Predicting Stock Price Movement after Disclosure of Corporate Annual Reports: A Case Study of 2021 China CSI 300 Stocks

Fengyu Han and Yue Wang
School of Information, Central University of Finance and Economics, Shahe Higher Education Park, Changping District, Beijing, China.
Email: hfy1999@qq.com, yuelwang@163.com

Abstract. In the current stock market, computer science and technology are more and more widely used to analyse stocks. Not same as most related machine learning stock price prediction work, this work study the predicting the tendency of the stock price on the second day right after the disclosure of the companies' annual reports. We use a variety of different models, including decision tree, logistic regression, random forest, neural network, prototypical networks. We use two sets of financial indicators (key and expanded) to conduct experiments, these financial indicators are obtained from the EastMoney website disclosed by companies, and finally we find that these models are not well behaved to predict the tendency. In addition, we also filter stocks with ROE greater than 0.15 and net cash ratio greater than 0.9. We conclude that according to the financial indicators based on the just-released annual report of the company, the predictability of the stock price movement on the second day after disclosure is weak, with maximum accuracy about 59.6% and maximum precision about 0.56 on our test set by the random forest classifier, and the stock filtering does not improve the performance. And random forests perform best in general among all these models which conforms to some work’s findings.

1. Introduction
In the last century, people still relied on financial data analysis to establish a variety of financial indicator systems to evaluate the value of an enterprise, the growth of the enterprise, and the changes in the stock price of the enterprise, and finally people decided whether to invest on it. However, many things can affect the stock price of a company. Financial indicators are an important aspect, and there is no way for humans to comprehensively and objectively evaluate the impact of each indicator on the stock price [1]. Since the advent of computers, the workload of human beings has been greatly reduced, and people can pay more attention to the correlation between various financial data, and whether financial data can objectively reflect the current state of the company [2]. In the early days, computers usually did the computing work instead of the analysis work. Analysis and decision-making have always been seen as
the privilege of human beings. Humans seem to have always been the most reliable decision-makers. However, the emergence of machine learning, neural networks, etc. has made people discover that computer decisions are more accurate and reliable than humans in some aspects, since computers can beat humans in games like go, which require extremely human talent, they may be able to achieve decent results in areas that humans can not do well. For example, in a large number of image recognition, such a workload is too large, and manual classification requires a lot of time and money. For example, the proposal of VGG [3] is revolutionary, and there are more effective algorithms about image recognition, including ResNet [4].

In this article, we try to predict stock price movement with machine learning. The difference from previous work is that in the past, a lot of work was to predict the price tendency in normal time, but our work focuses on the time the company's annual report is released. Because at this special time, the new released corporate performance may directly promote the change of stock price, which is more favorable for stock price forecast. We compare commonly used classification models, ranging from simple classification model (logistic regression and decision tree) to advanced models (random forest and neural network) and a few-shot learning model (prototypical networks), which perform well in general data prediction tasks [5].

2. Related Work
There are many papers on predicting stock prices. Shah et al. [1] review the related work and conclude that more and more machine learning methods are now being used to predict the tendency of stock price. For example, ANN (artificial neural network) can be used to predict stock price, and SVR (support vector regression) can be used to predict the index of market. Jiang [5] wrote another review paper which divides stock forecasts into four types according to the output and frequency of the results: daily classification, daily regression, intraday classification and intraday regression. The author also carefully studies different stock markets, record and analyze them carefully. In the following, we list a few very related papers.

Milosevic et al. [6] attempts to use manual feature screening. The authors use a number of different models including SVM, Naive Bayes, logistic regression and random forest. They use more than 1000 stocks, and find that the random forest prediction accuracy is stronger than other models', which is very close to our experimental results (Section 4).

Hegazy et al. propose an algorithm called PSO and a new kind of SVM named LS-SVM. The PSO algorithm is used to optimize the LS-SVM so that it can select the best free parameter combination for LS-SVM, and avoid overfitting and local minima problems, and improve prediction accuracy. This model uses stock historical data to make predictions and compares with artificial neural networks using the Levenberg-Marquardt (LM) algorithm [7].

RA Araújo et al. [8] propose a new deep learning model, called DIDLP, each layer of this model contains a special linear layer and an activation function, and they improve the gradient descent method, which is more suitable for their own model, used to optimize the parameters of this complex model. They do the experiment with multiple financial time series, and they conclude that financial time series are predictable.

