Stock Market Prediction using Artificial Intelligence

Rafael Isaac Cano Guitton
School of Computer Science
Universidad Católica San Pablo
Arequipa, Perú
Email: rafael.cano@ucsp.edu.pe

***Abstract*—**

Keywords: Stock Market, Machine Learning, Deep Learning , Predictions

I. Introduction

The stock market is a collection of markets where stocks of a firm are traded (bought and sold) [1]. Its behaviour is considered chaotic [2] and it has always been ambiguous for investors because of several influential factors. This unreliable behaviour make investors live the potential to lose big sums of money, or to treat the stock market as gambling. There's a correlation between investment psychology and market behaviour. With a predictive model, we can give a solution for impulsive market selling and buying. Assuring a percentage of reliability in trading this impulsive action can be reduced. While there's many automated ways to assist investors' decisions in a timely manner [3], it's not enough to adress this first issue.

The main reason behind prediction is buying stocks that are likely to increase in price, and then selling stocks that are probably going to fall [3]. Stock prices react to events related to business performance or overseas markets. Investors judge on the basis of technical analysis, such as company's charts [4].

Now it is difficult to predict market trends and many AI approaches have been investigated to predict them automatically. For example, investment simulation analysis with artificial markets [4].

We'll focus on Machine Learning and Deep Learning:

- Machine Learning: It's a branch of AI that parses data, handles this data more efficiently and learns from that data. The main advantage is that, once the learning process is in a mature state, it can make informed decisions [5]. We use it in stock market prediction by learning patterns among big amounts of information. They can tackle the predicting task of price fluctuations to improve trading strategies [3].
- Deep Learning: It's a subfield of Machine Learning, so technically we can say that Deep Learning is Machine Learning. It's considered an evolution of Machine Learning, it structures algorithms in layers to create an Artificial Neural Network, this enables accurate decisions without help from humans. The stock market's instability and nonlinearity cause problems for data analysts to develop a predictive model [3]. And thus we can use Neural Networks for a predictive model.

Artificial Neural Networks are specially useful since it's main usage is to recognize patterns. They're essentially simplified models of brains. In a brain, you have neurons, which are either activate or inactive, and synapses, which connect the neurons together. The neurons are represented as simple booleans, and the synapses are represented by generally small numbers between negative one and one. The "weigth" of all the synapses connected to a neuron determine it's state [1].

Training a Neural Network generally takes a lot of time, but using one to make predictions is very fast.

For our datasets we can use news articles, financial reports and posts from microblogs made by analysts [6]. Using Convolutional Neural Networks(CNN) and Recurrent Neural Networks(RNN) we can catch semantics from text and context information. We can also use data from individual stocks, using information like stock symbol, stock series, stock date, previous closing, high, low, last, closing and average price of each stock. [1].

The ample research literature, combined with the vast underlying models, tasks and training methods make it very hard to identify the most appropriate approach or the most effective [7].

And there's where our challenge resides. On this survey we aim to compare these techniques. We'll compare prediction values for certain stocks with real stock market behavior. Also we'll compare score based predictions for some approaches. The purpose is to analyze current state of the art techniques

for stock market accuracy [2].

II. Related work

ANN have been used in past decade in stock market prediction [1]. ANN and HMM were proposed with the purpose to transform daily stock values to independent group of prices as inputs to HMM [3]. A common factor among current proposed models is that an improvement to conventional neural networks is attempted. In Amir's study, nine ML methods (Decision Tree, Random Forest, Adaboost, XGBoost, SVC, NaïveBayes, KNN, Logistic Regression and ANN) and two DL methods (RNN and LSTM) were researched [3] , putting special focus on performance. They calculated indicators by stock trading values, using them as continuous data, and then converting indicators to binary data before using it.

In Amrita's study they used National Stock Exchange of India (NSE) and New York Stock Exchange (NYSE). Extracting day-wise closing price of stocks within different criteria, and then normaliznig data before feeding them to the networks to train. They found similarities in patterns within those markets.

