Recurrent Neural Networks (RNN) are more appropriate for forecasting challenges than Feed Forward Neural Networks [18]. In order to provide a temporal dependency between the processed sequences the RNN is retaining the state, and the output of the current time step becomes the input of the next time step [19]. Despite this fact, the greatest disadvantage of RNN can be seen during a long temporal dependency [19, 20]. To bypass this weakness, the authors in [21] proposed the Long-Short Term Memory (LSTM) algorithm as an improvement of the Recurrent Neural Network [20]. Several areas of research interest such as language modelling or speech recognition have adopted Recurrent Neural Network flavors such as Long-Short Term Memory and Gated Recurrent Unit (GRU) [41], in order to implement solutions for time series forecasting related challenges.

Many authors reported the use of the Long-Short Term Memory and the Gated Recurrent Unit algorithms for analyzing, modelling and predicting the stock market's behavior. In the work presented in [31], the authors developed a method for predicting the trend's movement direction of a stock considering the probability of a retracement relying a Self-Organizing Map Neural Network (SOM) and on an integrated Recurrent Neural Network. In [32] the authors employed a method based on ensemble learning which in turn is based on a multi-time-lag sampling technique in order to train a Gated Recurrent Unit. The authors evaluated their proposed method using datasets of annual electricity demand which were collected from the Australian energy market operator. In [33] the authors present the results of various stock market forecasting techniques, such as SVR, SVM, RNN and GRU. The dataset used for this comparative study included both technical features, like historical prices and technical indicators, and also fundamental features, like public sentiment scores. The results presented by the authors conclude that even if the even if the Gated Recurrent Unit algorithm performed better than the other proposed models, the accuracy offered by the SVM is better than the accuracy offered by the GRU algorithm.

The work presented in [34] proposes a Long-Short Term Memory based approach in order to tackle financial time-series prediction challenges. The authors studied the performance of multiple deep learning-based models with regards to their prediction accuracy. The work present in [35] proposes another architecture based on deep learning, which aims to forecast time series using the Long-Short Term Memory model in combination with Stacked Autoencoders (SAE). The results show that LSTM model produce a high prediction accuracy when integrated with SAE.

There are other LSTM based models available in the literature that are using text information in order to perform sentiment analysis having the objective predicting the stock market's fluctuations. In [36] the authors used a Naïve Bayes model to extract the investor's sentiment and other market factors from forum posts that were then inputted into a LSTM model constructed for financial forecasting. In the paper [37] the authors extracted information from financial news and articles and using their constructed dataset they evaluated the performance of a LSTM model and a SVM model. They tested the proposed models on multiple stock market indices, and the results show that the Long-Short Term Memory based model is more accurate. The authors of the paper [38] took on the challenge of predicting the Nifty Index by proposing solution based on the Long-Short Term Memory model while leveraging technical price features. After training their proposed model for 500 epochs, the authors reported a root mean squared, in terms of daily percent fluctuations, error equal to 0.0086.

The work presented in [39] proposes an Echo State Network (ESN), which is also a sub branch of the Recurrent Neural Network family, in order to predict the evolution of S&P 500 stocks. For the training of their proposed model these authors also use technical features and indicators like price variations, moving averages, and volume scores as input. The error of their proposed model was expressed in a daily price change percentage which was equal to 0.0027.

In contrast to with the available methods presented in the literature, our research is focused on building a deployable smart trading system which can be integrated with existing electronic trading platforms available for retail traders.

III. Conceptual Arhitecture Of The Intelligent Trading System

After studying the financial markets and the various algorithmic trading approaches, we propose a conceptual high-level architecture of an intelligent trading system that consists of four large modules: the electronic trading platform, the data collector module, the financial forecast module and the strategy generator as shown in Figure 1.

