Min-Qi Jiang, Master Candidate
E-mail: jiangmq95@163.com
Professor Jia-Peng Liu, PhD
E-mail: jpliujp@126.com
Lu Zhang, Master Candidate
E-mail: zhanglu1001@qq.com
Department of Economics and Management, Jiliang University, China

AN IMPROVED STACKING FRAMEWORK FOR PREDICTING STOCK PRICE INDEX DIRECTION

Abstract. *Stock price index is an essential component of financial systems and indicates the economic performance in the national level. Even small improvements in its predictive performance can be very profitable and meaningful. This paper proposes an improved Stacking framework which contains multiple layers for predicting whether the stock price index will increase or decrease with respect to the price prevailing some time earlier, if necessary, a month. Support Vector Machines (SVM), Neural Networks (NN), ElasticNet-logistic regression (ENLR), Random Forest (RF) and Extreme Gradient Boosting (XGBoost) are stacked as base classifiers in the first layer. Instead of cross-validation, we deploy a more appropriate approach called timeslice grid search method, to select the optimal hyper-parameters and generate the Stacking features simultaneously. For the purpose of comparison and transparency, three meta classifiers which fuse the information of base classifiers and original data are implemented in the second layer, based on the algorithms ENLR,RF and XGBoost. In the last layer, majority vote method is used to ensemble these three meta classifiers. The results indicate that our model (namely Meta.MV) provide betterand more stable prediction performance compared to the single classifiers or even the state-of-the-art classifiers such as RF and XGBoost in terms of accuracy, kappa statistic and AUC value, and prediction performance improves as the layer rises due to the increased information content of model fusion. Our findings provide an integrated Stacking framework in the financial area.*

Keywords: *stock price index direction, timeslice grid search, Stacking algorithm, information fusion.*

JEL Classification: G17

1. Introduction

In finance, the value of the stock index is derived from stocks with high market capitalization. Due to its partially predictable characteristic, there have been many empirical researches which deal with the issues of predicting stock price index direction. On one hand, accurate predictions of the stock price index movement are

DOI: 10.24818/18423264/53.3.19.11

meaningful for developing effective market trading strategies (Leung, Daouk, &Chen, 2000), as investors can hedge against potential market risks. Even small improvements in predictive performance can be highly profitable(Ballings, Van den Poel, Hespeels, & Gryp, 2015). Furthermore, speculators and arbitrageurs are given the opportunity to maximize capital gains and minimize losses by trading in stock price indices (Kumar & M, 2006). On the other hand, stock price indices in all global financial markets have been taken into account as one of the most imperative performance evaluation measure of stock exchanges(Moein Aldin, Dehghan Dehnavi, & Entezari, 2012).

In the area of machine learning, the ensemble models which employ several learning algorithms in order to solve one specific problem (i.e. stock price index forecasting), have been shown to perform better than single models when forecasting financial time series(Ballings et al., 2015; Basak, Kar, Saha, Khaidem, & Dey, 2018; Tsai, Lin, Yen, & Chen, 2011).Stacking algorithm is a means of non-linearly combining generalizers in order to achieve higher prediction accuracy and reducing the generalization error. Different from Bagging(Breiman, 1996) and Boosting(Freund & Schapire, 1996)ensemble algorithm, Stacking is applied to base learners built by various learning algorithm. It is generally desirable that the base learners are of all types, and not just simple variations of one another. In this method, all possible ways of examining the learning set and trying to extrapolate from it are being exploited(Wolpert, 1992). Thus, the Stacking algorithm uses the base learns to span the learning space and gets more information from it. Indeed, the ensemble model built by Stacking method does have the ability to surpass some state-of-art models based on Bagging and Boosting method, such as Random Forest (RF) and Extreme Gradient Boosting (XGBoost), and has been gradually used by some top competitors in the data science competition like Kaggle.

Although developed some years ago and the advantages listed above, Stacking is less widely used than Bagging and Boosting methods. To the best of our knowledge, there were few papers using Stacking method while studying the problem of forecasting stock price index movement, partly because it is difficult to analyze theoretically. With our manuscript, we manage to start bridging this gap. Since the classification performs better than regression in predicting financial mark data(Enke & Thawornwong, 2005; Leung et al., 2000) and can execute profitable trading strategies(Cheung, Chinn, & Pascual, 2005; Pesaran & Timmermann, 1995), we construct a classification problem and follow Takeuchi and Lee (2013) and Krauss, Do, and Huck (2017) to use lagged returns of S&P500, NASDAQ and Dow Jones indices for the sake of predicting their price movement, i.e., increase or decrease.

Specifically, we spilt the entire data set which is from Jan 2000 until Nov 2018to sequentially training and testing our models. For each study period, we develop an improved Stacking framework, which contains three layers, to implement the steps in sequence. In the first layer, we deploy Support Vector Machines (SVM), Neural Networks (NN),ElasticNet-logistic regression (ENLG),Random Forest

DOI: 10.24818/18423264/53.3.19.11

(RF)and Extreme Gradient Boosting (XGBoost) as base classifiers, and use timeslice method (instead of cross-validation) to find the optimal by per-parameters and train the models. In the second layer, Stacking features are constructed by combining the prediction probability of the base classifiers and the original data. Concretely speaking, we respectively promote the model and original data. In the third layer, majority vote method is ENLG, RF and XGBoost as meta classifier to fuse the information extracted from utilized to ensemble three meta classifiers aforementioned. For model comparison and analysis purposes, the widely used metrics in machine learning area such as accuracy, sensitivity, specificity, kappa statistic, F-score and AUC values (Lantz, 2015)are considered in the study and analyzing the structure of our models is further considered.

