Stock Returns Prediction with Various Ensemble Learning and Stacking Method

Keonwoo Kim
School of Industrial Engineering
UNIST
keonwookim@unist.ac.kr

Minki Kim
School of Business Administration
UNIST
Mingi3314@unist.ac.kr

Abstract

Predicting future stock returns is one of the hardest challenges in finance and AI area. One main reason why it is hard to predict future prices is market efficiency. If the market is efficient, all available information is already reflected in stock prices and stock price would follow the random walk. In this study, we assumed market is not efficient and technical indicators, fundamental indicators, and other macro factors would affect to the stock price based on Efficient Market Hypothesis. So, we utilized these features as our predictor variables. Feature extraction with Principal Component Analysis dimension reduction method is used to make dimension of input data simpler. Then, several ensemble learnings are used to solve out multi-class classification methods. Both parallel ensemble learning which reduces the bias and sequential ensemble learning which decreases the variance were used. After making models, they were stacked into one model to improve predictions. The results came out better than usual stock prediction models. However, it is not enough to say that our model shows great predictive performance compared to other multi-class classification problem. Thus, we concluded that market is efficient and thus it is hard to predict future stock returns.

I. Introduction

Since the creation of artificial intelligence at the Dartmouth Conference in 1956, artificial intelligence has made a lot of improvements overcoming the dark ages in the late 20th century. Computational power has been increased, which became a cornerstone of development in Machine Learning and Deep Learning, subfields of artificial intelligence. Nowadays, artificial intelligence shows its great performance in various fields including Computer Vision, Automatic Speech Recognition, Pattern Recognition, Natural Language Processing, etc.

With a heightened interest in artificial intelligence, the financial industry also paid attention to it. Especially, the Recurrent Neural Network's capability of predicting future data in time series models seemed hopeful for those who working as traders and researchers in the financial area. Despite the expectation of a rosy future of predicting securities prices using machine learning and deep learning model, However, a lot of models failed to show their usefulness in the real trading environment and forecasting future stock prices remains a challenging task.

Although there are several reasons for the failure of the attempt to predict stock price under machine learning or deep learning frameworks, a characteristic of stock prices following random walk would be a one of the main reasons. A question of whether stock price follows random walk or not ultimately comes down to the market efficiency. Efficient Market Hypothesis (EMH) is one of the most famous hypotheses about market efficiency in the financial economics area.

According to the Efficient Market Hypothesis, all available information is already reflected in the stock price. So the asset price moves like a random walk and it is impossible to predict it and beat the market consistently. In detail, EMH can be classified into three types: weak-form EMH, semi-strong-form EMH, and strong-form EMH. First, if the weak form of EMH holds, all historical prices and returns already would be reflected in the stock price. Second, if the semi-strong form of EMH holds, all public information already would be reflected. And if the strong form of EMH holds, all information including both public and private already would be reflected in the stock price.

If the Efficient Market Hypothesis holds and market turns out to be efficient, it is impossible to predict future stock prices and a lot of attempts would go back to nothing. However, there also exist criticisms against EMH. Investors, including the likes of Warren Buffet, George Soros, and researchers have disputed the efficient-market hypothesis both empirically and theoretically. Behavioral economists attribute the imperfections in financial markets to a combination of cognitive biases such as overconfidence, overreaction, representative bias, information bias, and various other predictable human errors in reasons and information processing. Anomalies are another ground for EMH critics. There are well-known EMH anomalies such as “Small-minus-big” factor, “High-minus-low” factor, Momentum factor, and so on. These factors disprove the state of EMH that all available information is reflected to the stock prices.

In this study, we are going to assume that market is not efficient and try to predict future cumulated returns after one month. As our assumption goes against the market efficiency, we are going to use our predictor variables as historical stock prices data and publicly available data such as published account statements and information found in annual reports. With these data, we build several Ensemble learning models and stacked Ensemble learning models.

