Quantitative Finance

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/rquf20

Stock market prediction based on adaptive training algorithm in machine learning

Hongjoong Kim, Sookyung Jun & Kyoung-Sook Moon

To cite this article: Hongjoong Kim, Sookyung Jun & Kyoung-Sook Moon (2022) Stock market prediction based on adaptive training algorithm in machine learning, Quantitative Finance, 22:6, 1133-1152, DOI: 10.1080/14697688.2022.2041208

To link to this article: https://doi.org/10.1080/14697688.2022.2041208

Published online: 11 Mar 2022.

Quantitative Finance, 2022
Vol. 22, No. 6, 1133–1152, https://doi.org/10.1080/14697688.2022.2041208

Stock market prediction based on adaptive training algorithm in machine learning

HONGJOONG KIM †, SOOKYUNG JUN† and KYOUNG-SOOK MOON *‡

†Department of Mathematics, Korea University, 145 Anam-Ro, Seongbuk-Gu, Seoul 02841, Korea
‡Department of Mathematical Finance, Gachon University, 1342 Seongnam-Daero, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do 13120, Korea

(*Received 25 February 2020; accepted 4 February 2022; published online 11 March 2022*)

This study deals with one of the most important issues for understanding financial markets, future asset fluctuations. Predicting the direction of asset fluctuations accurately is very difficult due to the uncertainty of the stock market, the influence of various economic indicators, and the sentiment of investors, etc. In this study, we present a new method to improve the effectiveness of machine learning by selecting appropriate training data using an adaptive method. The application to various sector data of the S&P 500 and many machine learning methods shows that the proposed adaptive data selection algorithm improves the prediction accuracy of the stock price direction. In addition, it can be seen that the adaptive data selection method increases the return on the asset investment.

Keywords: Adaptive data construction; Empirical validation; Financial forecasting; Machine learning

JEL Classifications: C53, C63, C18

1. Introduction

Machine learning (ML) techniques have been widely and successfully utilized in the prediction of the data in many application fields, including the financial markets. Traditionally, the efficient market hypothesis in finance leads to the belief that the returns from speculative assets are unforecastable and it is difficult to outperform the overall market. However, many recent works identify and successfully exploit short-term inefficiencies of financial markets, see Jensen (1978) and Timmermann and Granger (2004). A better understanding of the financial market and the prediction with higher accuracy may offer more profits or returns to investors, but this is a challenging task due to volatile non-parametric random financial data and various economic factors affecting the stock markets, see Campbell *et al.* (1997).

There have been various studies to accurately analyze the financial variables, such as stock market prices, stock indices or currency exchange rates. Both statistical learning and artificial intelligence methods and even their hybridization have been developed to predict the trend of stock or stock index. Statistical learning methods include single classifiers such as logistic regression (LR) (Harrell 2015), decision tree (DT) (Lai *et al.* 2009, Rokach and Maimon 2014), k-nearest neighbors (KNN) (Teixeira and Oliveira 2010, Biau and Devroye 2015), support vector machine (SVM) (Huang *et al.* 2005, Vapnik 2013, Law and Shawe-Taylor 2017), see more methods and references in Atsalakis and Valavanis (2009), Friedman *et al.* (2013) and James *et al.* (2013). Ensemble methods such as Majority Voting (MV), XGboost and Random Forest (RF) use multiple learning classifiers to obtain more reliable performance than single classifiers as in Ballings *et al.* (2015), Scholz *et al.* (2015), Moon *et al.* (2018) and Huang *et al.* (2020). Artificial intelligence models such as artificial neural networks, fuzzy systems, and genetic algorithms are also developed for financial stock market predictions in Kaastra and Boyd (1996), Hadavandi *et al.* (2010), Schoneburg (1990) and Zhong and Enke (2017). Recently, deep learning models, such as Long-Short-Term-Memory are applied for forecasting the stock markets, see Fischer and Krauss (2018) and Chen and Ge (2019).

But, these machine learning techniques have a weakness that even though they work well on the training data, they may not have satisfactory results for the prediction of *unseen* test data due to the complex and non-stationary property of the stock market. To solve such deficiency, two-stage architectures or hybridizations with other existing methods have been developed. Hsu and Hsu (2009) combine the self-organizing map and Support Vector Regression (SVR) and empirically

*Corresponding author. Email: ksmoon@gachon.ac.kr

Table 1. A summary of recent studies on stock market prediction.