There's also the textual information approach, in Vargas' proposal [6] they used 106.494 news articles from Reuter's website, aiming to the topic of financial news. They found that using titles is more useful than the entire article for forecasting purposes, so their proposed model only uses news titles. Now the proposed model uses only news from the day before the forecasting day, comparing them to models that use news from past day, week and month, Vargas' model outperforms said models. On textual information approaches, Akita's proposal [4] also uses financial news, but they also use numerical information. They use them to predict 10 companies' closing stock prices by regression. Also learning correlation between companies. This since news from a company can have an effect in several companies' stock pricing within the same industry.

III. Predictive Models

Taking in account research mentioned in related work, algorithms used for stock market prediction were clasified. This so we can have an overview of which techniques were used and how these were applied. If there was preprocessing of data so we can know if that can have an impact.

A classification based on dataset type was also made, this since several techniques were used in both cases but having different results.

Therefore we'll have a look at models used in stock market prediction by:

- Machine Learning Algorithms.
- Deep Learning Algorithms.

Also describing preprocessing and dataset type.

TABLE I
Models by Technique

Technique	Models	Preprocessing	Dataset type
Deep Learning	CNN [1] [6]	Data Normalization	Textual & Numerical
	RNN [1] [2] [6]	Data Normalization & Converting continuous data (indicators)	
	(2D)²PCA + DNN [2]	-	Numerical
	(2D)²PCA + RBFNN [2]	-	
	DNN [3]	-	
	MLP [1]	Data Normalization	
	LSTM [1] [4]	Data Normalization & Converting continuous data (indicators)	
	ANN [3] [8]	-	
Machine Learning	KNN [3]	Converting continuous data (indicators)	
	Adaboost [3]		
	XGBoost [3]		
	SVC [3]		
	Naïve Bayes [3] [9]		

A. Machine Learning Algorithms

In Nabipour's approach [3], in order to use machine learning models, their steps were: normalizing features for their continous data, randomly splitting the main dataset into train data and test data, fitting the models and evaluating them by validation data to prevent overfitting, and using metrics for final evaluation with test data. A brief listing of Machine Learning Algorithms that were reviewed:

1) K-Nearest Neighbors: A method non-parametric for classification and regression, it's supervised and based on instances. It doesn't learn from a model explicitly, it memorizes the instances of training that are used as a "knowledge base" for prediction phase.

Now to apply KNN we have two properties suggested, lazy learning and non-parametric algorithm, this because there isn't any assymption for underlying data distribution. On Nabipour's approach [3], they used 100 Neighbors and K-dimensional tree algorithm for their KNN prediction approach.

2) Adaptative Boosting: It's a boosting technique that is used as an Ensemble Method. It's goal is to train weak learners sequentially for adjusting their previous predictions. This model is a meta-prredictor which starts by fitting a model on the basic dataset before fitting additional copies of it on the same dataset. During the process of traingin, samples' weights are modified based on the current forecasting error; therefore, the consequent model focuses on tough items.