II. Literature Review

A. Stock price prediction

Stock price prediction can be classified into two main parts. First one is a method based on the statistical time series prediction. Ariyo et al. (2014) attempted to predict future stock prices using Autoregressive Integrated Moving Average models in their paper, 'Stock Price Prediction Using the ARIMA Model' [1]. With Published stock data obtained from New York Stock Exchange and Nigeria Stock Exchange, they revealed the ARIMA model has a strong potential for short-term prediction. Friedman and Shackmurove (1997) tried to figure out the stock movement with a Vector-Autoregression model in their study 'Co-movements of major European community stock markets: A vector autoregression analysis' [2]. They found out that the results of the Granger causality tests may lead to a conclusion that EC markets are inefficient since current returns are predicted by their own and by other market's lagged values.

Another part of stock price prediction is a method based on machine learning and deep learning framework. Adebiyi et al. (2014) compared previously described ARIMA model and deep learning model in their paper, 'Comparison of ARIMA and Artificial Neural Networks for Stock Price Prediction' [3]. They compared ARIMA model to ANNs model in forecasting Korean Stock Price Index and concluded that ANN-based approach showed better performance than the ARIMA models. Selvin et al. (2017) tried to predict stock price using recurrent neural network and convolution neural network in their study, 'Stock price prediction using LSTM, RNN and CNN-sliding window model.' [4]. They applied a sliding window approach for predicting future values on a short-term basis and quantified the performance of the models with percentage error. Patel et al. (2015) used four machine learning algorithms for prediction in stock markets in their paper, 'Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques' [5]. They focused on data pre-processing to improve the prediction accuracy and discretized technical indicators by exploiting the inherent opinion. As a result, Prediction accuracy of algorithms increases when discrete data was used.

B. Market efficiency

Whether the market is efficient or not has caused controversy since Eugene F. Fama (1965) made public his doctoral thesis, 'The Behavior of Stock Market Prices' [6]. He persuasively suggested that in a dynamic market involving intelligent investors armed with a lot of information, it would clearly reflect all possible information in stock prices. Samuelson, P.A. (1965), supported Fama's arguments in his writing, 'Proof that properly anticipated prices fluctuate randomly' and contributed to hypothesize the Efficient Market Hypothesis [7]. Burton G. Malkiel (1963) also stated that market is efficient, and securities prices follow random walk in his famous writing, 'A Random Walk Down Wall Street' [8]. He argued that asset prices typically exhibit signs of a random walk and that one cannot consistently outperform market averages.

C. Stock Prediction with Machine Learning models

In 'On Stock Market Movement Prediction Via Stacking Ensemble Learning Method', written by Samuel Asante Gyamerah, different machine learning techniques are used to predict the movement on the stock market. Stacking ensemble learning method with adabosot and K-NN as base learner and gradient boosting machine as meta learner showed outperformed performance compared to an individual classifier. This gave us confidence that stacking's performance in stock price prediction is also superior to that of individual classifiers. Also, in 'Deep learning for Stock Market Prediction', written by Mojtaba Nanbipour, stock value predictions are created for 1, 2, 5, 10, 15, 20 and 30 days in advance. The machine learning algorithms which are mainly tree-based model such as decision tree, random forest, adaboost, gradient boosting and XGBoost are used. Also, the deep learning techniques which are ANN, RNN and LSTM are utilized. Among them, LSTM shows the best performance and for tree-based models, boosting types techniques show the significant performance. Through this, we thought to utilize the boosting-based ensemble model as a base-learner for stacking.

III. Method

A. Data

Basically, we use the data consists of Samsung Electronics' Open, High, Low, Close prices and volumes from 2011 to 2020. And we further utilize other predictor variables based on the Efficient Market Hypothesis. As we assumed, if EMH does not hold, then it is able to predict future stock returns using a variety of information related to it. More specifically, if weak-from EMH does not hold, future stock prices will be affected by historical prices and technical indicators which is a mathematical calculation based on historical price, volume, open interest information that aims to forecast financial market direction. If semi-strong form of EMH does not hold, future stock prices will be affected by publicly available information such as published accounting statements and information found in annual reports. These are referred as fundamental indicators and we selected several representative financial data about the firm. The last step of EMH is a strong form. If the strong form of EMH does not hold, securities prices would be affected by private information. However, it is hard to gather private information data. Instead, rather than using private information data as predictor features, we utilize other macro factors that would affect to the stock price.

