10-22-2020

# Towards High Performance Stock Market Prediction Methods

Warren M. Landis
*Harrisburg University of Science and Technology*, wmlandis@my.harrisburgu.edu

Sangwhan Cha
*Harrisburg University of Science and Technology*, scha@harrisburgu.edu

# Towards High Performance Stock Market Prediction Methods

Warren Landis, Sangwhan Cha
*High Perfomance Computer Lab*
*Harrisburg University*
Harrisburg, United States
WMLandis@my.harrisburgu.edu, scha@harrisburgu.edu

***Abstract*— Stock markets of today, and will continue to in the future, rely on the metrics of timeliness and efficiency to reach optimal profits. A way stock investors have continued to strive for the best of these two factors of the business is through the use of predictive machine learning systems to help aid in their decision making. However, among the many systems currently in use, it could be said that the myriad of data that they are based on may not be sufficient. In an effort to devise an ensemble learning predictive system that will utilize an array of big data sources, we conducted research into the use of long-term short-term recurrent neural networks in stock prediction and planned experiments around the optimization of the machine learning model's timeliness for it to be an effective implementation into our proposed predictive system.**

***Keywords—Stock Market Prediction, Big Data, Machine Learning, Neural Networks, System Assisted Trading, Parallelization***

## I. INTRODUCTION

Stock market prediction can be one of the challenges that financial-related workers need to face because of unique characteristics of the stock market such as the non-existence of patterns and high dynamicity. Predicting the change of stock price should be based on various data sets such as financial data, daily stock market data (open/median/close), company announcements, and news feeds that can be observed from many different sources. Efficient mining of these data sets with machine learning (ML) algorithms have been an active research area during the past decade to develop stock market behavior modeling [8]. Since artificial neural networks (ANN) have been a predominant solution for regression and classification application, it could be applied to stock market prediction based on time series data that is defined as a chronological sequence of observations for a selected variable [12].

Many researchers have used various kinds of ANN techniques such as auto-regressive moving average models, along with two types of back propagation neural networks and multi-layer perception to predict stock trends [8]. The main drawback of a feed forward neural network is that it is not able to handle sequential data because it considers only the current input and cannot memorize previous inputs. Recurrent Neural Networks (RNN) that overcome this drawback have been a predominant solution. RNNs have gained great attention in recent stock prediction research because they are designed to learn sequential or time varying patterns [9]. There is a special kind of RNN, which are Long Short-Term Memory Networks (LSTM), capable of learning long-term dependencies. Gao et al. [8] proposed a model to predict stock closing prices by applying LSTM showing that LSTM provided the best prediction performance of 100/200/400 days with S&P 500 index. As indicated in [10], there have been many pieces of research conducted [11-13,15,17] for forecasting stock prices based on LSTM. The reason for this being that the precision of predictions from traditional methods such as Autoregressive Moving Average (ARIMA), support vector regression (SVR), and random forest are not high enough because of the unique characteristics of stock data.

In this paper, we explain and put forward the research on predictive methods and our proposed approach towards a system that will utilize big data for predicting stock market prices and how it can be optimized further than is typically implemented based on LSTM. Furthermore, we propose our system to utilize parallelization with Ray framework [20] to reduce the latency of training a model.

The rest of the paper will be organized as follows: related works are introduced in Section II. Section III presents background and Section IV explains a proposed system design. Planned experiments is described in Section V and the conclusion is presented in Section IV.