Dey et al. use the novel XGBoost method, in addition to other models such as MLP, where XGBoost achieves excellent results. But the authors' test accuracy on the same stock is very high, probably because of overfitting [9].

Karathanasopoulos et al. [10] propose a hybrid model where they combine momentum effects with a new method named Deep Beliefs Networks (DBN in short) to predict the Dubai Financial general index. They compare this model with three linear models. And the mixed model gives more stable results. An innovative method for selecting inputs using momentum effects is also introduced.

Yang et al. [11] use the data of the Chinese stock market to train a neural network. However, their treatment of the data set seems unreasonable, they do not set a validation set, nor do they have batch normalization of the data, such a data set does not seem to be reliable, so the test results are not very convincing.
In the experiments of other scholars, some are very similar to our experimental results. Milosevic et al. find that random forest has better results than SVM, logistic regression and other models do on the accuracy. Ballings et al. compare the AUC of AdaBoost, kernel factory, logistic regression, KNN and so on [12]. They test them on the data set of more than 5000 European listed company stocks, and finally find random forest has the best AUC, which is very similar to our results, traditional algorithms like random forest have better performance in stock prediction. Milosevic et al. try to predict whether some companies’ value will be 10% higher or not over the period of one year. Ballings et al. try to predict how many stocks will rise in a year.

**Our contributions.** We try to predict the direction of stock price on the second day after the disclosure of the annual report in China’s stock market. It is a different scenario than most works.

### 3. Experiment Setup

#### 3.1 Evaluating Models

In this paper, a variety of machine learning algorithms are used in the experiments, including decision tree [13], logistic regression [14], random forest [15], MLP [16], prototypical networks [17].

**Decision tree.** The decision tree algorithm adopts a tree structure and uses layer-by-layer reasoning to achieve the final classification. The decision tree includes a variety of elements, including root nodes, internal nodes, and leaf nodes. It judges the attribute value at a node, and then decides which node to enter to get the final classification.

**Logistic regression.** Logistic regression is often used for binary classification. Logistic regression is loved by the industry because of its simplicity, parallelizability, and strong interpretability. The essence of logistic regression is to assume that the data obeys logistic distribution, and then it uses the maximum likelihood estimation for parameters.

**Random forest.** The basic unit of random forest is decision trees. Every tree of random forest is used to classify, and represents a type. Random forest is a kind of ensemble method.

**Multilayer perceptron network.** It is a kind of Artificial Neural Network (ANN). Neurons between two adjacent layers are fully connected, and MLP can deal with non-linear questions. The most simple MLP only has three layers, but it still can get the classification very well on some questions [18].

**Prototypical networks.** The basic idea of prototypical networks is to create a prototypical representation for each classification. And for a query that needs to be classified, it is determined by calculating the distance between the prototype vector of each category and the query. This method learns an embedding function $f_0$ which is a neural network, through which the feature vector of the sample is converted into a representative vector, and then the average of representative vectors of a class is calculated, called a *prototype*. The classification of a query is made by the standard euclidean distance to these center points of classes. Prototypical networks divides the data into three categories, i.e., support set, query set and test set.

#### 3.2 Indicators and Labels

We divide these data indicators into two categories (with overlapping indicators):

- **Key indicators**: the year-on-year growth of operating income and net profit for the full year and the fourth quarter of 2020, $\text{PE}_{\text{TTM}}$, $\text{PB}$, the historical percentile of $\text{PE}_{\text{TTM}}$ and $\text{PB}$, and the percentage change of the stock price over the past year.

- **Expanded indicators**: expanded indicators also include ROE, gross profit margin on sales, net-to-now ratio besides key indicators.

The two types of indicators are then trained with the same machine learning method to evaluate the impact of different indicators on the predicted results. During the experiment, some financial indicators are taken out to try, such as earnings per share, operating cash flow per share, operating income, net profit, and net assets per share. These financial indicators are found to have a small impact on the prediction, so we eliminate them.
**Labels:** On the second day after the company's annual report is disclosed, we compare stock prices than the previous day. If the amount of increase (percentage) of a stock price on the second day is positive, it is labeled as 1; otherwise, it is labeled as 0. Note that the percentage is not the original one and is subtracted by the amount of increase of its belonged index in order to eliminate the index impact.