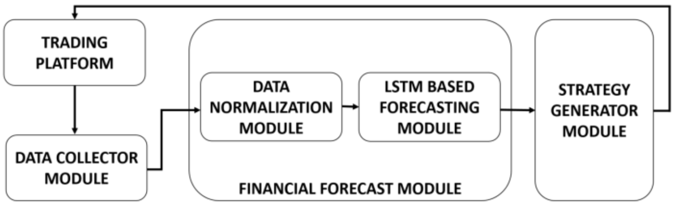

Figure 1. The proposed conceptual architecture of the intelligent trading system

Trading Platform – This module represents an existing electronic trading platform managed by a regulated broker, which exposes its functionalities though a consumable protocol. For our research we used the Oanda [42] platform in order to connect to the forex market which exposes its functionalities through a REST API [43], and in order to connect to the cryptocurrencies market we used the Bitfinex [44] electronic trading platform, which also exposes its functionalities through a REST API [45].

Data Collector Module – This module is responsible for collecting interest data, such as the price of one currency for sale and purchase by another currency at a given point in time. The Data Collector Module interrogates on regular basis a trading platform to acquire the data, so it must implement the communication protocols exposed by the electronic trading platform.

Financial Forecast Module – This module is responsible for forecasting the future price of a parity. The prediction is based on the data received from the data collection module.

Strategy Generator Module – This module is responsible for developing a trading strategy and placing financial transactions in the trading platform. Based on the predicted price received from the Financial Forecast Module, the Strategy Generator Module has the responsibility of placing multiple transactions at different time points, with different amounts, having the goal of maximizing profits. Since the Strategy Generator Module interacts directly with the electronic trading platform, the placing of transactions must be done using the protocol exposed by the trading platform.

As can be seen in the proposed conceptual architecture presented in Figure 1, the Financial Forecast Module is placed in the middle of the intelligent trading system. Financial forecasting is a difficult problem studied in detail, because the assessment of a currency is influenced by many factors, both economic and human. The Financial Forecast Module must be able to process large sets of data within a short period of time. Since both markets, the forex market and the crypto market are volatile markets, the data collection module must interrogate the trading platform multiple times per minute, in order to acquire data that can be used for forecasting the future price of a specified asset. The Financial Forecasting Module is limited by the execution time because it will provide a time-based prediction, which in turn is passed to the Strategy Generator Module which also requires a certain amount of execution time to develop the strategy with maximum profit potential. If the execution time of the Financial Forecasting Module is too high, the scenario where predictions are no longer useful because they correspond to past periods of time is very likely to occur.

IV. LSTM Based Forecasting Module

The Financial Forecast Module works based on the data received from the Data Collector Module. The Data Collector Module interrogates the trading platforms at a 10-second interval in order to acquire the features of interest.

The cryptocurrencies features gathered from the Bitfinex [44] platform are as follows:

- Ask: Represents the price at which the market is willing to sell a currency. A trader can buy the base currency at this specified price [46].
- Bid: Represent the price at which the market is willing to buy a currency. A trader can sell the base currency at this specified price [46].
- High: Represent the highest price of a currency from the last 24 hours [47].
- Low: Represents the lowest price of a currency from the last 24 hours [47].
- Last: Represents the price at which the last transaction occurred [47].
- Volume: Represents the total amount that has been traded within the last 24 hours [47].
- Orderbook: Is presented in equation (1), where V_{ask} represents the total volume that the participants are willing to buy from the last 100 transactions, and V_{bid} is the total volume that the participants are willing to sell from the last 100 of transactions.

$$Orderbook = \frac{V_{bid} - V_{ask}}{V_{bid} + V_{ask}} \quad (1)$$

The interest attributes of forex parities collected by the Data Collector Module from the Oanda [42] trading platform the Bid and Ask prices.

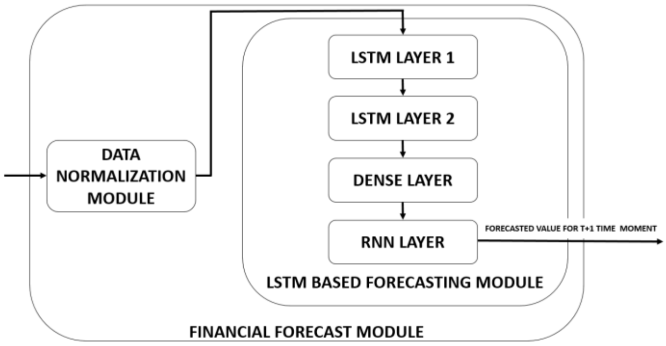

Figure 2. Conceptual architecture of the Financial Forecast Module

Once the data is collected, it is sent to the module responsible for the normalization. The Data Normalization Module normalizes data using the MinMaxScaler to bring all data to a range of [0, 1] or [-1, 1] if there are negative values.

$$MinMaxScaler(x_i) = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2)$$

The normalized data is sent as input to LSTM Based Forecasting Module whose primary objective is to predict the Last feature for cryptocurrencies parities collected from the Bitfinex [44] platform, and to predict the Bid feature for forex parities collected from the Oanda [42] trading platform for the next time step.