The remainder of this paper is organized as follows. Section 2 briefly reviews the existing literature. Section 3 covers the data sample and Section 0 the framework of Stacking methodology. Section 5 presents the results and discusses key findings in our study. In the end, Section 6 concludes and provides direction for further research.

2. Literature review

Most relevant for our applications are the works of Tsai et al. (2011) , Ballings et al. (2015), Weng et al. (2018), Basak et al. (2018), Takeuchi and Lee (2013) and Krauss et al. (2017), providing initial applications of ensemble method to predict the stock or stock price index movement.

Tsai et al. (2011) deploy two types of ensemble classifiers (i.e. homogeneous ensemble classifiers and heterogeneous ensemble classifiers) by majority vote and bagging method. The author considers 19 financial ratios and 11 macroeconomic indicators in Taiwan stock market to investigate the prediction performance of stock returns. The result indicates that ensemble classifiers outperform single classifiers in terms of prediction accuracy and returns on investment. Similarly, in Ballings et al. (2015) comparative study, they benchmark ensemble methods (Random Forest, AdaBoost and Kernel Factory) against single classifier models (Neural Networks, Logistic Regression, Support Vector Machines and K-Nearest Neighbor), and predict the one-year ahead stock price direction of European companies. The AUC result shows that Random Forest is the top algorithm.

For the purpose of investigating whether the information hidden in macroeconomic variables (alone) can be used to accurately predict the one-month ahead price for major U.S. stock and sector indices. Weng et al. (2018) develop four ensemble models including quantile regression random forest, quantile regression neural network ensemble, bagging regression ensemble, and boosting regression ensemble. The result indicates that four ensembles prediction performances are superior to that of time-series models. Further more, this study proposes a hybrid approach of LSTM (Long Short-Term Memory) and proves that the macro-economic indicators are leading predictors of the price of U.S. sector indices. Basak et al. (2018)deploy Random Forest and Extreme Gradient Boosting algorithm for the classification problem which predicts whether the stock price will increase or

DOI: 10.24818/18423264/53.3.19.11

decrease, with respect to the price prevails n (i.e. 3,5,10,15,30,60,90) days earlier. Prediction performances show that the accuracies for a variety of companies have improved over existing predictions.

Takeuchi and Lee (2013) develop an enhanced momentum strategy on the U.S. CRSP stock universe from 1965 until 2009. Specifically, Deep Neural Networks (DNN) are employed as classifiers to predict if each stock will outperform the cross-sectional median return. The out-of-sample testing result proves the profitability of this strategy. Krauss et al. (2017) enhances this approach with implementing and analyzing the effectiveness of Deep Neural Networks, Gradient Boosted Trees, Random Forest and several ensembles of these methods in the context of statistical arbitrage. In addition, they promote a simple, equal-weighted ensemble method consisting of the above ensemble classifiers. The findings pose a severe challenge to the semi-strong form of market efficiency.

Regarding the existing literature, our contribution is threefold. First, we implement five machine learning techniques including Support Vector Machines, Neural Networks, ElasticNet-logistic regression, Random Forest and Extreme Gradient Boosting as the base classifiers, and use Stacking method to fuse prediction probability of these models and the original data. To our knowledge, this study is unique in deploying Stacking method on the prediction issue of stock price index movement. We deploy three meta classifiers (i.e. ElasticNet-logistic regression, Random Forest and Extreme Gradient Boosting) to carry out the above Stacking method and further develop a majority vote strategy to ensemble these meta classifiers.

Second, we provide a timeslice grid search method instead of cross-validation or artificial setting to find the valid hyper-parameters of the model. Since it will not use future information while searching the hyper-parameters, timeslice grid search method is more suitable for time series data and can effectively prevent overfitting.

Third, we create feature space by using lag returns in order to predict the stock price index direction of S&P500, NASDAQ and Dow Jones. Accuracy, sensitivity, specificity, kappa statistic, F-score and AUC value are used to evaluate the prediction performance. The result shows that our model does have the ability to surpass two state-of-the-art machine learning techniques (i.e. Random Forest and Extreme Gradient Boosting), and we further analyze the source of its great performance. All of the above contributions take a meaningful perspective on the prediction problem and are indicative for the framework of Stacking method in machine learning.

3. Research data

For the empirical application, this study utilizes total nineteen years of historical data from Jan 2000 to Nov 2018 of three stock market indices S&P500, NASDAQ and Dow Jones which are highly voluminous. All the data is obtained from https://finance.yahoo.com/ website.

DOI: 10.24818/18423264/53.3.19.11

For each study period, we generate the feature space (input) and the response variable (output) as follows:

Input: Let P_t denotes the closing price process of the stock price index. Then, we define the simple return $R_{t,m}$ for it over m periods as

$$R_{t,m} = \frac{P_t - P_{t,m}}{P_{t,m}} \tag{1}$$

In our application to daily data, we consider $m \in \left\{\{1,...20\} \cup \{40,60,...,240\}\right\}$. In other words, we follow Takeuchi and Lee (2013) and Krauss et al. (2017) and first focus on the return of the first 20 days, approximately corresponding to one trading month. Then, we switch to a lower resolution and consider the multi-period returns corresponding to the subsequent 11 months. In total, we thus count 31 features, corresponding to one trading year[1] with approximately 240 days. Using multi-period lag returns as our study features is much more convenient since its normalization properties (i.e. which is similar range of values), and can effective avoid local optima and numerical problems. Figure 1 visualize the correlation matrix of multi-period returns by the heat map, which illustrates that short-term returns ($Return.1 \sim Return.20$) and long-term returns ($Return.40,..,Return.240$) are correlated separately. Where as the correlation between these two types of returns is very low.