### 3.3 Data Collection
We use crawlers to download the disclosure data of the 2020 annual report of the Chinese stock market by 2022-03-24, 00:00:00 from https://data.eastmoney.com/bbsj/202012/yjbb.html and only retain the stocks of CSI 300, which are 300 stocks in China with good liquidity and large scale. Then we crawl the historical stock prices of CSI 300 and the historical data of the CSI 300 index from NetEase Finance https://money.163.com/. According to the price on the day of the disclosure date of each stock, we compare the average stock price in the same month a year ago, and supplement the year-on-year price percentage change as a feature, then we use the existing data to calculate the PE	extsubscript{TTM}, PB, and the historical percentiles of PE	extsubscript{TTM} and PB, to serve as features. There are also two important features, the year-over-year increase in operating income and net profit in the fourth quarter which we can calculate from historical data.

### 4. Experiments Result
We use two sets of financial indicators and trained on five models, respectively (decision tree, logistic regression, prototypical networks, random forest, neural network). We obtain accuracy, AUC, precision, recall and F1 of training, validation and test. Through these indicators, we evaluate the models’ prediction results, and make a summary.

#### 4.1 Decision Tree
The main hyperparameters involved include: (a) criterion (gini or entropy). (b) splitter (best or random). (c) min_samples_split: it is the smallest number of sample points to split a tree node. If it is a floating point number, it donates a percentage. The min_samples_split here are 0.05, 0.1, 0.15, 0.2, 0.25, 0.3. The results tested on key indicators and expanded indicators are shown in Table 1 and 2, respectively. The precision indicates 45.2% of the stocks that are predicted to increase actually increase, and the recall indicates that the proportion of actually increasing stocks that are predicted to increase is 39.8%. The training accuracy on key indicators is significantly better than the test accuracy, even reaching 67.1%, which is largely the result of overfitting, because the test set is still at 55.0%.

Note that in the following of the paper, we only talk about the scores on the test set which is most objective.

The accuracy, AUC and precision of the expanded indicators are slightly better than those of key indicators, but the improvements are less than 3%. And the recall becomes lower when we add more indicators. Note that we care more about precision because there is no enough short mechanism in China and most investors do not short. The precision is less than 50% meaning that there is no arbitrage opportunity. In summary, decision tree is not well behaved in predicting stock prices, and the accuracy is low and the precision is not even applicable.

| Table 1. The results of decision tree training and testing on key indicators (the optimal hyperparameters are: criterion='gini', min_samples_split=0.05, splitter='random'). |
|-----------------|--------|------|------|------|        |
|                 | Accuracy | AUC  | Precision | Recall | F1      |
| Training        | 67.1%    | 72.6%| 64.2%    | 54.4%  | 58.8%   |
| Validation      | 53.5%    | 53.5%| 45.3%    | 39.8%  | 42.2%   |
| Test            | 55.0%    | 52.8%| 45.2%    | 39.8%  | 42.4%   |
Table 2. The results of decision tree training and testing on expanded indicators (the optimal hyperparameters are: criterion='gini', min_samples_split=0.1, splitter='random').

|          | Accuracy | AUC    | Precision | Recall | F1   |
|----------|----------|--------|-----------|--------|------|
| Training | 66.0%    | 71.5%  | 63.2%     | 51.9%  | 56.3%|
| Validation | 53.3%  | 52.4%  | 45.8%     | 38.2%  | 41.2%|
| Test     | 57.9%    | 54.5%  | 49.0%     | 34.7%  | 40.7%|

4.2 Logistic Regression
The main hyperparameters involved in logistic regression are: (a) penalty ('l1', 'l2'). (b) C is inverse of regularization strength which is float number greater than 0. (c) multi_class: it tells the model whether the classification problem to be handled is binary or multi-class. (d) solver: it is used to specify the optimization method of the logistic regression loss function.

In this experiment, the parameters "l1", "l2" of penalty are set; the parameter adjustment range of C includes 0.01, 0.1, 0.2, 0.3, 0.5, 0.6, 0.7, and 1. The results tested on key indicators and expanded indicators are shown in Table 3 and 4.