Feed-Forward Neural Network expects that both, input and output data, to be represented two-dimensionally, that is data of the type [number of the observations, size of the input data], which determines the dataset size in a Feed-Forward Neural Network to be equal to the number of observations, each observation to correspond to a number of columns which is has to be equal to the size of the input data. Likewise, the date outputted by a classical Feed-Forward Neural Network has also to be represented in two dimensions, having the form [number of observations, size of input]. Instead, the data processed by a Recurrent Neural Network, correspond to time series entries, and are represented using three dimensions: an extra size for the time property is added. The input data of a Recurrent Neural Network has the form [number of the observations, size of the input, size of the time series], and the output data are in the form [number of observations, output size, timer series size].

The LSTM Based Forecasting Module is built using two stacked LSTM layers, a Dense layer, and a Recurrent Neural Network for output layer. The LSTM Based Forecasting Module learns at a rate of 0.05, and it uses the Stochastic Gradient Descent algorithm for the purpose of optimization. The weights of the LSTM Based Forecasting Module are initialized using the Xavier method, ensuring that the values of the weights are not too small, but not too large to in order to accurately propagate signals. The authors of the work presented in [26] suggested initializing weights in a zero-zero distribution:

$$Var(W) = \frac{2}{n_{in} + n_{out}} \quad (3)$$

The essential elements of a Long-Short Term Memory unit are the memory cell, the input gate and the forget gate. The memory cell's internal state is processed by the input and forget gates. Presuming that the input and forget gates closed, the internal state of the memory cell will not be changed from one-time step to another [22]. The objective of using the gating strategy is to keep the information across multiple steps of time, and to allow consistent passing of the gradient through the required time steps, overcoming the Vanishing Gradient [25] problem which is common in Recurrent Neural Network based approaches.

A Long-Short Term Memory layer expects as input a non-fixed vector and one output value. The layer's output is influenced by the vector received as input and it's also influenced by the history of all inputs received from the recurrent connections. The layer is influenced by the history of inputs through the recurrent connections. The internal state of the LSTM unit is a hidden vector which is updated every time when an input is sent to the layer [21, 22]. Long-Short Term Memory models are leveraging supervised learning in order to update the weights, which is accomplished by training on only one input vector at a time sampled out from an entire sequence of vectors, which in turn will end up as activations for the input nodes [21].

The first step of the Long-Short Term Memory model is the decision of which piece of information will be discarded from the cell's state which is taken by the forget gate. The second step is represented by the decision of which piece of new information should be remembered in the cell's state. This decision is done in a two-step process: Firstly, the input gate determines which values should be updated. Secondly, the tanh layer generates a vector that contains new values and the old cell's state, C_{t-1} is updated, into a new state, C_t. The last step is to calculate the output, which will be a filtered version of the current state [25].

The first LSTM layer of the proposed LSTM Based Forecasting Module consists of 256 hidden units, and uses the tanh function as the activation function, and the Hard-Sigmoid function in order to activate the gates. For regularization, the first LSTM layer uses the Dropout mechanism, where during training some neurons are randomly ignored, valued at 0.2.

The second LSTM layer consists of 256 hidden units, and uses the tahn function as an activation function, and the Hard-Sigmoid activation function in order to activate the gates. For regularization, the second LSTM layer also uses the Dropout mechanism, valued at 0.2.

The third layer is a Dense layer, which is constructed using a fully connected Feed-Forward Neural Network that contains 32 hidden units and uses the relu function for activation.

The last layer of the proposed LSTM Based Financial Forecast Module consists of a Recurrent Neural Network layer that outputs a single neuron which provides a three-dimensional time series output. This last level uses the Identity function as an activation function.