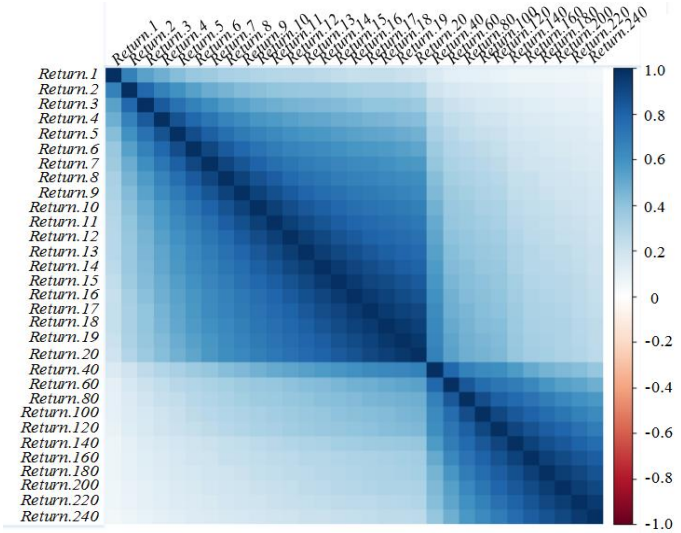

Figure 1. The correlation matrix heatmap of the returns (S&P500)

Output: We construct a binary response variable $Y_t \in \{-1,1\}$ for each stock price index and the response variable to be predicted in the t^{th} day is calculated as

[1] For simplicity, we use the expression of one year instead of one trading year in the remained of this paper (the same as trading days and trading months)

DOI: 10.24818/18423264/53.3.19.11

follows:

$$Y_t = sign\left(P_{t+20} - P_t\right) \quad (2)$$

Where P_t is the closing price of the stock price index at time t and P_{t+20} is the closing price after 20 days. In general, we use the machine learning algorithms to extract information from the historical data of the stock price index and to predict its direction (i.e. whether increase or decrease after 20 days, approximately one month), based on the momentum features.

Table 1. The number of cases in the entire data set

Stock Price Index	Increase	Decrease
S&P500	61.11%	38.89%
NASDAQ	61.19%	38.81%
Dow Jones	61.32%	38.68%

4. Methodology

Our methodology consists of three steps. First, we split our entire data into study periods consisting of training and test sets. Training sets are required for in-sample training of the specific models and test sets for their out-of-sample application. Then for each of these study periods, an improved Stacking framework consisting of three layers is developed and can be generalized as: a)In the first layer, SVM, NN, ENLG, RF and XGBoost are implemented as the base classifiers and respectively trained on the training sets. Timeslice grid search method is deployed to select the optimal hyper-parameters of the base classifiers and generate the Stacking features simultaneously; b) We develop ENLR, RF and XGBoost as our meta classifiers in the second layer to extract and fuse the information from the base classifiers and original data; c) In the last layer, majority vote method is used in order to reduce the error rate from the previous layer and to ensemble these three meta classifiers. Finally, we use some comprehensive evaluation measures to validate the prediction performance of each model. These sections follow the three steps logic outlined above and we describe the framework of the Stacking methodology using a flowchart (Figure 4).

4.1. Generation of training and test sets

We define a study period as a training-test set, consisting of 1200 days training period (approximately five years), and a subsequent 60 days test period (approximately one quarter). The long horizon of the training set enables us to have enough data for finding the optimal hyper-parameter, with the timeslice grid search method mentioned below. Furthermore, any patterns that exist are subject to change as investors themselves learn over time and compete for trading profits, thus we use a much shorter test set to timely update the model[1].

[1] Theoretically speaking, the shorter test period may generate better prediction performance due

DOI: 10.24818/18423264/53.3.19.11

4.2. Base classifier

This subsection highlights the base classifiers with the proposed Stacking strategy listed below. These base classifiers used in our study can be further summarized as three types:

1). We implement two non-linear machine learning algorithms, namely, Support Vector Machines(Vapnik, 1999) and Neural Networks[1](Haykin, 2002), for their powerful and effective performance as single models.

2). Considering the multicollinearity of the highly correlated lag returns (please see the Figure 1), we eventually choose the ElasticNet-logistic regression (Zou & Hastie, 2005)instead of logistic regression as the base classifier and the estimation of the parameter β is defined as:

$$\hat{\beta}_{Elastic} = \arg\min_{\beta}\left[-\sum_{i=1}^{n}\left\{y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)\right\} + \lambda_1\sum_{j=1}^{P}\left|\beta_j\right| + \lambda_2\sum_{j=1}^{P}\beta_j^2\right] \quad (3)$$

Where $\lambda_1\sum_{j=1}^{P}\left|\beta_j\right|$ and $\lambda_2\sum_{j=1}^{P}\beta_j^2$ are the penalty terms that regularized the estimates and depend on the positive tuning parameter (hyper-parameter), λ_1 and λ_2, which controls the amount of shrinkage.

3). We deploy two state-of-the-art ensembles models called Random Forest (Breiman, 2001)and Extreme Gradient Boosting(Chen & Guestrin, 2016) for the Stacking strategy. The objective function of Extreme Gradient Boosting (XGBoost) algorithm is computed as follows:

$$Obj(\theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K}\Omega(f_k) \quad (4)$$

Where the first part represents the training loss of the model, which can be logistic loss or squared loss, and the second part represents the sum of the complexity of each tree. The complexity of the k^{th} tree is computed as $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$, where γ is the complexity parameter, T is the number of leaf nodes, λ is a fixed coefficient, $\|w\|^2$ is the ℓ_2 – norm of leaf weight.