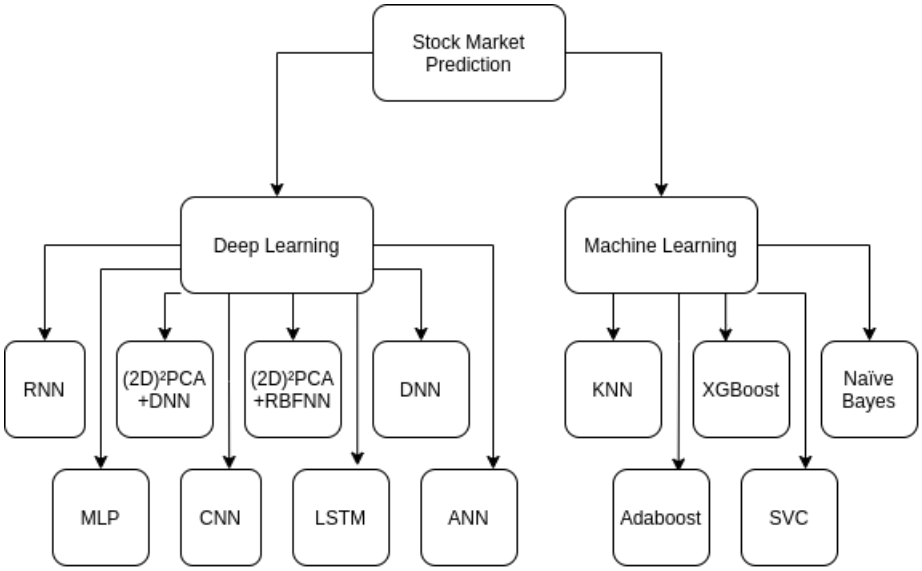

Fig. 1. Paper classification

On Nabipour's approach [3], the parameters for their Adaboost were: a Max Depth of, for their estimator they used a decision tree, having from 50 to 500 trees progressing in a rate of 50, and a learning rate of 0.1.

3) eXtreme Gradient Boosting: It provides an efficient and effective implementation of the gradient boosting algorithm, designed to be computionally efficient and highly effective. Similarly to Adaboost, it employs the rules of Boosting for weak learners but itt has better performance and speed compared to other tree-based models. It has the advantages of regularization for preventing overfitting, in-built cross-validation capabilities, proficient handling of missing data, catch awareness, paralelized tree building and tree pruning.

On Nabipour's approach [3], the parameters for their XGBoost were: a Max Depth of 10, having from 50 to 500 trees progressing in a rate of 50, and Logistic Regression for Binary classification as their objective.

4) Support Vector Clasifier: Supervised learning model, it's objective is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. It looks to maximize the margin between data points and the hyperplane.

For SVC, on Nabipour's approach [3], they used Linear, Poly(degree=3),RBF, Sigmoid for Kernel, a C of 1.0 and their Gamma value was 1/((num)x(variance)) f:features.

5) Naïve Bayes: It's a probabilistic method that is based in Bayes' theorem, called Naive since given some additional simplifications that determine the independance hipotesis of predict variables.

For Naïve Bayes, on Nabipour's approach [3], they used Gaussian Algorithm, for their Gamma value they used 1/((num)x(variance)) f:features, and a C value of 1.0.

There were several classification metrics across papers to evaluate performance of models, among of which we can metion F1-Score, accuracy, receiver operating characteristics [3], Mape (Mean Absolute Percentage Error) [1], effectivenes of Paragraph Vector, effectivenes of LSTM [4], index prediction [6], forecast values, Root Mean Square Error (RMSE), Hit Rate (HR) and Total Return (TR) [2].

According to Deepak [8], Naïve Bayes by itself can and is used for automatic stock purchase model. In the real world it is used to develop web portals for automatic trading.

Each of the machine learning models have their limitations when it comes to solve the stock market prediction problem. For training machine learning models in Nabipour's approach [3], they normalized features, randomly split the main dataset into train data and test data,fitting the models and evaluating them by validation data. When an extra layer to convert continuous data to binary one based on the nature and property of the features there was a clear improvement. So continuous data has terrible performance on every model except some Deep Learning ones, so they're unviable with machine learning. Binary data on the other hand holds the aforementioned improvement to the level of up to 83 percent. This since the layer that converts to it is able to convert non-stationary values to trend deterministic ones.

Now despite it's clear performance disadvantage towards deep learning models, machine learning models have a better runtime performance. This due to the fact that less computational power is required for prediction. This combined with binary data on Nabipour's approach [3], gets no less than 0.83 F1 score which would make it a viable option. This being metioned, some are already used as mentioned on Deepak's paper [8], you can make algorithmic trading bots but it's limited to favourable outcome and will not provide stock analysis.