The proposed LSTM Based Financial Forecast Module uses L2 (Ridge Regularization) to prevent over-fitting, the lambda parameter being 10^{-4}. The L2 regulation adds a penalty equal to the sum of the square value of the coefficients and forces the parameters to be relatively small, the higher the penalty, the lower coefficients**.**

Because the LSTM Based Financial Forecast Module to predict the Last feature for cryptocurrencies and the bid feature for forex parities at the next time step on the basis of previous information received from the Data Collector Module that collected information at every 10 seconds, the value 22 was chosen to be the length of the time series.

Because the Data Collector Module is offering as input high volume data, the LSTM Based Forecast Module is trained with fractions of the dataset using the mini-batch technique. For example, with the mini batch size = 64, it is assumed that the start the time is t = 0, the input data has the form [number of observations, input size, time series size], where number of observations = 64, input size = 7 time series size = 22. The output has the form [number of observations, size of output, time series size], where the number of the observations = 64, output size = 1 (the prediction of the Last or Bid feature at the next time step, i.e. over 10 seconds).

The LSTM Based Financial Forecast Module is trained using the Truncated Backpropagation Through Time algorithm with a window length equal to 22. The training of Recurrent Neural Networks can be highly computational, the traditional approach to their training being the use of the Backpropagation Through Time (BTT) algorithm. Backpropagation Through Time is very similar to the classical backpropagation, using a chaining rule that constructs the gradients based on the structure of the network's connections, but the gradients are also passed back from the next step to the current one [26].

Making more frequent updates to the parameters helps in speeding up the training of a Recurrent Neural Network. Truncated Backpropagation Through Time divides the forward and backward passes into more small and manageable operations. The Truncated Backpropagation Through Time presented in Figure 3 performs a 22-time steps pass forward and then it turns around and perform another 22 time steps pass, but this time it is done backwards. After these two passes are completed, the parameter is then updated [22, 26].

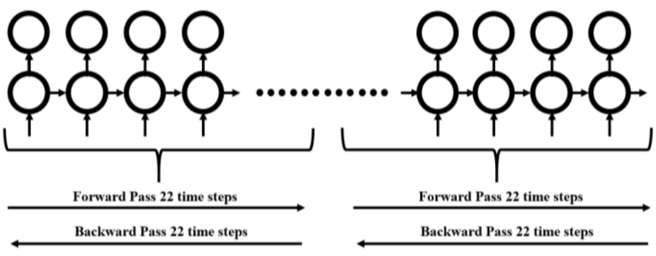

Figure 3. Truncated BPTT

The academic literature is considering the Truncated Backpropagation Through Time algorithm the most practical approach for training Recurrent Neural Networks and their variants, due to the fact that the learning long-term dependencies is done with a less computational cost rather than using the classical Backpropagation Through Time technique.

V. EXPERIMENTAL RESULTS

The proposed LSTM Based Financial Forecast Module presented in the previous chapter was evaluated using 2 cryptocurrency parities: BTC/USD and DASH/USD, and 2 parities for the forex market: EUR/GBP and EUR/USD. All tests were executed using a computer with an Intel Core I5-4200U processor with 2.3 GHz, 12 GB RAM, without dedicated video card.

The experiments were evaluated using the Root Mean Square Error normalized by the standard deviation on the test data set.

$$NRMSE(x) \frac{\frac{\sqrt{\sum_{i=1}^{n}(x_i-\hat{x})^2}}{n}}{\sigma(x)} \quad (4)$$

A. BTC/USD

For the BTC/USD parity test, the multi-layer neural network presented in the previous chapter was trained using the following parameters:

- Complete data set: 36001 points (approximately 124 hours)
- Features used in training: ask, bid, high, low, last, volume, order book.
- Training data set: 90% of full data set, 32401 points.
- Test data set: 10% of the full data set, 3577 points.
- Number of training epochs: 100.