Furthermore, after Taylor expansion, the original objective function can be expressed as follows:

to the update frequency of the parameter, but will significantly increase the computing costs and cause data-imbalance when deploying the timeslice grid search method.

[1] In this paper, we use a feed-forward artificial neural network optimized by a Quasi-Newton method called BFGS. This method is more efficient, reliable and convenient than back propagation. Follow Ballings et al. (2015), we use one layer of hidden neurons. This is generally sufficient for classifying most data sets (Dreiseitl & Ohno-Machado, 2002).

DOI: 10.24818/18423264/53.3.19.11

$$
\begin{aligned}
Obj^{(t)} &\approx \sum_{i=1}^{n}\left[L\left(y_i,\hat{y}_i^{(t-1)}\right)+g_i f_t\left(x_i\right)+\frac{1}{2}h_i f_t^2\left(x_i\right)\right]+\Omega\left(f_t\right) \\
&=\sum_{i=1}^{n}\left[g_i w_{q(x_i)}+\frac{1}{2}h_i w_{q(x_i)}^2\right]+\gamma T+\frac{1}{2}\sum_{j=1}^{T}w_j^2 \\
&=\sum_{j=1}^{T}\left[\left(\sum_{i\in I_j}g_i\right)w_j+\frac{1}{2}\left(\sum_{i\in I_j}h_i+\lambda\right)w_j^2\right]+\gamma T \\
&=\sum_{j=1}^{T}\left[G_j w_j+\frac{1}{2}\left(H_j+\lambda\right)w_j^2\right]+\gamma T
\end{aligned}
\tag{5}
$$

Where $g_i=\partial_{\hat{y}^{(t-1)}}L\left(y_i,\hat{y}^{(t-1)}\right), h_i=\partial^2_{\hat{y}^{(t-1)}}L\left(y_i,\hat{y}^{(t-1)}\right)$, I is defined as a set of samples on each leaf, $I_j=\left\{i\mid q(x_i)=j\right\}, G_j=\sum_{i\in I_j}g_i, H_j=\sum_{i\in I_j}h_j$. Formula (5) indicates that XGBoost performs second order Taylor expression for the objective function. Thus, it can process the first and second order derivative in parallel to accelerate the convergence rate of the model while training. Meanwhile, XGBoost adds a regularized term to the objective function that smooths the contributions of Decision Trees to prevent overfitting

Since Stacking method should be applied to base classifiers built by various learning algorithms, the base models aforementioned can effectively span the learning space and thus extract more information from the original data. Figure 2 intuitively illustrate this phenomenon generated by simulated data, where the learning space of tree-based ensemble classifiers (i.e. Bagging Trees, Random Forest, AdaBoost, Gradient Boosting Decision Trees and Extreme Gradient Boosting) is overlapped in most cases due to the similarity of their algorithms. This is contrary to the models in our study, which can effectively span the learning space and focus on different aspects of the information.

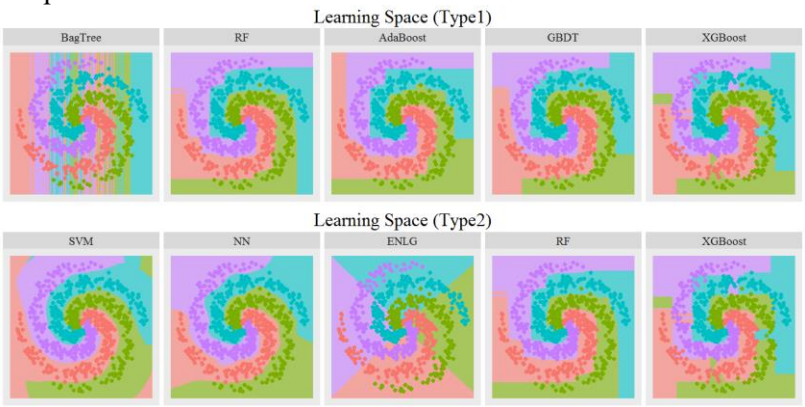

Figure 2. The learning space of various machine learning models

4.3. Timeslice Grid Search (TSGS)

In this paper, we develop a method called timeslice grid search method to

DOI: 10.24818/18423264/53.3.19.11

find the optimal hyper-parameters and generate Stacking features simultaneously. Moreover, we compare it with cross-validation to further illustrate the merits of this method.

Cross-validation is a widely used method in statistics. In k-fold cross-validation, we randomly split the training set into k folds without replacement, where k-1 folds are used for the model training and one fold is employed for testing. This procedure is repeated k times so that we obtain k models and performance estimates. We then obtain the average performance of the models to find the optimal hyper-parameter values of the base classifiers. Therefore, prediction probabilities made by these models on the OOB (Out-of-Bag) samples (i.e. the test folds in Figure 3), along with the true label of OOB samples, form the Stacking features.

However, things may be different when dealing with the financial problems for its time series characteristics. The most severe drawback of cross-validation method is the leakage of future information. Reviewing this method, training and test set are randomly splited into k folds without replacement. This means the model may be trained by the future data and predict on the OOB samples which belong to the past information. Since model performance will always be great if we use the future information to predict the past, cross-validation method may cause serious over-fitting problems in this situation.

Hence, we implement a method called timeslice grid search to address this drawback. Specifically, for each iteration, base classifiers are trained on the training subset (please see Figure 3), and performance estimations are then calculated on the subsequent test subset[1]. This chronological nature is kept during the whole process, and the selection of optimal hyper-parameters and Stacking features are further generated simultaneously based on the OOB samples (i.e. the test subset). All the processes are shown in the Figure 3.