TABLE 1
PREDICTOR VARIABLES - TECHNICAL INDICATORS

Price indicator	MA (moving average)
	BB (Bolinger Band)
	PSAR (Parabolic SAR)
Momentum indicator	RSI (Relative Strength Index)
	MACD (Moving Average Convergence & Divergence)
Volume indicator	CO (Chaikin Oscillator)

Volatility indicator	ATR (Average True Range)
Cycle indicator	HT_DCPERIOD (Hilbert Transform – Dominant Cycle Period)

TABLE 2
PREDICTOR VARIABLES - FUNDAMENTAL INDICATORS

Fundamental indicator	Net income
	Operating income Growth Ratio
	PBR (Price to Book ratio)
	PER (Price Earning ratio)
	ROE (Return on Equity)
	Debt Ratio
	Asset turnover ratio
	Current ratio
	Gross profit growth ratio
	Operating cash flows
	Shares outstanding

TABLE 3
PREDICTOR VARIABLES – MACRO FACTORS

Global market index	Dow-Jones, NASDAQ, S&P500
	HangSeng, Sanghai, Nikkei
	EuroSTOXX
	FTSE
Commodity future	Crude oil, Natural gas
	Gold, Silver, Copper, Aluminum
Foreign Exchange (FX)	KRWCHN (Won / Yuan)
	KRWEUR (Won / Euro)
	KRWUSD (Won / USD)
Other index	Semiconductor index

Along with these predictor variables, our target variable is 1-month later cumulative returns. Cumulative returns are categorized into multi-classification label.

B. Feature extraction

In this paper, rather than using feature selection, feature extraction is used in the method of selecting features of input data to fit into the model. Principal Component Analysis, PCA, is chosen among feature extraction methods. It is a dimensionality-reduction method which is often utilized to reduce the dimensionality of data. The principle of operation of this technique transforms large number of features into a smaller one that still preserving the information of original set as possible. The first procedure on PCA is standardizing the variables in order to make each variable contributes equally to the analysis. (1) illustrates the method of calculating standardization.

$$z = \frac{value - mean}{standard\ deviation} \qquad (1)$$

The next procedure is to identify relationship among variables by computing the covariance matrix depicted on (2).

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix} \quad (2)$$

If the covariance value has positive value, it can be seemed that two variables are somehow correlated and vice versa if negative. Further, eigenvectors and eigenvalues of the covariance matrix is calculated to find out principal components. They are constructed as first PC has the largest possible variance which illustrates the original data, and the remaining PCs are also formed in descending order. In the last step, by multiplying the transpose of the original data by transpose of feature vector we get new dataset in lower dimension which we were expected.

C. Ensemble learning

1) Random Forest

Random Forest is one of the strongest parallel ensemble learning which uses bagging method while classifying. It is a combination of tree predictors, where each one depends on the values of a random vector which is sampled. The principle of voting is that each tree gives its own predicted values and the whole random forest model averages it together. In other words, a group of ‘weak learners can come together to form a ‘strong learner’ which is illustrated on Fig.1. There are a lot of advantages while using Random Forest. It has high accuracy results compared to other models such as LDA or Logistic Regression. Also, Random Forest is also immune against overfitting problem since weak learners do not overfit to data. This is because, unlike a single tree model which has low bias and high variance when it has deep depth, Random Forest maintains a bias while reducing variance so to increase the performance without considering overfitting.