Authors (Year)	Data type (Period)	Method
Ballings *et al.* (2015)	5767 European stocks (Jan, 2009–Dec, 2014)	LR, NN, KNN SVM AB, RF, KF
Patel *et al.* (2015)	India CNX and BSE (Jan, 2003–Dec, 2012)	1st: SVR 2nd: ANN, RF, SVR
Oztekin *et al.* (2016)	Turkey BIST 100 index (Jan, 2007–Dec, 2014)	1st: cross validation 2nd: SVM, ANN, fuzzy
Qiu *et al.* (2016)	Nikkei 225 index (Jan, 2008–Jul, 2013)	1st: ANN 2nd: GA, SA
Zhong and Enke (2017)	S&P 500 index (Jun, 2003–May, 2013)	1st: clustering, dimension reduction 2nd: ANN, LR
Chen and Hao (2017)	China SSE and SZSE (Oct, 2008–Dec, 2014)	1st: SVM 2nd: KNN
Our study	55 stocks in 11 sectors of S&P 500 (Oct, 2010–Oct, 2020)	1st: ADS, cluster, recent, all 2nd: LR, DT, KNN SVM, RF, MV

LR: Logistic Regression, DT: Decision Tree, KNN: k-Nearest Neighbors, SVM: Support Vector Machine, MV: Majority Voting, RF: Random Forest, AB: AdaBoost, KF: Kernel Factory, ANN: Artificial Neural Network, GA: Genetic Algorithm, SA: Simulated Annealing, ADS: Adaptive Data Selection.

show that the two-stage method improves the performance in stock price prediction. Qiu *et al.* (2016) apply the artificial neural network model together with genetic algorithm and simulated annealing to improve the prediction accuracy of the Japanese Nikkei 225 index. Pai and Lin (2005) propose a hybridization of the Auto Regressive Integrated Moving Average (ARIMA) and Support Vector Machine (SVM) and illustrate that the hybrid model forecasts better than single SVM or ARIMA. Patel *et al.* (2015) propose a two-stage fusion approach involving SVR in the first stage and ANN, RF, and SVR for the second stage in predicting Indian stock market. Chen and Hao (2017) propose a hybridized framework of the feature weighted support vector and k-nearest neighbors for the prediction of Chinese stock market indices, and Oztekin *et al.* (2016) predict stock price movements based on fuzzy system, artificial neural networks and support vector machines. Table 1 summarizes those hybridizations.

The standard prediction procedure in machine learning is as follows: (1) Construction of training data; (2) Selection of machine learning classifier; (3) Training the classifier to fit the training data; (4) Application of the prediction model to the test data, see Michie *et al.* (1994), Friedman *et al.* (2013) and Raschka (2015). Many studies try to improve the prediction by mostly focusing on the modification of classifiers in machine learning. Even though classifiers are an important part, they are not the only component of the prediction algorithm. Note that the prediction results not only depend on the machine learning classifier used but are also influenced by the data sets as shown in table 1. Thus, the current study shows that one may achieve enhancements by improving the way how those classifiers are trained in machine learning.

Let us consider the following basic question for the procedure above: how can *proper* training data be constructed? It is known that training of machine learning requires sufficiently many data to find optimal parameters or hyperparameters. But, not many references propose a suitable or appropriate way to construct the training data or distinguish their importances. The usage of all the available data for training does not always improve the accuracy because the characteristics of the market may change in time and the usage of the whole data may include out-of-date data. In addition, the computational cost increases as the amount of data increases, which may lower the efficiency of the prediction. So, some studies even suggest using a *subset* of the whole data for training, for instance, training with recent data only. Even though the usage of recent data may capture up-to-date trends in the market, it may be biased depending on the amount of data. Other studies suggest to use unsupervised learning to split the whole data into several clusters then select a suitable cluster for training, see Lai *et al.* (2009) and Aggarwal and Reddy (2013). But, the selected cluster may not fit the test data and the control of unsupervised learning for grouping is quite limited.