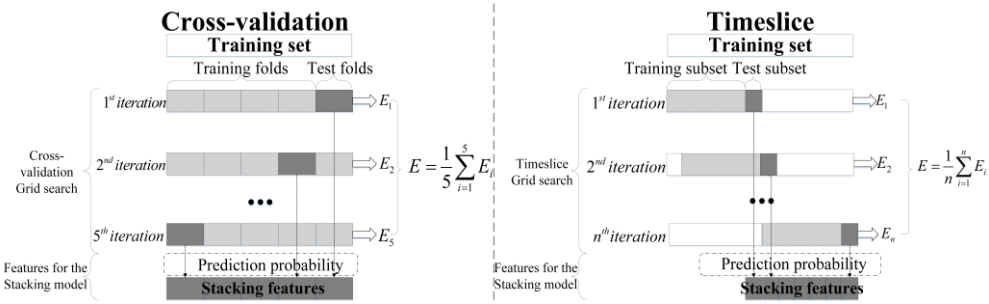

Figure 3. The cross-validation and timeslice grid search method

4.4. Stacking (Stacked generalization)

In this study, we modify the basic Stacking framework to rolling predict the

[1] Our training subset consists of 240 days (approximately one year) compared to the training set (1200 days, approximately five years), in order to increase the number of interactions for finding the suitable hyper-parameter values. The length of test subset is same as the test set which contains 60 days. Accuracy is used as the metric to select the optimal hyper-parameters.

DOI: 10.24818/18423264/53.3.19.11

direction of the stock price indices. Our improved Stacking method can be further summarized as three layers and TSGS method aforementioned is used to select optimal hyper-parameters. In the first layer, five base models which can effectively span the learning space (please see the Figure 2), namely SVM, NN, ENLG, RF and XGBoost, are developed to extract the information from the original data.

In the second layer, three of the machine learning algorithms called ENLR, RF and XGBoost are deployed as meta classifiers to implement the Stacking steps, in order to fuse the information from models and original data by using features which are combined with prediction probabilities of base classifiers and lag returns. Herein, we use these two types of data since prediction probabilities contain more information compared to the prediction classes, and original data may not be fully utilized by the base classifiers. Moreover, the design of the second layer enables us to:1. We are able to figure out which source of information is more useful in predicting the stock price index, since ENLR can regularize the estimates and a model-specific variable importance metric is available for RF and XGBoost; 2. The meta classifiers deployed aforementioned have the ability to reject all the decision from the base models (i.e. one ticket veto) when ENLR shrink all the coefficients of models to zero or variable importance com to zero in RF and XGBoost; 3. Cause the same algorithms (ENLR, RF and XGBoost) are employed in both base classifiers and meta classifiers, we can investigate whether base classifiers have extracted most of the information in the original data. For example, it indicates that most information of the original data have been learned by the base classifiers, when the variable importance value of lag returns is much lower in the meta classifier compared to the base classifiers.

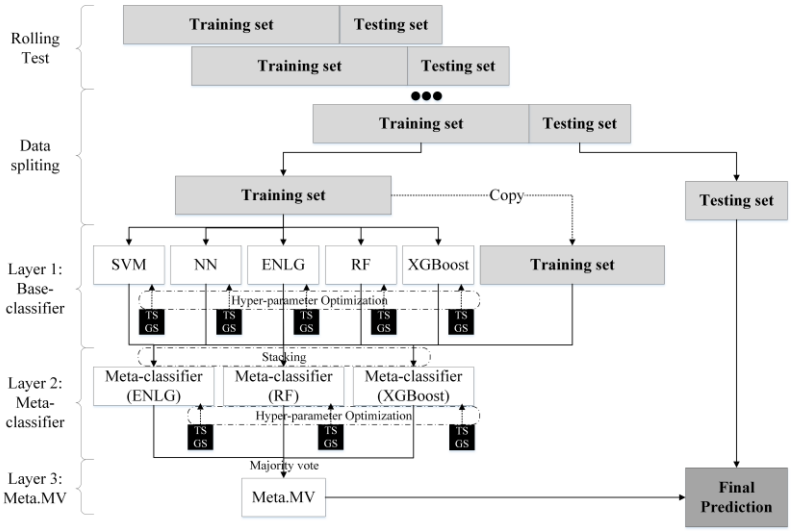

Figure 4. The framework of the improved Stacking method

Finally, we deploy majority vote method to further reduce the error rate and ensemble these three low-correlated meta classifiers in the third layer (namely

DOI: 10.24818/18423264/53.3.19.11

Meta.MV). The above experimental procedure is illustrated in Figure 4.

5. Result

5.1. Prediction performance

In this subsection, we compare the average prediction performance of the models previously discussed in Section 0. For demonstrating the efficacy and consistency of our approach, this is a general result implying a trend of performance across various indices and is not pertaining to the only one stock price index.

Figure 5 depicts the trend of prediction accuracy, kappa statistic, F-score and AUC value against models that belong to different layers. In general, the prediction performance of models improves as the layer rises due to the increased information content of model fusion. What's more, the Meta.MV model belonging to the last layer surpasses the existing theoretical models in almost every respect (except for F-score slightly lower than RF), and can effectively improve the stability of prediction performance, since the optimal meta classifier is not always the same.