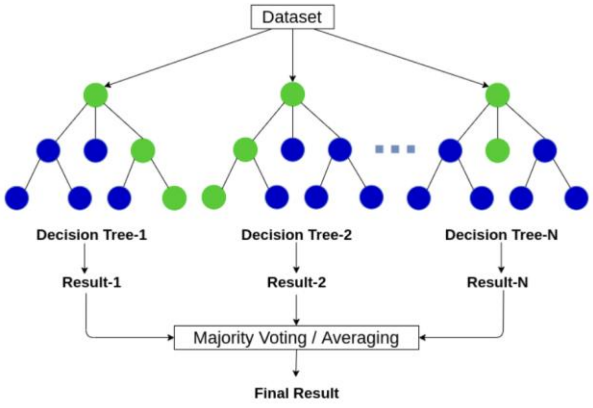


Fig. 1 Random Forest

2) Extra Trees Classifier

Extra Tree Classifier is also one of the strongest parallel ensemble learning which works similar to Random Forest. The overall method of learning mechanism is same as Random Forest, but the only different thing is that Extra Tree pursues extreme randomness over Random Forest. In order to make the tree more random, instead of finding the optimal threshold, such as Random Forest method, it randomly splits the tree using randomly selected candidate properties from Random Forest method and then choose the best split nodes among them. Thus, this is why it is called as Extreme Decision Tree in a way that adds more randomness in Random Forest. It has the advantage of having a relatively high bias while having a lower variance, and of being able to learn in a faster speed compared to Random Forest methods which spend a lot of time finding optimal thresholds.

3) Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting, XGBoost, is one of the strongest sequential ensemble learning which uses boosting method while classifying. XGBoost is developed to increase speed and performance and reduce overfitting by introducing regularization parameters and to compensate for Adaboost's shortcomings. Gradient Boosting Tree uses CART (Classification and Regression Tree) in sequential learning processes as a weak learner illustrated on Fig 2. These trees are similar to decision trees but are summed using consecutive scores allocated to each leaf to make a final prediction. A score 'w' is calculated on predicting 'y' on each iteration 'i' while developing trees. The learning process is aimed to minimize the overall score consisting of a loss function of 'i-1' and a novel tee structure of 't'. This allows algorithms to grow trees sequentially and learn from previous iterations. It also addas a method for handling overfitting like Adaboost or Random Forest to the regularization parameters found in gradient boosting. XGBoost has regularization parameters that successfully reduce speed and variance compared to other boosting algorithms such as Adaboost. However, it has difficulty on expressing visualizations, and it can be a problem to tune a model with considerable background knowledge and high time cost.

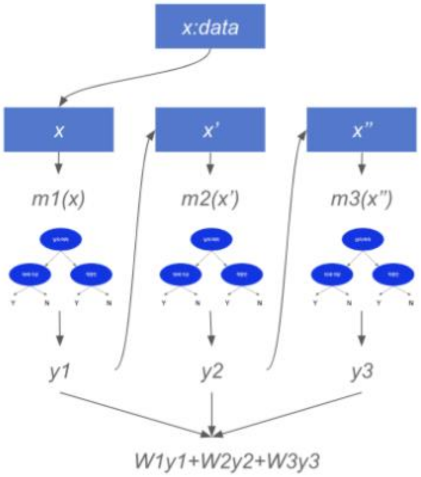


Fig. 2 Boosting Method

4) Light Gradient Boosting (LightGBM)

Light Gradient Boosting, LightGBM, is also one of the strongest sequential ensemble learning which works similar to XGBoost. XGBoost was a model developed to compensate Adaboost's shortcomings, while LightGBM is a model developed to compensate XGBoost's shortcomings. While doing hyperparameter tuning in XGBoost using grid-search method, it takes a long time, but LightGBM compensates for this problem. This enables efficiently processing large amounts of data, uses fewer resources than other models, and shows faster speed. LightGBM works differently from traditional gradient boosting algorithms. Existing boosting models use tree level-wise stretching methods, while LightGBM uses tree level-wise stretching methods illustrated in Fig 3. Level-wise tree analysis reduces the depth of the tree due to balancing the tree level, so it needs more complexity computation used for balancing. On the other hand, LightGBM does not balance but generates the depth of the tree, proceeding by continuously rapidly stretching the leaf noeds.