Obviously, the result of logistic regression is still unsatisfied. The test accuracy is 56.9% and 57.7% for the two kinds of indicators, respectively, which are comparable to the results of decision tree, but recall and F1 are much lower than decision tree. The precisions are still less than 50%.

Table 3. The results of logistic regression training and testing on key indicators (the optimal hyperparameters are: C=0.2, penalty='l2').

|          | Accuracy | AUC    | Precision | Recall | F1   |
|----------|----------|--------|-----------|--------|------|
| Training | 57.0%    | 51.1%  | 53.3%     | 5.9%   | 10.5%|
| Validation | 56.3%  | 49.2%  | 39.8%     | 4.9%   | 8.6% |
| Test     | 56.9%    | 49.4%  | 36.7%     | 5.1%   | 8.9% |

Table 4. The results of logistic regression training and testing on expanded indicators (the optimal hyperparameters are: C=0.7, penalty='l2').

|          | Accuracy | AUC    | Precision | Recall | F1   |
|----------|----------|--------|-----------|--------|------|
| Training | 57.1%    | 52.0%  | 54.2%     | 5.8%   | 10.4%|
| Validation | 56.0%  | 49.9%  | 36.5%     | 4.6%   | 8.1% |
| Test     | 57.7%    | 50.0%  | 40.9%     | 4.2%   | 7.6% |

4.3 Random Forest
The main hyperparameters involved in random forest are: (a) n_estimators: this is the number of base estimators. (b) Other parameters are the same as decision tree.

In this experiment, we adjust the following parameters: criterions of 'gini' or 'entropy', n_estimators in the range from 1 to 50, min_samples_split in the range of (0.05, 0.1, 0.15, 0.2, 0.25, 0.3). The test results on the key indicators and expanded indicators are shown in Table 5 and 6.

The accuracies, AUCs and precisions of random forests are overall stronger than the corresponding decision trees and logistic regressions (except one case of AUC less than the decision trees), but the recall and F1 of random forest are slow. One thing is note is that the precision for the expanded indicators is 56.0% which is applicable.

Table 5. The results of random forest training and testing on key indicators (the
optimal hyperparameters are: criterion='gini', min_samples_split=0.1, n_estimators=11).

|          | Accuracy | AUC  | Precision | Recall | F1  |
|----------|----------|------|-----------|--------|-----|
| Training | 67.7%    | 76.4%| 75.9%     | 36.5%  | 49.3%|
| Validation | 56.3% | 54.8%| 48.7%     | 22.1%  | 30.4%|
| Test     | 57.7%    | 52.9%| 48.2%     | 24.5%  | 32.5%|

Table 6. The results of random forest training and testing on expanded indicators (the optimal hyperparameters are: criterion='gini', min_samples_split=0.2, n_estimators=16).

|          | Accuracy | AUC  | Precision | Recall | F1  |
|----------|----------|------|-----------|--------|-----|
| Training | 63.2%    | 72.5%| 75.9%     | 21.3%  | 33.2%|
| Validation | 56.3% | 54.4%| 48.5%     | 11.1%  | 17.9%|
| Test     | 59.6%    | 52.9%| **56.0%** | 13.0%  | 21.1%|

4.4 MLP

We design the network structure of MLP as a total of three linear layers using the ReLu activation function and dropout is added to each layer, and the input and output dimensions of each layer are (input_dim, 64), (64, 32), (32, 2), where input_dim is the dimension of the input vector (11 for the key indicators and 14 for the expanded indicators). The loss function uses CrossEntropyLoss in pytorch, and the optimizer uses SGD.

The hyperparameters we adjust are: (a) learning rate: it indicates the step size taken during gradient descent. (b) batch size: it is the number of examples which are chosen to participate the training, validation and test progress. (c) k: this represents the number of k-fold cross-validation. (d) epoch size: it represents the number of iterations.

The accuracy on the test set of key indicators and expanded indicators is shown in Table 7 and 8. On these two sets of indicators, the performance of MLP is not ideal, the accuracy of expanded indicators is about 49-50%.