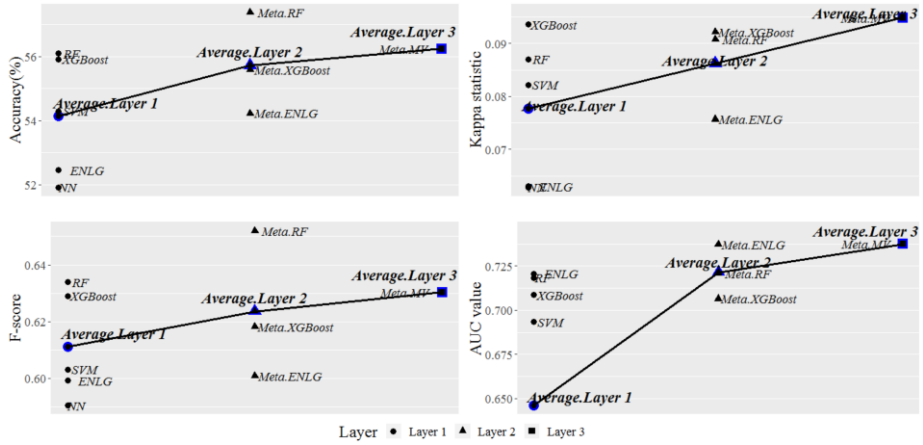

Figure 5. The trend of prediction performance against different layers

Table 2. The prediction performance of different models

Model	accuracy(%)	sensitivity(%)	specificity(%)	kappa	F-score	AUC
SVM	54.2938	67.6581	40.8381	0.0822	0.603	0.6934
NN	51.9115	63.3312	44.698	0.0628	0.5903	0.3887
ENLG	52.4482	66.5983	40.4388	0.0631	0.5991	0.7207
RF	56.0923	73.5483	36.6283	0.087	0.6339	0.7183
XGBoost	55.9134	70.3671	41.2745	0.0936	0.629	0.7087
Meta.MV	56.2429	71.8416	39.312	0.0948	0.6303	0.7374

Table 2 demonstrates the average prediction performance of the three major stock price indices in U.S. (S&P500, NASDAQ and Dow Jones) from Oct 2004 until Nov 2018, since we lose the first 1200 days of sample due to model training.Measurement used for evaluating the robustness of binary classifiers are

DOI: 10.24818/18423264/53.3.19.11

accuracy, sensitivity, specificity, kappa statistic, F-score and AUC value, which is described in detail in previous sections. The results of models pertaining to Layer 2 are omitted for simplicity. Across the entire output domain (refer toTable 2), Meta.MV, which ensembles three meta classifiers namely Meta.ENLR, Meta.RF and Meta.XGBoost provide better performance than the single classifiers or even the state-of-the-art ensemble classifiers (i.e. RF and XGBoost) in terms of accuracy, kappa statistic and AUC value.

In addition, all of these classification models perform more accurately for predicting the increase situation compared to the decrease situation in terms of sensitivity and specificity. This is a significant negative impact on model fitting due to the disparity in the frequencies of the observed classes (i.e. class imbalance) which is demonstrated in Table 1.

Neural Networks generally perform worse than other machine learning methods. This may attribute to the simplicity of its structure since we only use one hidden layer in our study. Furthermore, the result implies that Stacking method still has potential for the improvement of prediction performance.

5.2. Further analyses

With all three meta classifiers including Meta.ENLR, Meta.RF and Meta.XGBoost, we can extract variable frequency and variable importance of the models, i.e., the relative predictive strength of each feature. For Meta.ENLR, variable frequency can be calculated since the regularized terms are able to shrink the coefficients to zero. For Meta.RF and Meta.XGBoost which belong to the tree-based methods, variable importance is determined by computing the relative influence of each variable, i.e., by assessing whether a particular variable is used during splitting when growing trees, and by how much the loss function improves as a result on average across all trees. Furthermore, we denote the variable frequency in percent and normalize the most important variable to an index value of 100. Average variable frequency and relative importance across all the stock price indices are depicted in Figure 6, Figure7 and Figure 8. $Return.m$ refers to the multi-period return calculated over m periods, as in (1).

Some meaningful conclusions can be drawn from the figures above. Above all, the multi-period returns can be split into two groups, i.e., the short-term features which are less than one month ($Return.1 \sim Return.20$) and long-term features which are over one month ($Return.40, \ldots, Return.240$). Specifically, long-term features are under a higher frequency and relative importance in contrast to short-term features. This result becomes more significant in the models based on Random Forest algorithm (i.e. RF and Meta.RF) and models based on Extreme Gradient Boosting algorithm(i.e. XGBoost and Meta.XGBoost), with relative importance sharply dropping after $Return.20$. Specially, the ranking of short-term features of RF and Meta.RF seems sequential and not arbitrary. In general, the frequency and relative importance of multi-period returns can be found to decrease (although they're still greater than zero), when we combine the lag returns and prediction probabilities of

DOI: 10.24818/18423264/53.3.19.11

models as Stacking features in meta classifiers (i.e. Meta.ENLR, Meta.RF and Meta.XGBoost).