## II. RELATED WORKS

It is very challenging to predict stock price mainly due to the underlying nature of the financial domain, which can be affected by the mix of known parameters such as average/high/low/closing price and unknow events such as election results and rumors [1]. However, there have been various Machine Learning (ML) techniques which have been applied to stock trading based on historic and real time data. Support Vector Machines (SVM) [14], SVR [16], Decision Stumps, and LSTM have been widely used to investigate the law of the stock market recently. As high dimensional input space, sparse document vectors, and regularization parameters are main advantages of SVM, there are many applications of SVM such as face detection, text and hypertext categorization, classification of images, bioinformatics, and stock prediction. Chen et al. [2] proposed the prediction of stock trading signal based on SVM, which has been excellent at handling a pattern classification problem showing the excellent generalization

ability of SVM. In [3], authors proposed a hybrid Cuckoo Search Optimization-Support Vector Machine (CS-SVM) to complement high dimensional input parameters and noisy data claiming that SVM can outperform ANN. Furthermore, SVR, which is another form of SVM with continuous ordered variables, has been applied to stock forecasting with time series stock data. Hou et al. [4] proposed a short-term stock price prediction method based on SVR that is optimized by improved fruit fly algorithm (IFOA), claiming the prediction accuracy was improved. The main objective of using SVR is to make the sum of the distances from the data points to the hyperplane as small as possible for choosing a hyperplane with a small norm [5].

Another mainstream focus of stock prediction research has been stemming from ANN because its ability to perform complex nonlinear mappings and tolerances to noise in time series stock data has been well established [6,7]. Many researchers have used various kinds of ANN techniques such as auto-regressive moving average models, along with two types of back propagation neural networks and multi-layer perception to predict stock trends [8]. The main drawback of a feed forward neural network is that it is not able to handle sequential data because it considers only the current input and cannot memorize previous inputs. Recurrent Neural Network (RNN) that overcomes this drawback has been predominant solution and gained lots of attention in recent stock prediction research because it was designed to learn sequential or time varying patterns [9]. There is a special kind of RNN, which is Long Short-Term Memory Networks (LSTM), capable of learning long-term dependencies. Gao et al. [8] proposed a model to predict stock closing price by applying LSTM showing that LSTM provided the best prediction performance of 100/200/400 days with S&P 500 index. As indicated in [10], there have been many researches [11-13,15,17] for forecasting stock prices based on LSTM as the precision predictions of traditional methods including Autoregressive Moving Average (ARIMA), SVR and random forest are not high enough because of unique characteristics of stock data.

Before choosing the prediction methods we would utilize, we analyzed different stock market prediction systems. We conducted in depth research into the related works surrounding the two prediction methods we were focusing on, SVM and RNN. While SVR is great when used with historical time series data, it is still not very applicable for prediction and simply more suited for assisting decisions given its pervasiveness for error when used to predict past historical patterns [16]. Alternatively, SVM proves to have a relatively good record of predictive accuracy when being used in stock market predicting. SVM, unlike SVR, has a greater ability to generalize, thus allowing it to be better suited for predicting rather than simple historical pattern analysis [14].

RNN's versatility for prediction and the accuracy it returns despite the type of input shows great reason as to why it should be chosen as the method for us to analyze and develop some way of optimally implementing it into an overarching stock predicting system. Referring to this stock predicting system and RNN, this system is best described as an ensemble machine learning model given its purposed different method modules that are cooperatively working towards a singular goal of accurately predicting future stock market prices. Besides traditional time series stock data, this can be done through unconventional data like social networking sentiment analysis [18]. The modules within an ensemble machine learning predictive system could include additional types of data like news headline analysis or stock trend comparisons [19]. Ensemble machine learning is the eventual goal of our proposed predictive system, however, first, we must analyze the best predictive method, LSTM, and furthermore how it can be best applied to our use cases of them.