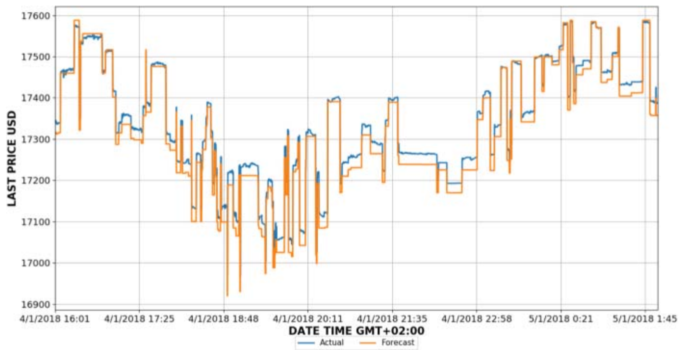

Figure 4. BTC/USD - Actual vs. Forecast

The run time of the BTC/USD test was approximately 12.50 hours, resulting in a NRMSE = 7.59.

B. DASH/USD

For the DASH/USD parity test, the multi-layer neural network presented in the previous chapter was trained using the following parameters:

- Complete data set: 36001 points (approximately 124 hours)
- Features used in training: ask, bid, high, low, last, volume, order book.
- Training data set: 90% of full data set, 32401 points.
- Test data set: 10% of the full data set, 3577 points.
- Number of training epochs: 100.

Figure 5. DASH/USD - Actual vs. Forecast

The run time of the test was approximately 16.14 hours, resulting in an error NRMSE = 0.18.

C. EUR/GBP

For the EUR/GBP parity test, the multi-layer neural network presented in the previous chapter was trained using the following parameters:

- Complete data set: 19470 points (approximately 54 hours)
- Features used in training: ask, bid.
- Drive data set: 90% of full data set, 17523 points.
- Test data set: 10% of the full data set, 1947 points.
- Number of training epochs: 100.

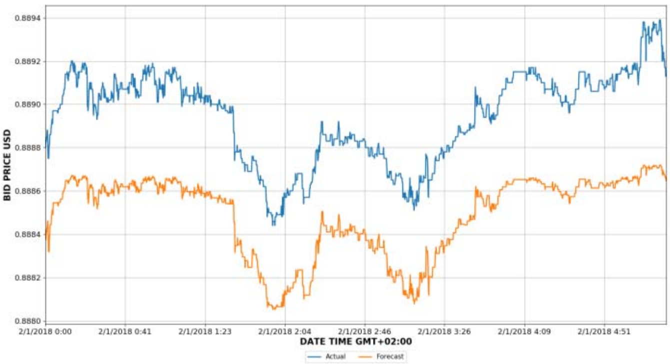

Figure 6. EUR/GBP – Actual vs. Forecast

The run time of the test was approximately 10.33 hours, resulting in a NRMSE = 0.0010.

D. EUR/USD

For the EUR/USD parity test, the multi-layer neural network presented in the previous chapter was trained using the following parameters:

- Complete data set: 19470 points (approximately 54 hours)
- Features used in training: ask, bid.
- Drive data set: 90% of full data set, 17523 points.
- Test data set: 10% of the full data set, 1947 points.
- Number of training epochs: 100.

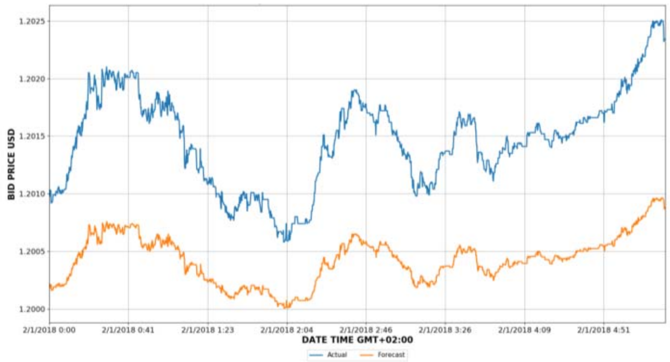

Figure 7. EUR/USD – Actual vs. Forecast

The run time of the test was approximately 10.40 hours, resulting in a NRMSE 0.0027.

VI. Discussion

All charts presented in the previous chapter contain more than 1000 points, so in order to analyze the difference between the actual and forecasted curves, we zoomed in the chart resulted from the experiment conducted on DASH/USD parity.