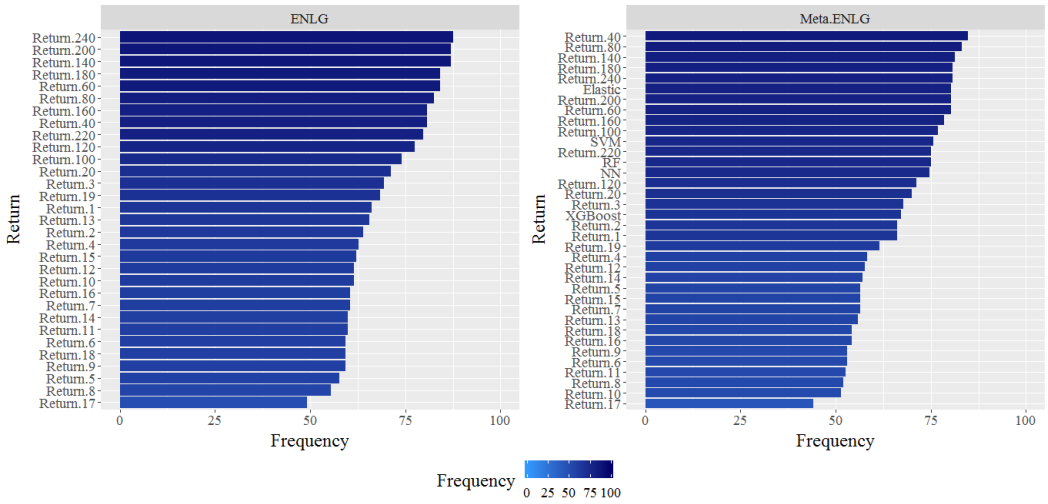

Figure 6. Variable frequency extracted from ENLG and Meta.ENLG

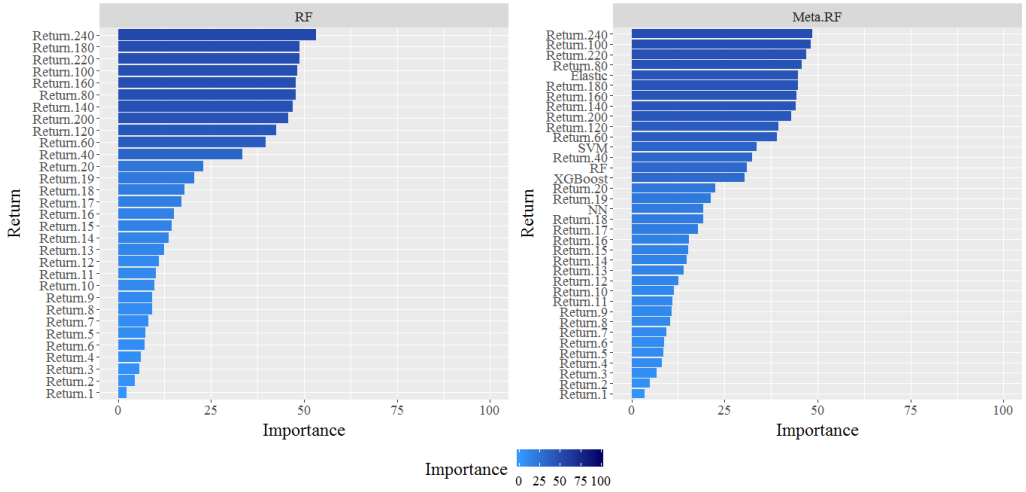

Figure7. Variable relative importance extracted from RF and Meta.RF

DOI: 10.24818/18423264/53.3.19.11

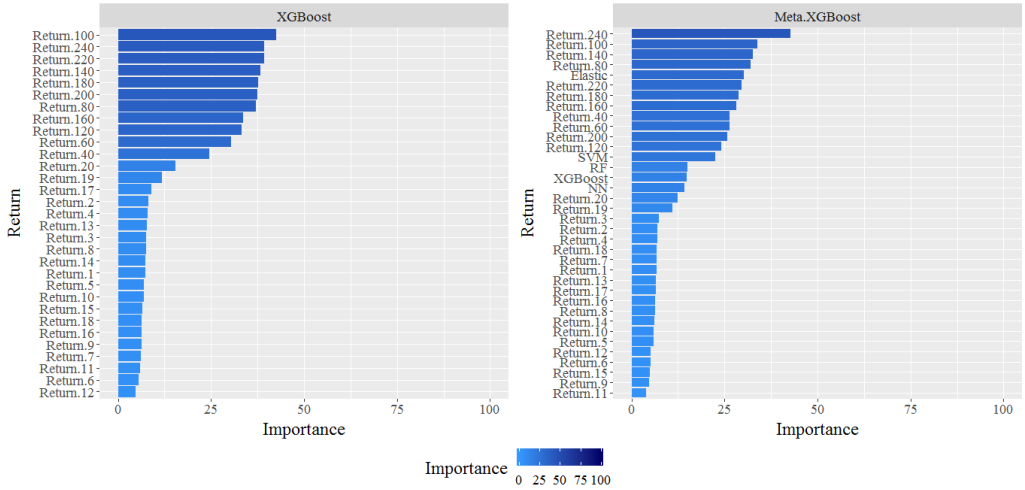

Figure 8. Variable relative importance extracted from XGBoost and Meta.XGBoost

Subsuming all the results listed above, the following logic seems to apply: the long-term features corresponding to the lag returns over 20 days have greater explanatory power and prediction performance, as confirmed by all models. This result indicates that stock price indices are mainly driven by the long-tem momentum effect from a monthly perspective. Moreover, some machine learning models have the ability to extract some part of valuable information from the historical data of stock price indices and construct an enhanced version of the momentum effect, since the frequency and relative importance of variables which represents base classifiers are bigger than zero (even large in some cases). Besides, information which has not been fully utilized by base classifiers is further used in the meta classifier, due to the consequences of decreased and nonzero frequency (relative importance) of multi-period returns in the meta classifiers.

6. Conclusion and future research

Being able to predict the stock price index direction more accurately is beneficial to investors in hedging against potential mark risks, speculators and arbitrageurs specialized in dealing stock price index, or even countries, since market index is an indication of future economic performance in national layer. This study set out to implement an improved multi-layer Stacking method in predicting the stock price index direction. To the best of our knowledge, this is the first study to include Stacking algorithm in the prediction application of stock price index direction.