## III. Background

In these past findings, we had concluded that amongst the present predictive methods, LSTM RNNs were the most optimal and accessible choice for two reasons. The first of these two reasons were that generally, LSTM RNNs outperformed the other predictive methods we looked at in almost every way. The second reason would be that, given our use case of stock prediction produced by an overarching ensemble predictive system, it would be more suited to use LSTM RNNs as they would be very versatile and workable models to be adjusted into the different facets of what we have planned for future projects. Whatever the application may be within our use case, we found that LSTM RNN is suitable for a majority of cases, until we meet otherwise in later research and development of the aforementioned predictive system we are hypothesizing. The primary issue with utilizing LSTM RNN, let alone utilizing it in a multitude of ways, is that the time it can take for the model to iterate through its epochs when training can be exceedingly lengthy. Because of this, the best improvement we could make unto our implementation of LSTM RNN would be finding a way to speed up the training process of the models. Doing so would drastically increase the overall efficiency of our hypothesized system. As without improvements to the training time, the more the system has LSTM RNN utilizing features added onto it, the slower and less efficient it will become. After initial research into the potential ways in which this can be achieved, we have come to the plan that our attempt to solve this issue will be to utilize parallelization. What this means is that we will attempt to parallelize the process of training the models, which in turn should hopefully cut down on the time consumed by the training process. Additionally, along with the use of parallelization, we also plan to host the models within a cloud environment. This will allow us to have both resource availability and flexibility, which will allow for multiple instances of LSTM RNN to occur at the same time and for these instances' resources to be allocated appropriately. With both of these solutions, the first is more experimental, while the second is a given. This is because no matter what, a healthy availability of computational resources will be unquestionably needed when running many predictive models at the scale we propose. Otherwise, no matter how successful we are at parallelizing the training process, the use of the models would still drag the time down.

## IV. Proposed System Design

The current implementation of LSTM RNN is present in a Python test module we have created. These test modules are designed, at this stage, to have their primary function as predicting stock prices using time series data. For future use of LSTM RNN, the test modules are designed to be relatively modular to the point where they need to be altered to handle the new input and be tooled with the proper handling to be used for

the other applications of LSTM RNN later on. Disregarding this, as they stand now, they are meant to take time series data in the form of CSV text files, one for the training data and one for the test data. The output of the test module are several graphs that display the historical price trend of the stock, rolling mean analyses of the price trend, and finally a graph which displays a comparison graph between the actual stock price and the predicted stock price. In addition to these graphs, for the final graph, an accuracy value is provided to give a non-visual and concrete basis for its predicting success. Some additional numerical information is provided as well, which includes time elapse values for monitoring the efficiency of the test module as a whole. Currently, there are several libraries that our test modules utilize to perform their primary function of stock prediction. These libraries and their uses within our test modules are the following:

- Pandas is used to structure the input CSV data into data frames for the creation of training sets to be used in the training of the LSTM RNN models
- Scikit-learn supplied the MinMaxScaler function which is used for data scaling
- NumPy is used to format the created time series datasets into useable arrays and matrices
- Datetime is used to organize and format the input CSV data based on the historical data information of the entries in the datasets
- Time is used simply as a means to record time efficiency of the test module
- Keras is used for the functions and methods for the creation and execution of LSTM RNN
- Matplotlib is used for formatting data into a plottable form then plotting the several output graphs that the test modules provide

These are the libraries that encompass and help operate the pre-parallelization test module. For the parallelization of the test modules, we had looked at several solutions for the best way to go about achieving such a thing. We decided that the way we would attempt this would be to use a currently in development Python library called Ray. The Ray library would allow for the running of the test module in a single-machine distributed way that would not affect the pre-existing code of the test module as well. The Ray library brings with it a distributed system framework that accommodates for both actor-based and task-parallel programming abstractions. The actor-based portion of the framework is the side we would like to primarily focus on. Actors within this feature set of Ray allow for the framework to support and enable the operation of stateful computations, like for model training in our case. Ray's cluster-computing framework is general-purpose, meaning that despite our use, it can support a wide array of other uses as well including graphing and reinforcement learning. In our use, actors within the Ray framework are stateful computations that, while executed serially, we will be attempting to utilize them in parallelizing the functions that allow for the training of our models within the test modules. Alternatively, should this not demonstrate an improvement in training latency, a task-parallel route can be taken. This would encompass using the designation of functions as remote tasks in attempt to run them in parallel. However, this has the potential to cause issue with accuracy and general operative health of the training as epoch training is traditional sequential. With doing this, our only concern would be any effect done onto the accuracy of the predictive models by executing the epochs in this fashion. The creation and execution of our predictive model happens almost exclusively within the Keras library. Because of this, the actual implementation of the Ray library could not take place inside the test module and rather needs to take place inside the Keras library. With the implementation of the Ray library inside of Keras, specifically as it resolves to the functions that operate the epochs of the model training, parallelization is possible. The aforementioned potential impact on accuracy will be something we address and look after when running the test modules and can be compared with the pre-parallelization modules to see the impact if any exists. For the other solution of improving the operation of the LSTM RNN models through being hosted on a cloud environment, we have a means for this as well. This would be achieved through the test modules simply being hosted on Linux virtual machines on Harrisburg University's cloud environment ran out of their High-Performance Computing Lab (HPCL). In this environment, we could easily manage the resources needed to properly test whether or not the parallelization of the test modules was successful or not in an overall efficiency test. This environment could also be used to eventually be the staging ground for our later hypothesized ensemble predictive system.