B. Deep Learning Algorithms

On Nabipour's approach [3], they used three dimensional values (samples, time-steps, features), so they used a function to reshape the input values. Also, weight regularization and dropout layer are employed to prevent overfitting. A brief listing of Deep Learning Algorithms that were reviewed:

1) Convolutional Neural Network: It's an Artificial Neural Network with supervised learning that processes layers imitating the visual cortex of the human eye to identify several characteristics on entries that definetly make it able to identify and see objects, these based in patterns.

On Hiransha's approach [1], their CNN almost captured the pattern between 1500 and 2300 days since it accounts only the data in a particular window. It another stock it actually failed to capture change in system between 1400 and 1800 days.

2) Recurrent Neural Network: A class of neural networks that allow previous outputs to be used as inputs while having hidden states. They can process inputs of any length and take them from two sourcesm one is from the present and the other from the past. Information from these two sources are used to decide how they react to the new set of data.

On Hiransha's approach [1] their RNN failed to identify the seasonal pattern which can be considered as change in behaviour of system. In another stock, RNN was almost successful in identifying the pattern.

With Nabipour's approach [3] since RNN has a recurrent nature, the technical indicators of one or morre days (up to 30 days) are considered and rearranged as input data to be fed into the model. Their parameters were 500 Hidden Layer Neuron Count, 1-2-5-10-20-30 for number of training days, both Tanh and softmax as activation function, learning rate of 0.00005, B_1 of 0.9, B_2 of 0.999 for optimizer, Max EPOCHS OF 10000 and their training stop condition was: Early stopping: monitoring parameter = validation data accuracy patience = 100 epochs.

3) 2-Directional 2-Dimensional Principal Component Analysis + Radial Basis Function NeuralNetwork: The main idea behind $(2D)^2$PCA is that it's based on 2D matrices as opposed to the standar PCA, it has higher accuracy. It is used for dimensionality reduction in the researched proposals. It's output is fed to the RNN.

Since the main idea behind this approach is to use dimension reduction to the dataset, it projects our original raw data matrix into a projection matrix and there is loss of information but the obvious advantage is that the processing time is lower and the convergence speed of the model increases many fold.

With Singh's approach [2], it was found that compared to RNN (that was performing poorly) there is a 15.6% improve.

4) 2-Directional 2-Dimensional Principal Component Analysis + Deep Neural Network: Same as last but it's output is instead fed to Dimension Neural Network, which is the simplest neural netowrk, it has some level of complexity, usually two layers. Now, the dimensional reduction provided by $(2D)^2$)PCA is recommended for large datasets, since the aformentioned information loss wouldn't cause variation in the output.

Compared to RBFN with $(2D)^2$)PCA on Sing's approach [2], the proposed method performed better than the existing method. It had an improved accuracy of 4.8% for hit rate with a window size of 20. So compared to RNN and (2D)²PCA+RBFNN, it's proved to be the better predictor for trend prediction of stock market.

5) Multilayer Perceptron: Also known as feed forward neural network, a simple example of a neural network. Each input neurons are linked to the succeeding hidden layer neuron through a weighted matrix.

On Hiransha's approach [1], MLP had a mixed performance, having identified the pattern in the beginning for AXIS BANK stock in the 1400 to 1700 days period. But was overall the most successful model since it captured the seasonal pattern for HCLTECH between 1600 and 1900 days.

6) Long Short-Term Memory: Is a type of RNN capable of leaning order dependence in sequence prediction problems. A behaviour specially usefull to process our datasets.

On Akita's approach [4], they trained the LSTM for 50 epochs with Adam, with one layer and were unrolled 20 steps, they also applied 50 percent dropout on the non recurrent connections. Now comparing it to an MLP it achieved higher profits in all industries, also achieving higher profits in 4 out of 5 industries comparing it to SVR. In their comparison, LSTM significantly outperformed simple-RNN.