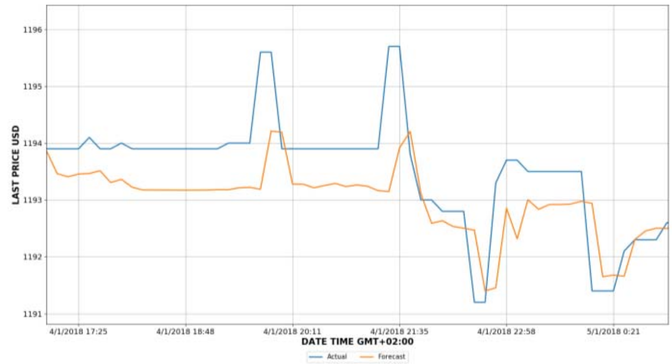

Figure 8. DASH/USD – Actual vs. Forecast zoomed in chart

Analyzing the chart presented in Figure 8, we noticed that the forecasted values, are just a filtered shifted version of the actual values. The forecasted values are delayed by 10 seconds, with corresponds to the sample rate of the inputted dataset.

The proposed LSTM based Forecasting Module is also experiencing the following limitations:

- The LSTM based Forecasting Module is currently limited to forecasting only the price corresponding to the next time step, i.e. only for the next 10 seconds, it cannot be used for multi time-step forecasting.
- The training time required by the LSTM based Forecasting Module is very currently very high. In order to have a minimal error of a deep learning algorithm it is recommended to offer as much training data as possible. It is expected that if the dataset increases, the training timeline will also increase.
- All hyper parameters of the LSTM based Forecasting Module were selected by a manual trial & error method based only on the bibliographic study.

- The forecasted values provided by LSTM based Forecasting Module presented in this paper are not based on any fundamental analysis of the financial markets in order to determine the influence of human, political, geopolitical, or social factors.

To overcome these limitations encountered by the proposed LSTM based Forecasting Module, the following further developments are recommended:

- A forecast for the next 10 seconds does not offer enough time for a trading strategy be profitable, we recommend modifying the module so that it provides multi-time step forecasting for at least one hour.
- To solve the problem of long training time, we recommend moving the LSTM based Forecasting Module into a special learning infrastructure that uses GPU.
- To choose the best hyper parameter, we recommend implementing a genetic algorithm whose chromosome is represented by the total hyper parameters, and its fitness function is the model error and the training time.
- In order to solve the lack of fundamental analysis, we recommend implementing a natural language processing module that should process the latest news related to the current state of the economy.

VII. Conclusions

The presented paper proposed a conceptual architecture of an Intelligent Trading System which is centered around a Financial Forecasting Module which in turn is based on a module built using stacked LSTM architecture. We evaluated the proposed LSTM based Forecasting Module using datasets that contain the evolution of the forex and cryptocurrencies financial markets, observing that even if the error corresponding to the forecasted values is low, the forecast cannot contribute positively to the Intelligent Trading System since it is just a shifted filtered version of the actual price.

References

[1] T.E. Copeland, J.F. Weston, "Financial Theory and Corporate Policy", Addison-Wesley, 1988.

[2] Triennial Central Bank Survey, Bank for International Settlements, 2016. [Online]. Available: https://www.bis.org/publ/rpfx16fx.pdf. [Accesed Feb. 21, 2019].

[3] B. Waters, "The Rise of Retail Foreign Exchange", 2008. [Online]. Available: http://tradingmarkets.com/recent/the_rise_of_retail_foreign_exchange-640269.html. [Accesed Jan. 12, 2019].

[4] Andy Greenberg, "Crypto Currency", 2011. [Online]. Available: https://www.forbes.com/forbes/2011/0509/technology-psilocybin-bitcoins-gavin-andresen-crypto-currency.html#5e5c71ca353e. [Accesed Feb. 10, 2019].

[5] Chohan, Usman W., Cryptocurrencies: A Brief Thematic Review, 2017. SSRN Electronic Journal. Available: https://ssrn.com/abstract=3024330.

[6] Ian Allison, "If banks want benefits of blockchains they must go permissionless",2015. [Online]. Available: https://www.ibtimes.co.uk/nick-szabo-if-banks-want-benefits-blockchains-they-must-go-permissionless-1518874. [Accesed Feb. 10, 2019].