We propose an adaptive method called the *Adaptive Data Selection* (ADS) to construct an appropriate training data, which is explained in detail in section 2.2 and practically modified as *Inductive ADS* in section 2.3. In fact, when a future value is predicted at a certain point in time, the feature data up to *that point* are available at the moment of training even though the corresponding labels are not available. Since the feature data are still accessible, those can be compared with those of each data in the entire data set available. Then, only data having mathematically close similarity will be selected to be used as the *actual* training data. We study the effects of the ways how the training data are constructed on the prediction accuracy of various stock prices. The merits of the proposed ADS method are as follows:

(i) For an accuracy point of view, machine learning methods such as LR, DT, KNN, SVM, MV, RF combined with ADS outperform those without ADS in two aspects. Firstly, the methods with ADS improve the *mea*n prediction accuracy when tested with various methods or data, see table 9 and figure 12 in section 4. Secondly, the worst case prediction accuracies with the proposed ADS method are more accurate than

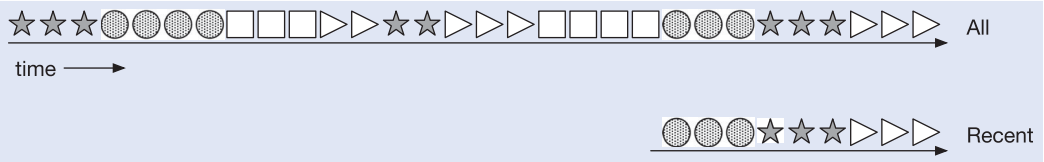

Figure 1. All- and recent-type construction of the training data. Each symbol represents a pattern of the financial data on each day and different symbols represent different patterns.

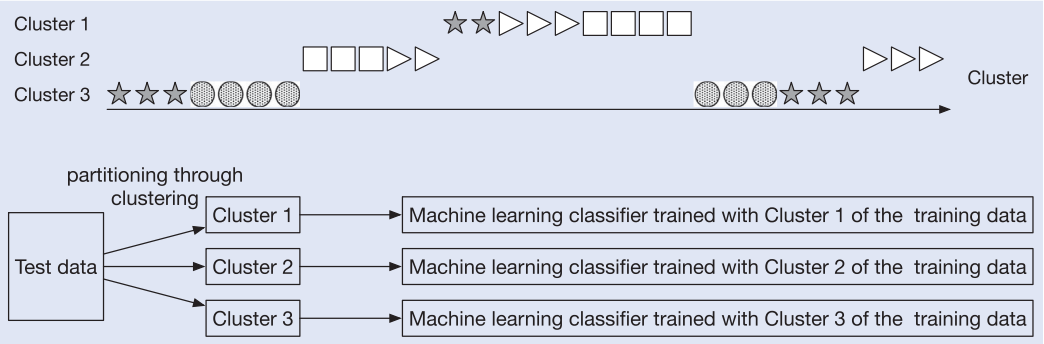

Figure 2. (Top) Cluster-type construction of the training data. (Bottom) Cluster-type prediction of the test data.

those without ADS, which is an important indicator for credibility. See figures 15 and 16 in section 4.

(ii) As can be seen in the figure 17 in section 4.2, the high prediction accuracy from the ADS method leads to high returns on the asset allocation problems.

(iii) The proposed ADS training data selection can be easily combined with various machine learning algorithms and it can be applied to not only financial data such as stock prices, stock indices or currency exchange rates but also other types such as images, audio or biomedical data.

(iv) Inductive ADS first forms a group of mathematically similar data, and then learns by finding a group similar to the data to be tested. Therefore, it can be said to act as a bridge between unsupervised learning and supervised learning for prediction. The detailed algorithm is summarized in section 2.

The rest of the paper is organized as follows. Section 2 describes the standard construction method of training data and proposes the adaptive data selection method. Then, it explains how to combine the ADS with other machine learning algorithms. The statistics of the data, brief descriptions of machine learning methods and accuracy measures are studied in section 3. Experimental results are presented and discussed in section 4 and conclusions are drawn in section 5.

2. Training data selection

2.1. *Data construction methods for training data*

Machine learning requires sufficiently large amount of data for training. Given all the known historical stock values, the whole data can be used as the training data so that parameters are tuned to fit the entire dataset, or only a subset of the whole data can be used as the training data. When the entire data set available is used as the training data as in figure 1 (Top), such a construction method will be denoted by *All* in this study. Each symbol in figure 1 represents a pattern of the financial data on each day and different symbols represent different patterns. *All*-type training data may include out-of-date data and the resultant training may not capture the recent trend properly. In order to improve the efficiency, one may use only a subset of the entire data. For instance, the training data can be constructed in terms of the proximity in time. When a prediction is to be made at a certain point in time, only recent stock values in the historical values can be used as the training data as in figure 1 (Bottom). Such a construction method will be called *Recent* in this study.