To further explain this hypothesized ensemble predictive system for more context, a general system architecture overview can be seen illustrated in Fig. 1. This proposed system would encompass multiple input data types varying from time series stock data, financial news data, and contractual data. All of these data types would be processed into useable information for the LSTM RNN modules. The processing method for each data type will inherently vary given the data type, such as simple data formatting for time series data or natural language processing (NLP) with word embeddings for financial news as examples. With the data formatted, it can then be used with the LSTM RNN modules to attempt to predict the outcome of whatever stock related data we are focusing on. Each prediction will then be utilized within an ensemble learning system, where each prediction will be given a varied weighting. It is important to note as well that within this speculated architecture currently, the LSTM RNN modules will be dispersed across the Ray cluster framework for the hopeful increased efficiency that it may provide. This weighting will be determined later, however the general idea is that some sources of financial data should inherently have more impactful correlations with the actual price of the stock and therefore will have heavier weighting when calculating a final prediction. With weightings assigned, these outcomes from each module after being trained and processed on the cluster computing framework will be calculated together within the ensemble learning system to derive a final output. At this time, what the final output could be is purely speculative. However, the hypothesized idea for what it could be is either a more weighted prediction of what the price could be, similar to the current prediction returned by the time series stock data

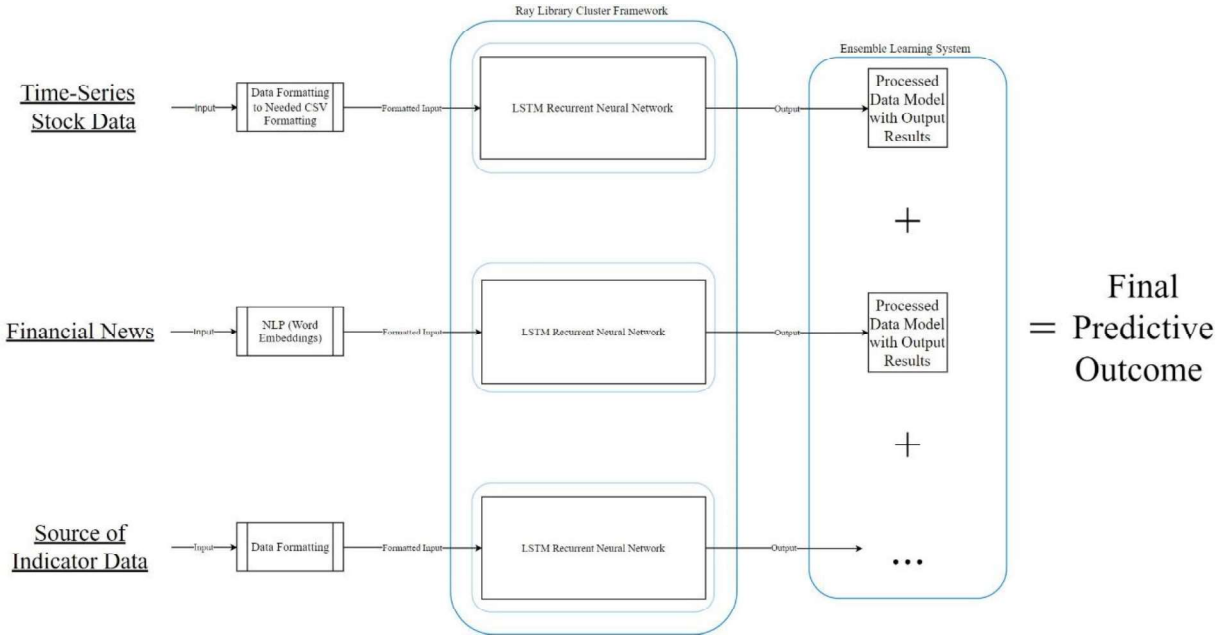


Fig. 2. Hypothesized predictive system utilizing ensemble learning architecture

LSTM RNN module. Alternatively, the final output simply could be a return of whether it is predicted that the price of the stock will go up or down in comparison to the day prior of the targeted date of prediction. This hypothesized architecture at this time is quite high-level in abstraction as well as speculative given a majority of its specifications have yet to been thoroughly explored. Works, such as this present one, will be optimally working towards fully realizing these specifications.