[7] Matteo D'Agnolo, "All you need to know about Bitcoin", 2015. [Online]. Available: https://web.archive.org/web/20151026140555/http://economictimes.indiatimes.com/news/international/business/all-you-need-to-know-about-bitcoin/articleshow/48910867.cms. [Accesed Feb. 11, 2019]

[8] Jon Matonis, "Bitcoin 101 How to get started with the new trend in virtual currencies", 2016. [Online]. Available: https://vdocuments.site/bitcoin-101-5652e66d7a5db.html. [Accesed Feb. 11, 2019]

[9] Ernest P. Chan, "Quantitative Trading How to Build Your Own Algorithmic Trading Business", John Wiley & Sons, 2009.

[10] Jan G De Gooijer, Rob J Hyndman, "25 years of time series forecasting", *International Journal of Forecasting*, vol. 22, no 3, pp. 443–473, 2006.

[11] Qiang Yang, Xindong Wu, "10 Challenging Problems in Data Mining Research", *International Journal of Information Technology and Decision Making*, vol. 5, no. 4, pp, 597-604, 2005.

[12] S. L. Ho, M. Xie, "The Use of ARIMA Models for Reliability Forecasting and Analysis", *Computers & Industrial Engineering*, vol. 35, no. 1-2, pp. 213-216, 1998.

[13] U. Thissen, R. Van Brakel, A. De Weijer, W. Melssen, L. Buydens, "Using Support Vector Machines For Time Series Prediction", *Chemometrics and Intelligent Laboratory Systems*, vol. 69, no. 1-2, pp. 35-49, 2003.

[14] Bin Gui, Xianghe Wei, Qiong Shen, Jingshan Qi, Liqiang Guo, "Financial time series forecasting using support vector machine", *Tenth International Conference on Computational Intelligence and Security*, pp. 39-43, 2014.

[15] Joseph de Brabanter, Bart De Moor, Johan A. K. Suykens, Joos Vandewalle, "Least Squares Support Vector Machines", World Scientific Publishing, 2001.

[16] Haifeng Wang, Dejin Hu, "Comparison of SVM and LS-SVM for Regression", *The Brain & Neural Networks*, pp. 279-283, 2005.

[17] Peter G. Zhang, B. Eddy Patuwo, Michael Y. Hu, "Forecasting with artificial neural networks: The state of the art", *International Journal of Forecasting*, vol. 14, no. 1, pp. 35-62, 1998.

[18] Michael Husk, Peter Stagge, "Recurrent Neural Networks for Time Series Classification", *Neurocomputing*, vol. 50, pp. 223-235, 2003.

[19] Justin Bayer, "Learning Sequence Representations", 2015. [Online]. Available: https://d-nb.info/1082034037/34. [Accesed March 18, 2019]

[20] Pascanu Razvan, Tomas Mikolov, Yoshua Bengio, "On the difficulty of training recurrent neural networks", *30th International Conference on Machine Learning*, 2012. [Online]. Available: https://pdfs.semanticscholar.org/53fa/71c175133dd265f3d64122b9310ef2e7359a.pdf?_ga=2.150450678.1376110693.1564228459-1266668994.1564228459. [Accesed March 20, 2019]

[21] Sepp Hochreiter, Jürgen Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[22] Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Relly Media, 2017.

[23] Alex Graves: "Supervised Sequence Labelling with Recurrent Neural Networks", Springer, 2012.

[24] Xavier Glorot, Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks", *Journal of Machine Learning Research*, vol. 9, pp. 249-256, 2009.

[25] Klaus Greff, Rupesh K. Srivastava, Jan Koutnik, Bas R. Steunebrink, Jurgen Schmidhuber ", LSTM: A Search Space Odyssey", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, 2016.

[26] Paul J. Werbos, "Backpropagation Through Time: What It Does and How to Do It", *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-1560, 1990.

[27] Cem Kadilar, Muammer Sdmsek, Çağdaş Hakan Aladag, "Forecasting The Exchange Rates Series With ANN: The Case of Turkey", *Istanbul University Econometrics and Statistics e-Journal*, vol. 9, pp. 17-29, 2009.

[28] Joarder Kamruzzaman, Ruhul Sarker, “Application of Support Vector Machine to Forex Monitoring”, *IEEJ Transactions on Electronics Information and Systems*, vol. 124, no. 10, pp, 1944-1951, 2004.