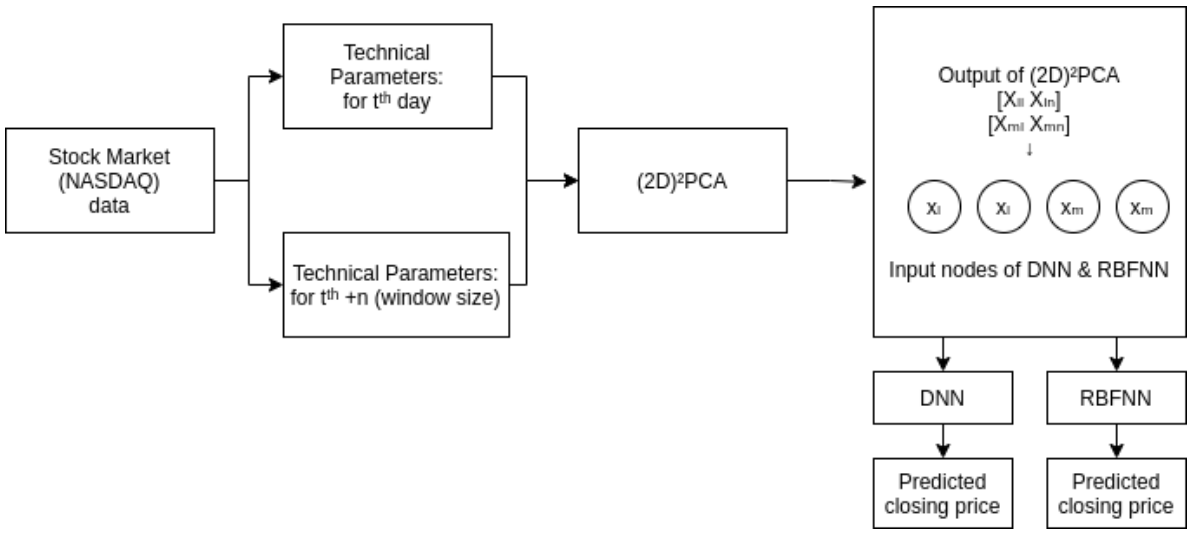

Fig. 2. 2-Directional 2-Dimensional Principal Component Analysis

On Nabipour's approach [3] since LSTM has a recurrent nature, the technical indicators of one or morre days (up to 30 days) are considered and rearranged as input data to be fed into the model. Their parameters were 500 Hidden Layer Neuron Count, 1-2-5-10-20-30 for number of training days, both Tanh and softmax as activation function, learning rate of 0.00005, B_1 of 0.9, B_2 of 0.999 for optimizer, Max EPOCHS OF 10000 and their training stop condition was: Early stopping: monitoring parameter = validation data accuracy patience = 100 epochs. On Hiransha's approach [1] similarly to their RNN, their LSTM failed to identify the seasonal pattern and also failed to capture change in system between 1400 and 1800 days.

7) Artificial Neural Network: Usually single or multilayer nets whuch fully are fully connected together. By the rise in the number of hidden layers, it is able to form the network deeper.

On Nabipour's approach [3] the parameters they used for their ANN were: Hidden Layer Neuron Count of 20,50,100,200,500; ReLU, Sigmoid, Tanh as activation functions; B_1 of 0.9, B_2 of 0.999 for optimizer; and their training stop condition was: Early stopping: monitoring parameter = validation data accuracy patience = 100 epochs.