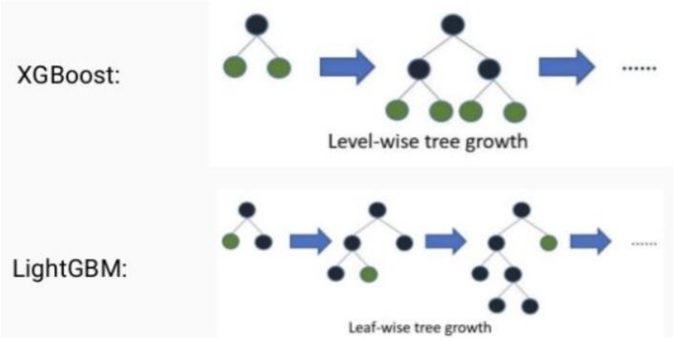
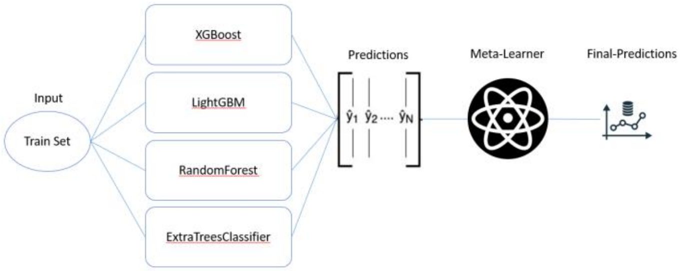


Fig. 2 Stretching Method

5) Stacking

Stacking is one of ensemble models which is often used to improve the performance of model. It is learned by using predicted data by the individual model as a training set again. Thus, the Stacking ensemble largely requires two kinds of models, one of which is individual model which uses original data to train, and the other one is final model which uses predicted data to train. In other words, the principle in which the y-prediction and the actual value act as independent and dependent variables.



IV. APPLICATION

The shape of the original data has 33 columns, which are treated as features. Using feature extraction method by PCA, the posterior number of features changed into 4. To maintain the original data information as much as possible, axis with having large variances were selected.

Parallel ensemble learnings, Random Forest and Extra Trees Classifiers, were used to predict stock returns' class. Both ensemble classifiers share same hyperparameters which are 'n_estimators', 'max_depth', 'min_sample_leaf' and 'min_samples_split'. Since they share same hyperparameters, the candidates in each hyperparameter were set as same which will be utilized while computing 'grid-search' method to find out the best hyperparameter set. Candidates for hyperparameter is illustrated in Table 4.

TABLE 4
HYPERPARAMETER CANDIDATES FOR PARALLEL ENSEMBLE

hyperparameter	candidates
n_estimators	range (100,500,100)
max_depth	[4,6,8]
min_samples_leaf	[3,4,5,6]
min_samples_split	[2,3,4]

Sequential ensemble learnings, XGBoost and LightGBM were used to predict stock returns' class. For setting hyperparameters on each model, some candidates, in each hyperparameter was given as candidates since 'grid-search' method was used to find out best set.

First of all, for XGBoost, since the problem in this paper is multi-class classification problem, 'multi: softmax' was set as objective function with 5 number of classes. And XGBoost's hyperparameter candidates set is illustrated on Table 5.

TABLE 5
HYPERPARAMETER CANDIDATES FOR XGBOOST

hyperparameter	candidates
n_estimators	range (100,500,100)
max_depth	[4,6]
min_child_weight	[1,3,5]
num_class	[5]
gamma	[0,0.1]
sub_sample	[0.6,0.8]
colsample_bytree	[0.6,0.8]
learning_rate	[0.05,0.1]

The number of estimators, 'n_estimators', which refers to weak learner, and the maximum depth of each tree, 'max_depth', were used. Also, 'min_child_weight' which refers to minimum summation value of observed data's weight, and 'gamma' which decides additional split on leaf node, and 'sup_sample' which controls overfitting by adjusting data sampling ratio, and 'colsample_bytree' which works in feature sampling when making trees were utilized.

Second of all, for LightGBM, its hyperparameters were set as similar to XGBoost because they work in almost same process except for stretching method. And LightGBM's hyperparameter candidates set is illustrated on Table 6. Moreover, since LightGBM can compute more complex operations, it has a greater number of hyperparameter sets which will be used on 'grid-search' than XGBoost.