[29] Chih-Chaing Lu, Chih-Hung Wu, “Support Vector Machine Combined with GARCH Models for Call Option Price Prediction”, *International Conference on Artificial Intelligence and Computational Intelligence*, pp. 35-40, 2009.

[30] Joarder Kamruzzama, Ruhul A Sarker, “Comparing ANN Based Models with ARIMA for Prediction of Forex Rates”, 2003. [Online]. Available: https://www.researchgate.net/publication/245712720_Comparing_ANN_Based_Models_with_ARIMA_for_Prediction_of_Forex_Rates/link/54e335660cf2d618e1961fdf/download. [Accesed Apr. 4, 2019].

[31] Jiayi Zhang, Wenge Rong, Qiubin Liang, Haonan Sun, Zhang Xiong, “Data Augmentation Based Stock Trend Prediction Using Self-organising Map”, *Neural Information Processing. ICONIP 2017. Lecture Notes in Computer Science, vol 10635*, 2017.

[32] Xu Zhang, Furao Shen, GuoHai Yang, “Time Series Forecasting Using GRU Neural Network with Multi-lag After Decomposition”, *Neural Information Processing. ICONIP 2017. Lecture Notes in Computer Science, vol 10638*, 2017.

[33] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, Tarek Abdelzaher, “DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing”, 2017. [Online]. Available: https://arxiv.org/abs/1611.01942. [Accesed Apr. 5, 2019].

[34] Luca Di Persio, Oleksandr Honchar, “Artificial Neural Networks architectures for stock price prediction: comparisons and applications”, *International Journal of Circuits, Systems and Signal Processing*, vol. 10, 2016.

[35] Wei Bao, Jun Yue, Yulei Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory”, 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0180944. [Accesed Apr. 10, 2019].

[36] Jiahong Li, Hui Bu, Junjie Gavin Wu, “Sentiment-aware stock market prediction: A deep learning method”, *International Conference on Service Systems and Service Management*, pp. 1-6, 2017.

[37] Ishan Verma, Lipika Dey, Hardik Meisheri, “Detecting, Quantifying and Accessing Impact of News Events on Indian Stock Indices”, *International Conference on Web Intelligence*, 2017. [Online]. Available: https://www.researchgate.net/publication/319048522_Detecting_Quantifying_and_Accessing_Impact_of_News_Events_on_Indian_Stock_Indices. [Accesed Apr. 15, 2019]

[38] Murtaza Roondiwala, Harshal Jayeshkumar Patel, S Varma, “Predicting stock prices using LSTM”, *International Journal of Science and Research (IJSR)*, vol. 6, no. 4, 2017.

[39] Armando Bernal, Sam Fok, Rohit Pidaparthi, “Financial Market Time Series Prediction with Recurrent Neural Networks”, 2012. [Online]. Available: http://cs229.stanford.edu/proj2012/BernalFokPidaparthi-FinancialMarketTimeSeriesPredictionwithRecurrentNeural.pdf. [Accesed Apr. 17, 2019]

[40] Kel Butcher, “Forex Made Simple: A Beginner's Guide to Foreign Exchange Success”, Wiley, 2011.

[41] Kyunghyun Cho, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”, 2014. [Online]. Available: https://www.aclweb.org/anthology/D14-1179. [Accesed Apr. 17, 2019]

[42] Oanda Electronic Trading Platform. [Online]. Available: https://www1.oanda.com. [Accesed Nov. 23, 2018]

[43] Oanda REST API Refference. [Online]. Available: https://developer.oanda.com/rest-live/introduction. [Accesed Nov. 23, 2018]

[44] Bitfinex Electronic Tradin Platform. [Online]. Available: https://www.bitfinex.com. [Accesed Nov. 25, 2018]

[45] Bitfinex REST API Refference. [Online]. Available: https://docs.bitfinex.com/docs/rest-general. [Accesed Nov. 25, 2018]

[46] Bid-Ask Spread, Investopedia. [Online]. Available: https://www.investopedia.com/terms/b/bid-askspread.asp. [Accesed Nov. 12, 2018]

[47] Definition of terms, Bitfinex. [Online]. Available: https://support.bitfinex.com/hc/en-us/articles/115003291965-Definition-of-Terms. [Accesed Nov. 25, 2018]