To be more specific, multi-period lag returns are used as features to explore the momentum effect and predict the direction of the three major stock price indices in U.S. namely S&P500, NASDAQ and Dow Jones, from Jan 2000 until Nov 2018. Five machine learning models called SVM, NN, ENLR, RF and XGBoost are

DOI: 10.24818/18423264/53.3.19.11

implemented in the first layer of our improved Stacking framework, which are built by various learning algorithms and can effectively span the learning space. Considering the time series characteristic of the financial data, we further deploy a timeslice grid search method instead of cross-validation, to find the optimal hyper-parameters of the models and generate Stacking features simultaneously. ENLR, RF and XGBoost are developed as meta classifiers in the second layer, to fuse the information which respectively pertains to base classifiers and original data. Finally, majority vote is used to ensemble these three meta classifiers in the last layer.

In our study, we found that Meta.MV model which combines three meta classifiers is the top performer. Specifically, it provides better and more stable prediction performance compared to the single classifiers such as SVM, NN and ENLR, or even state-of-the-art machine learning algorithms RF and XGBoost. This is a significant conclusion since RF and XGBoost have already been widely deployed in the areas of AI or data science competitions and proven to be the best. Furthermore, the results illustrate that the prediction performance of models improves as the layer rises due to the increased information content of model fusion. Hence, we suggest that novel studies in the domain of stock price or stock price index direction prediction should consider the framework of Stacking method.

A future research topic may be to investigate and affiliate a wide range of machine learning models which are not overlapped to improve prediction accuracy. Besides, some macroeconomic features should be considered in our framework since multi-period returns are not the only source of information which may also be hidden in macroeconomic variables.

REFERENCES

[1] **Ballings, M., Van den Poel, D., Hespeels, N.& Gryp, R. (2015)**,***Evaluating Multiple Classifiers for Stock Price Direction Prediction***. *Expert Systems with Applications,* 42(20), 7046-7056;

[2] **Basak, S., Kar, S., Saha, S., Khaidem, L.& Dey, S. R. (2018)**,***Predicting the Direction of Stock Market Prices Using Tree-based Classifiers***. *The North American Journal of Economics and Finance*;

[3] **Breiman, L. (1996)**,***Bagging Predictors***. *Machine Learning,* 24(2), 123-140;

[4] **Breiman, L. (2001)**,***Random Forests***. *Machine Learning,* 45(1), 5-32;

[5] **Chen, T.& Guestrin, C. (2016)**,***XGBoost: A Scalable Tree Boosting System.*** In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794);

[6] **Cheung, Y. W., Chinn, M. D.& Pascual, A. G. (2005)**,***Empirical Exchange Rate Models of the Nineties: Are any Fit to Survive?****Journal of International Money & Finance,* 24(7), 1150-1175;

[7] **Dreiseitl, S.& Ohno-Machado, L. (2002)**,***Logistic Regression and Artificial Neural Network Classification Models: A Methodology Review***. *Journal of*

DOI: 10.24818/18423264/53.3.19.11

Biomedical Informatics, 35(5), 352-359;
[8] **Enke, D.& Thawornwong, S. (2005)**,***The Use of Data Mining and Neural Networks for Forecasting Stock Market Returns***.*Expert Systems with Applications,* 29(4), 927-940;
[9] **Freund, Y. & Schapire, R. E. (1996),*Experiments with a New Boosting Algorithm.*** In: *International Conference on Machine Learning* (pp. 148-156);
[10] **Haykin, S. (2002)**,***Adaptive Filter Theory, 5e***.
[11] **Krauss, C., Do, X. A.& Huck, N. (2017)**,***Deep Neural Networks, Gradient-Boosted Trees, Random Forests: Statistical Arbitrage on the S&P 500***. *European Journal of Operational Research,* 259(2), 689-702;
[12] **Kumar, M.& M, T. (2006)**,***Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest***. *Social Science Electronic Publishing*;
[13] **Lantz, B. (2015)**,***Machine Learning with R: Second edition***;
[14] **Leung, M. T., Daouk, H.& Chen, A.-S. (2000)**,***Forecasting Stock Indices: A Comparison of Classification and Level Estimation Models***. *International Journal of Forecasting,* 16(2), 173-190;
[15] **Moein Aldin, M., Dehghan Dehnavi, H.& Entezari, S. (2012)**,***Evaluating the Employment of Technical Indicators in Predicting Stock Price Index Variations Using Artificial Neural Networks (Case Study: Tehran Stock Exchange)***. *International Journal of Business and Management,* 7(15);
[16] **Pesaran, M. H.& Timmermann, A. (1995)**,***Predictability of Stock Returns: Robustness and Economic Significance***. *Journal of Finance,* 50(4), 1201-1228;
[17] **Takeuchi, L.& Lee, Y.-Y. (2013)**,***Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks***. *Working Paper*;
[18] **Tsai, C.-F., Lin, Y.-C., Yen, D. C.& Chen, Y.-M. (2011)**,***Predicting Stock Returns by Classifier Ensembles***. *Applied Soft Computing,* 11(2), 2452-2459;
[19] **Vapnik, V. N. (1999)**,***An Overview of Statistical Learning Theory***. *IEEE Trans Neural Netw,* 10(5), 988-999;
[20] **Weng, B., Martinez, W., Tsai, Y.-T., Li, C., Lu, L., Barth, J. R.& Megahed, F. M. (2018)**,***Macroeconomic Indicators Alone Can Predict the Monthly Closing Price of Major U.S. Indices: Insights from Artificial Intelligence, Time-Series Analysis and Hybrid Models***. *Applied Soft Computing,* 71, 685-697;
[21] **Wolpert, D. H. (1992)**,***Stacked Generalization***. *Neural Networks*(5), 241-259;
[22] **Zou, H.& Hastie, T. (2005)**,***Regularization and Variable Selection via the Elastic Net***. *Journal of the Royal Statistical Society. Series B (Statistical Methodology),* 67(2), 301-320.

DOI: 10.24818/18423264/53.3.19.11