TABLE 6
HYPERPARAMETER CANDIDATES FOR LIGHTGBM

hyperparameter	candidates
n_estimators	range (100,500,100)
max_depth	[4,6,8]
num_boost_round	[100,200,300]
num_class	[5]
metric	['multi_logloss']
num_leaves	[20]
sub_feature	[0.4,0.6]
learning_rate	[0.05,0.1]
min_data_in_leaf	range (30,60,10)
feature_fraction	[0.4,0.6]

Then, after setting each ensemble models, 'grid search' method was used to find out the best hyperparameters on each ensemble models. It works by checking the performance by putting hyperparameter from the set one by one. Also, the important thing on here is that we should consider the characteristics of finance stock data which has time series consecutive data. So, grid-search with using time series cross validation which uses the trained earlier continuously next process was used.

Finally, stacking ensemble part was executed. The prediction results on XGBoost, LightGBM, Random Forest and Extra Trees Classifier were used as training set on meta-learner which was set by XGBoost. So, after meta-learner was trained, the final prediction outcomes were come out with expected class.

V. RESULTS AND DISCUSSION

The model's performance was evaluated by accuracy metrics on the multi-class classification problem. Accuracy is calculated by only considering True/False actual values and True/False predicted values, illustrated on (3). Since the class ratios both on train set and test set are balanced about 20%, only concentrating on accuracy rather than using other metrics would work simpler and easier to analyze.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

To check whether the PCA, dimension reduction technique, is well applied on the model or not, both the results of the model without the PCA and the results of opposite case are experimented. The performance of each model, the performance of a model with two ensemble models stacked, and the performance of a model with three or more models staked, are largely divided into three to be analyzed.

1) Ensemble results without PCA

a. single ensemble model

Each ensemble models' accuracy is depicted on table 7. Random Forest shows the highest accuracy among ensemble models, 23.39%.

TABLE 7
SINGLE MODEL WITHOUT PCA

Models	Accuracy
XGBoost	19.51%
LightGBM	20.52%
RandomForest	23.39%
ExtraTreesClassifier	20.09%

b. dual ensemble stacking model.

Stacking models from a combination of the two ensembles is depicted on table 8. Stacking LightGBM and Random Forest shows the best accuracy in the entire model, 24.25%.

TABLE 8
DUAL ENSEMBLE STACKING MODEL WITHOUT PCA

Models	Accuracy
XGBoost + LightGBM	21.23%
XGBoost + RandomForest	21.52%
XGBoost +ExtraTreesClassifier	17.22%
LightGBM+RandomForest	24.25% (BEST)
LightGBM+ExtraTreesClassifier	23.10%
RandomForest+ExtraTreesClassifier	23.10%

c. three or more ensemble stacking model.

Stacking models from a combination of the three or more ensembles is depicted on table 9. Stacking XGBoost, LightGBM and Random Forest also shows the best accuracy in the entire model, 24.25% which is the same accuracy value from stacking LightGBM and Random Forest.

TABLE 9
DUAL ENSEMBLE STACKING MODEL WITHOUT PCA

Models	Accuracy
XGBoost + LightGBM + RandomForest	24.25% (BEST)
XGBoost + LightGBM + ExtraTreesClassifier	21.38%
XGBoost + RandomForest + ExtraTreesClassifier	20.52%
LightGBM + RandomForest + ExtraTreesClassifier	23.10%
XGBoost + LightGBM + RandomForest + ExtraTreesClassifier	24.10%

In Ensemble model without PCA, overall, models with random forest added performances were relatively higher than ensemble models based on boosting method. Also, one with stacking LightGBM and Random Forest, and one with XGBoost, LightGBM and Random Forest show the highest accuracy even higher than stacking all the four ensemble models for 0.15%.

2) Ensemble results with PCA

a. single ensemble model

Each ensemble models' accuracy is depicted on table 10. Unlike ensemble models without PCA's single model result, LightGBM shows the highest accuracy among ensemble models, 19.26%.

TABLE 10
SINGLE MODEL WITHOUT PCA

Models	Accuracy
XGBoost	17.93%
LightGBM	19.26%
RandomForest	15.49%
ExtraTreesClassifier	18.22%

b. dual ensemble stacking model.

Stacking models from a combination of the two ensembles is depicted on table 11. Stacking XGBoost and LightGBM shows the best accuracy in the entire model, 23.67%.