Table 7. The results of MLP testing on key indicators (the optimal hyperparameters are: learning rate=0.02, batch size=5, k=5).

|          | Accuracy | AUC  | Precision | Recall | F1  |
|----------|----------|------|-----------|--------|-----|
| Training | 57.1%    | 49.9%| 39.7%     | 0%     | 15.16%|
| Validation | 57.0% | 49.6%| 37.7%     | 0%     | 13.8%|
| Test     | 50.3%    | 50.3%| 50.5%     | 11.4%  | 17.19%|

Table 8. The results of MLP testing on expanded indicators (the optimal hyperparameters are: learning rate=0.02, batch size=5, k=5).

|          | Accuracy | AUC  | Precision | Recall | F1  |
|----------|----------|------|-----------|--------|-----|
| Training | 57.2%    | 50.0%| 40.5%     | 0%     | 15.7%|
| Validation | 57.3% | 49.9%| 39.9%     | 0%     | 14.9%|
| Test     | 49.1%    | 49.1%| 44.6%     | 10.1%  | 15.9%|
4.5 Prototypical Networks

The embedding function in a prototypical network has a total of three linear layers, and the input and output are: \((\text{input\_dim}, 64), (64, 32), (32, 2)\), where \(\text{input\_dim}\) is the dimension of the input vector (11 for key indicators and 14 dimensions for expanded indicators), and a dropout layer is added to each linear layer, the parameter is set to 0.5, and the optimizer uses SGD.

The hyperparameters include: (a) epoch size: this represents the number of iterations. (b) learning rate: it represents the step size taken during gradient descent. (c) output\_dim: it indicates the dimension of the feature vector output by the embedding function. (d) support size. the number of samples taken each time should not be too large in order to meet the requirements of the few-shot learning method. In this experiment, it is set to 5. (e) query size: it is the size of the query set, and it is set to 2.

The accuracy tested on key indicators and expanded indicators are shown in Table 9 and 10. About prototypical networks, when the actual test set the support size to 100 and the query size to 20, the accuracy changes about only 1%, so the size of the support size and query size has relatively little effect on the results. The epoch size has little effect on the prediction results, because the number of iterations ranges from 10 to 10,000, and the final results are not very different.

| Table 9. | The results of Prototypical Networks testing on key indicators (the optimal hyperparameters are: learning rate=0.02, batch size=5, k=5). |
|----------|----------------------------------------------------------------------------------------------------------------------------------|
|          | Accuracy | Precision | Recall | F1    |
| Training | 73.6%    | 9.5%      | 3.0%   | 4.5%  |
| Test     | 49.4%    | 49.3%     | 41.9%  | 42.9% |

| Table 10. | The results of Prototypical Networks testing on expanded indicators (the optimal hyperparameters are: learning rate=0.02, batch size=5, k=5). |
|-----------|----------------------------------------------------------------------------------------------------------------------------------|
|           | Accuracy | Precision | Recall | F1    |
| Training  | 72.1%    | 6.0%      | 1.8%   | 2.7%  |
| Test      | 50.5%    | 50.3%     | 48.0%  | 48.0% |

4.6 Filtered Stocks

We choose stocks with ROE greater than 0.15 and net cash ratio greater than 0.9, which are companies with good financial performance, and we test these stocks with random forest, MLP and prototypical network. In short, the precision, recall and F1 are between 40% and 56%. However, we find that the prediction results do not change much for companies compared with the previous results without filtering. We use random forest which has the best performance in the experiment before, but the results are very similar and slighted degraded for accuracy and precision. MLP and prototypical network have the same situation. The results are shown in Table 11, 12 and 13. Prototypical network does not perform well in the prediction of stock price tendency, and the accuracy always fluctuates around 50%.

| Table 11. | The results of random forest testing on filtered indicators (the optimal hyperparameters are: criterion='gini', min_samples_split=0.2, n_estimators=7). |
|-----------|------------------------------------------------------------------------------------------------|
|           | Accuracy | AUC      | Precision | Recall | F1    |
| Training  | 76.7%    | 85.9%    | 76.6%     | 77.6%  | 76.9% |
Validation 57.6%  62.4%  55.0%  62.4%  57.4%
Test 57.6%  56.8%  52.0%  50.0%  51.0%

**Table 12.** The results of MLP testing on filtered indicators (the optimal hyperparameters are: learning rate=0.02, batch size=5, k=5).