## V. Experiments

The experiments for the finished and potentially parallelized test modules are planned out to test a variety of things. For the experiments, which are graphically illustrated in Fig. 2 the test modules will be working with formatted time series stock data of twenty companies. The companies chosen will range from well-known and heavily traded companies like Apple and Google to less known companies and not as frequently speculated companies. This is in attempt to avoid any possible biases that could exist with only testing with data from popular companies, and also to train the models against a wide range of varying stock market scenarios. The training data will begin at the start of 2015 and stretch to the end of 2019, while the data that the models will be tested against will consist of the first month of 2020. We think that this four-year span of time is sufficient to encompass enough of the stock's price history for accurate prediction. For each company, the test modules will be trained several times, both the original test modules along with the potentially parallelized test modules. The accuracy and efficiency of both groups of test modules will be recorded. Then, the data will be compared between the two groups to see if the efficiency has been improved by the parallelization and if the accuracy has been impacted in any way by the parallelization as well.

## VI. Conclusion

Currently, the test modules are not completed and are still in progress of being finished. When the experiments have been completed, the success or failure of the parallelization attempt will likely decide our next course of action towards developing our mentioned hypothesized predictive system. If the results of the experiments are deemed a failure, we will have to go back

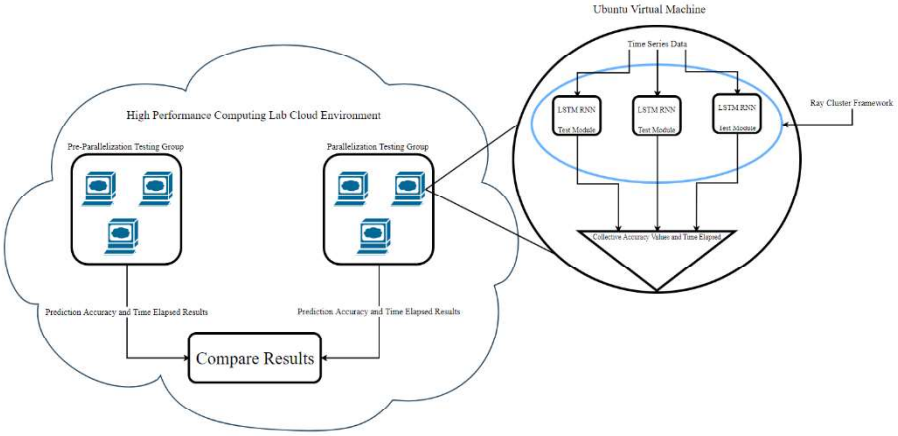


Fig. 2. Illustrated example of the proposed experiment's design

and devise another method of parallelization to remedy the time cost of training the LSTM RNN models. Either this, or we simply must account for the time-consuming process of training the models, however this seems unlikely as the time costs will become increasingly expensive the more LSTM RNN models we implement. If the results of the experiments are deemed successful, we will be able to use this method of parallelization moving forward in our use of LSTM RNN as we work towards developing an overarching ensemble predictive system for predicting stock prices.