Since Deep Learning models are more robust on predicting tasks, their performance is always better than Machine Learning models. On Akita's approach [4], they use Paragraph Vector to obtain the distributed representation by mapping variable length pieces of text to a fixed-length vector. It's classified in two categories, Distributed Memory Model of Paragraph Vectors (PV-DM) and Dis-tributed Bag of Words version of Paragraph Vector (PV-DBOW). For their expermients they used a combination of Numerical and Textual information, but they also compared it with only numerical information. It was able to get more profits in the four out of five industries and the total was also higher by 490 points. So it was effective to employ distributed representations by using their aformentioned Paragraph Vector. They simulated real-world stock trading effectively doing Market Simulation. LSTM and RNN were tested on this artificial market and despite both being capable of considering time series data, LSTM significantly outperformed Simple-RNN, this since LSTM has nondeterministic transactions. This meaning LSTM was able to capture the fluctuating time series changes well. When using binary data on Nabipour's approach [3], although there were only 2 deep learning models used (RNN and LSTM), they showed a clear superiority for the predictive task, this time having RNN be superior to LSTM but with a smaller margin. In Textual Information exclusively for Manuel's approach [6], they made heavy comparisons between CNN and RNN based models. Using a classification of inputs from the dataset being word embedding input, sentence embedding input, bag of word, structure events tuple input, sum of each word in a document and event embedding. Their results show that sentence embedding is better than word embedding, and that the RCNN proposed approach layering RNN and CNN has better structure than CNN for the index prediction task. This proposed model outperforms all baseline models with the expection of EB-CNN, this is likely ue to event embedding that is a more powerful method to model the concent in news articles. This also shows that the architecture used for the prediction model is also an important factor.

IV. Conclusion and future work

Conclusion: This paper analyzed and compared various prediction models across several research papers. With the recent successes of Deep Learning in Naturale Language Processing and Image Recognition task, it's pretty tempting

to apply AI to Stock Market Prediction. Researched methods showed that not only is it possible to predict certain trends but that there exist some models that are currently used in real life for assissted trading and automatic trading.
Nevertheless it was also proven that it hasn't, and probably won't reach the point where it can accurately predict the future of (any) market consistently There's many ways to implement a model, going from using numerical, textual or both as datasets. These models can be trained to create it's own trading strategy.

As mentioned, input data plays an important role in prediction, and with use of Numerical data we get high performance and efficiency on Machine Learning models. So performance also plays a core role in Stock Market Prediction model at least in forecasting for day-trading. We could define and implement a model. Resources, strategy and market will also have an impact on how effective models can be. How this data is treated is also important since converting continuous data to binary data also proved effective in performance gains.

To sum it all up, there's promising methods for this task and some of the approaches could be applied to real life trading strategies, some on the analysis side and some could just perform their own operations with their strategies. Both Machine Learning and Deep Learning models are effective, it will just depend on how you expect to use the model.

Future work: As research continues and new algorithms arise like Deep Belief Network, Regularization, Autoencoders and Advanced Optimization, a comparison needs to be made. Regarding dimension reduction, there's not enough work testing this technique and it would be interesting since it was shown that it can improve performance. Specially since Stock Market Prediction datasets tend to be large.

ACKNOWLEDGMENT

Not required for this presentation.

REFERENCES

[1] H. M, G. E.A., V. K. Menon, and S. K.P., "NSE stock market prediction using deep-learning models," *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.05.050

[2] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18 569–18 584, Dec. 2016. [Online]. Available: https://doi.org/10.1007/s11042-016-4159-7

[3] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150 199–150 212, 2020.

[4] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, Jun. 2016. [Online]. Available: https://doi.org/10.1109/icis.2016.7550882

[5] A. Dey, "Machine learning algorithms: a review," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1174–1179, 2016.

[6] M. R. Vargas, B. S. L. P. de Lima, and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*. IEEE, Jun. 2017. [Online]. Available: https://doi.org/10.1109/civemsa.2017.7995302

[7] M. Raghu and E. Schmidt, "A survey of deep learning for scientific discovery," *arXiv preprint arXiv:2003.11755*, 2020.

[8] R. S. Deepak, S. I. Uday, and D. Malathi, "Machine learning approach in stock market prediction," *International Journal of Pure and Applied Mathematics*, vol. 115, no. 8, pp. 71–77, 2017.

[9] V. K. S. Reddy, "Stock market prediction using machine learning," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 10, pp. 1033–1035, 2018.