|          | Accuracy | AUC | Precision | Recall | F1    |
|----------|----------|-----|-----------|--------|-------|
| Training | 50.6%    | 50.5%| 49.6%     | 0%     | 49.1% |
| Validation | 49.7%  | 49.6%| 48.8%     | 0%     | 47.3% |
| Test     | 49.3%    | 49.2%| 48.2%     | 40.2%  | 42.9% |

**Table 13.** The results of Prototypical Networks testing on filtered indicators (the optimal hyperparameters are: learning rate=0.02, batch size=5, k=5).

|          | Accuracy | Precision | Recall | F1    |
|----------|----------|-----------|--------|-------|
| Training | 50.6%    | 15.9%     | 6.12%  | 8.5%  |
| Test     | 49.8%    | 47.7%     | 56.1%  | 50.2% |

4.7 Summary
For a brief summary, the best models are random forests, and next are decision trees. Neural networks do not perform good, although we only try one network architecture (and there are many others). The precision is lower than 40% in most settings except for the case of random forest under extended indicators. Expanded indicators can improve the performance but only a bit percent. You can find the code and data at https://github.com/paopaohhh/Stock.

5. Conclusion
Through strict data screening, suitable stocks are screened and the corresponding stock information is obtained. The performance of each model on two sets of financial indicators (key and extended) is slightly different. But in general, the price changes of stock prices on the second day after the disclosure date are not very predictable. In the test set, the accuracy of each model is relatively low, with a few percent above 50%. And random forests perform the best in general, only using which we can get the accuracy and precision above 55%. And the test on the chosen stocks with ROE greater than 0.15 and net cash ratio greater than 0.9 does not show prediction improvement. We conclude the predictability after annual reports disclosure is week.

6. References
[1] Shah D, Isah H, and Zulkernine F 2019 Stock market analysis: A review and taxonomy of prediction techniques *International Journal of Financial Studies* vol. 7 no. 2 p. 26
[2] Sharma A, Bhuriya D, and Singh U 2017 Survey of stock market prediction using machine learning approach in *2017 International conference of electronics, communication and aerospace technology (ICECA)* vol. 2: IEEE p 506-509
[3] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition *arXiv preprint arXiv:1409.1556*
[4] He K, Zhang X, Ren S, and Sun J 2016 Deep residual learning for image recognition in *Proceedings of the IEEE conference on computer vision and pattern recognition* p 770-778
[5] Jiang W 2021 Applications of deep learning in stock market prediction: recent progress *Expert Systems with Applications* vol. 184 p. 115537
[6] Milosevic N 2016 Equity forecast: Predicting long term stock price movement using machine learning. *arXiv preprint arXiv:1603.00751*

[7] Hegazy O, Soliman O S, and Salam M A 2014 A machine learning model for stock market prediction. *arXiv preprint arXiv:1402.7351*

[8] Araújo R d A, Nedjah N, Oliveira A L, and Silvio R d L 2019 A deep increasing-decreasing-linear neural network for financial time series prediction. *Neurocomputing* vol. 347 pp 59-81

[9] Dey S, Kumar Y, Saha S, and Basak S 2016 Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting. *PESIT South Campus*

[10] Karathanasopoulos A and Osman M 2019 Forecasting the Dubai financial market with a combination of momentum effect with a deep belief network. *Journal of Forecasting* vol. 38 no. 4 pp 346-353

[11] Yang B, Gong Z-J, and Yang W 2017 Stock market index prediction using deep neural network ensemble in 2017 36th Chinese control conference (ccc): IEEE p 3882-3887

[12] Ballings M, Van den Poel D, Hespeels N, and Gryp R 2015 Evaluating multiple classifiers for stock price direction prediction Expert systems with Applications vol. 42 no. 20 pp 7046-7056

[13] Safavian S R and Landgrebe D 1991 A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics* vol. 21 no. 3 pp 660-674

[14] Wright R E 1995 Logistic regression

[15] Belgui M and Drăguţ L 2016 Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing* vol. 114 pp 24-31

[16] Pinkus A 1999 Approximation theory of the MLP model in neural networks. *Acta numerica* vol. 8 pp 143-195

[17] Snell J, Swersky K, and Zemel R 2017 Prototypical networks for few-shot learning. *Advances in neural information processing systems* vol. 30

[18] Soni S 2011 Applications of ANNs in stock market prediction: a survey. *International Journal of Computer Science & Engineering Technology* vol. 2 no. 3 pp 71-83