## REFERENCES


[1] V. H. Shah, "Machine learning techniques for stock prediction", Online available : www.vatsals.com, accessed in May, 2020

[2] Chen, X., & He, Z. J., Prediction of Stock Trading Signal Based on Support Vector Machine. Proceedings - 8th International Conference on Intelligent Computation Technology and Automation, ICICTA 2015, 651–654.

[3] Devi, K. N., Bhaskaran, V. M., & Kumar, G. P., Cuckoo optimized SVM for stock market prediction. ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems, 1–5.

[4] Hou, X., Zhu, S., Xia, L., & Wu, G., Stock price prediction based on Grey Relational Analysis and support vector regression. Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018, (61370154), 2509–2513.

[5] Xia, Y., Liu, Y., & Chen, Z. Support Vector Regression for prediction of stock trend. Proceedings of 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2013, 2, 123–126.

[6] Zheng, A., & Jin, J. Using AI to Make Predictions on Stock Market. Online available : http://cs229.stanford.edu/proj2017/final-reports/5212256.pdfhttp://cs229.stanford.edu/proj2017/final-reports/5212256.pdf, accessed in May, 2020

[7] Pinheiro, S., & Dras, M., Stock Market Prediction with Deep Learning: A Character-based Neural Language Model for Event-based Trading. Proceedings of Australasian Language Technology Association Workshop, 2017

[8] Gao, T., Chai, Y., & Liu, Y., "Applying long short term memory neural networks for predicting stock closing price", in 8th IEEE international conference on software engineering and service science (ICSESS), 2017, pp. 575-578

[9] Jain L. C. and Medsker L. R., "Recurrent neural networks: Design and Applications", in the joint conference on neural network, 1999, pp. 1537-1541

[10] J. Du, Q. Liu, K. Chen, and J. Wang, "Forecasting stock prices in two ways based on LSTM neural network", in IEEE 3rd information technology, Networking, Electronic and Automation control conference (ITNEC 2019), 2019, pp. 1083-1086

[11] S. E. Gao, B. S. Lin, and C. M. Wang, "Share price trend prediction using CRNN with LSTM structure", in International symposium on Computer, Consumer and Control (IS3C), 2018, pp. 10-14

[12] S. Selvin et al. "Stock price prediction using LSTM, RNN and CNN-Sliding window model", in 3rd international conference on Circuits, Control ,Communication and Computing (I4C), 2018, pp. 1643-1647

[13] Y. F. Lin, Y. L. Ueng, W. H. Chung, and T. M. Huang, "Stock price range forecast via a recurrent neural network based on the zero-crossing rate approach", in IEEE conference on computational intelligence for financial engineering & economics, 2019

[14] X. Chen and Z. He, "Prediction of stock trading signal based on support vector machine", in IEEE international conference on intelligent computation technology and automation (ICICTA), 2015, pp. 651-651

[15] A.J.P. Samarawickram and T.G.I. Fernando, "A recurrent neural network approach in predicting daily stock prices", in IEEE international conference on industrial and information systems (ICIIS), 2017, pp.1-6

[16] Y. Xia, Y. Liu, and Z. Chen, "Support Vector Regression for prediction of stock trend", in IEEE international conference on information management, innovation management and industrial engineering (ICIII), 2013, pp. 123-126

[17] R. Achkar, F. Elias-Sleiman, H.Ezzidine, N. Haidar, "Comparison of BPA-MLP and LSTM-RNN for Stocks Prediction", in IEEE international symposium on computational and business intelligence (ISCBI), 2018, pp. 48-51

[18] W. Chen, Y. Zhang, C. K. Yeo, C. T. Lau, B. S. Lee, "Stock market prediction using neural network through news on online social networks", in IEEE international smart sities conference (ISC2), 2017

[19] M. S. Hegde, G. Krishna, R. Srinath, "An ensemble stock predictor and recommender system", in IEEE international conference on advances in computing, communications and informatics (ICACCI), 2018, pp. 1981-1985