TABLE 11
DUAL ENSEMBLE STACKING MODEL WITHOUT PCA

Models	Accuracy
XGBoost + LightGBM	23.67%
XGBoost + RandomForest	20.66%
XGBoost +ExtraTreesClassifier	20.23%
LightGBM+RandomForest	23.39%
LightGBM+ExtraTreesClassifier	22.67%
RandomForest+ExtraTreesClassifier	19.80%

c. three or more ensemble stacking model.

Stacking models from a combination of the three or more ensembles is depicted on table 12. Stacking XGBoost, LightGBM and Random Forest also shows the best accuracy in the entire model, 24.25% which is same combination with ensemble model without PCA's three or more ensemble stacking model's result.

TABLE 12
DUAL ENSEMBLE STACKING MODEL WITHOUT PCA

Models	Accuracy
XGBoost + LightGBM + RandomForest	23.82%
XGBoost + LightGBM + ExtraTreesClassifier	21.66%
XGBoost + RandomForest + ExtraTreesClassifier	19.08%
LightGBM + RandomForest + ExtraTreesClassifier	21.52%
XGBoost + LightGBM + RandomForest + ExtraTreesClassifier	23.53%

In Ensemble model with PCA, the best model is stacking XGBoost, LightGBM and RandomForest but unlike our expectation its performance is lower than the best one on model without PCA. It is assumed to have some problem on procedure while extracting features.

To sum up, since there are five classes in this classification problem, there is a 20% probability that one will be chosen randomly, so the performance of our final model is higher than 20%, which is 24.25%, so we can say that the stock price prediction results in the stock market, which is hard to predict, are shown to have moderately high performance. However, compared to the average performances in other multi-class classification problems, it still seems to lack performance.

VI. Conclusion

With consideration about the market efficiency, we assumed market is not efficient and future stock prices are predictable. Under the assumption, we tried to predict future cumulative returns with technical indicators, fundamental indicators, and other macro factors. However, although we increase the predictive accuracy using various ensemble learning models and stacking methods, the result seems not so good. Thus, we conclude that even though the market could be temporarily inefficient due to several anomalies and behavioral psychology, eventually the market is efficient on average, and it is hard to predict future returns of stocks.

REFERENCES

[1] Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E. and S., S., 2020. Deep Learning for Stock Market Prediction. *Entropy*, 22(8), p.840.

[2] Gyamerah, S., Ngare, P. and Ikpe, D., 2019. On Stock Market Movement Prediction Via Stacking Ensemble Learning Method. *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*.

[3] Ariyo, A., Adewumi, A. and Ayo, C., 2014. Stock Price Prediction Using the ARIMA Model. *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*.

[4] Adebiyi, A., Adewumi, A. and Ayo, C., 2014. Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. *Journal of Applied Mathematics*, 2014, pp.1-7.

[5] Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V. and Soman, K., 2017. Stock price prediction using LSTM, RNN and CNN-sliding window model. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

[6] Patel, J., Shah, S., Thakkar, P. and Kotecha, K., 2015. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, 42(1), pp.259-268.

[7] Fama, Eugene F. "The Behavior of Stock-Market Prices." *The Journal of Business*, vol. 38, no. 1, 1965, pp. 34–105. *JSTOR*, www.jstor.org/stable/2350752. Accessed 17 June 2021.

[8] Samuelson, Paul A. "Proof That Properly Discounted Present Values of Assets Vibrate Randomly." *The Bell Journal of Economics and Management Science*, vol. 4, no. 2, 1973, pp. 369–374. *JSTOR*, www.jstor.org/stable/3003046. Accessed 17 June 2021.

[9] 2021. [online] Available at: <https://www.amazon.com/Random-Walk-Down-Wall-Street/dp/0393330338> [Accessed 17 June 2021].

[10]Chen, T. and Guestrin, C., 2016. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*,

[11]Zhou, A., Ren, K., Li, X. and Zhang, W., 2019. MMSE: A Multi-Model Stacking Ensemble Learning Algorithm for Purchase Prediction. *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*,

[12] Friedman, J. and Shachmurove, Y., 1997. Co-movements of major European community stock markets: A vector autoregression analysis. *Global Finance Journal*, 8(2), pp.257-277.