Alternatively, *Cluster* uses unsupervised K-means clustering algorithm to partition *All*-type training data into n *disjoint* groups as in figure 2 (Top) and several machine learning models are constructed from one machine learning classifier, one for each group. The test data are partitioned to n groups by the same K-means algorithm and then n groups of the training data are paired with n groups of the test data with respect to the distance between centroids of the groups. For the prediction of the test data in each group, the only data in the paired cluster are used as the training data of the machine learning classifier as in figure 2 (Bottom). The usage of the data only in that chosen cluster as the training data may improve the computational speed and prediction accuracy, see Lai *et al.* (2009) and Raschka (2015). Such a construction method will be called *Cluster* method in this study. However, it is very limited to control the way how the clusters are grouped in unsupervised learning. For instance, the number of cluster k is hard to choose and the result is sensitive to initial points. In particular, a data point can exist only one unique cluster. Thus, the data within each cluster may not have similar patterns or the selected cluster may have weak similarities with the given test data, in which case the accuracy may not be ameliorated.

In order to improve the accuracy, a method called the *Adaptive Data Selection* (ADS) is proposed to construct an

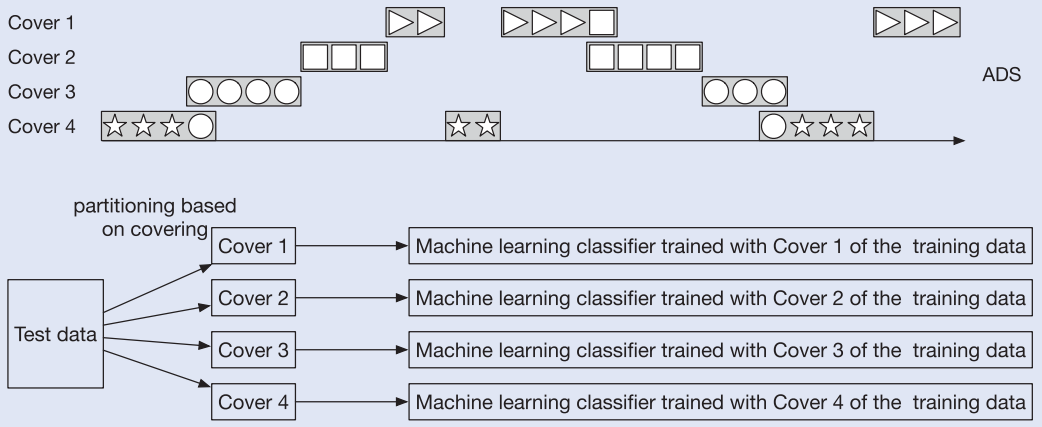

Figure 3. (Top) ADS-type construction of the training data. (Bottom) ADS-type prediction of the test data.

appropriate training data as explained in sections 2.2 and 2.3. ADS is an algorithm that collects and learns only training data of a shape similar to the current data trend as in figure 3. It can be confirmed with a variety of data that it shows good prediction results because it is trained only with similar types of data. However, as in the case of ADS, there is a difficulty in that the calculations increase every time to find training data similar to the current trend. Therefore, we divide the training data into groups of similar shapes in advance and learn with the training data of the group most similar to the current trend. Let us call this modified algorithm as Inductive ADS. Inductive ADS introduces a *cover* $\cup_k U_k$, i.e. a collection of sets whose union includes the whole data set. The subsets U_k's need not be disjoint so that the weakness of clustering near boundaries are overcome. In addition, since the distance in L_2 sense is considered, the data within each subset have similar patterns.

Given a machine learning classifier such as SVM, *All* and *Recent* construct one SVM model from the training data, which is applied to the whole test data. On the other hand, *Cluster* and *ADS* construct *several* SVM models and the prediction at each data in the test data is made with one and only one SVM model. Table 2 summarize 4 construction methods for the training data considered in this study. Note that even though *Recent, Cluster* and *ADS* methods selectively choose a subset from the entire data, *Recent* measures the temporal proximity, *Cluster* performs unsupervised K-means clustering and *ADS* computes the mathematical similarity to define the subset as explained below.

Table 2. Four construction methods for the training data.

Construction method	Amount of data used out of the whole data	Data selection method
All	Entire set	–
Recent	Subset	Proximity in time
Cluster	Subset	Unsupervised clustering
ADS	Subset	Mathematical similarity

2.2. *Adaptive data selection*

Suppose that historical values in finance denoted by $\{S_k\}$ are given (a dotted curve in figure 4 (Left)) and that some values are reserved for the tests (the right side of the vertical line in figure 4 (Left)). Suppose that one wants to predict whether the financial value increases or decreases in p days from $t = t_i$ (marked by the circle in figure 4 (Left)) given the values up to S_i. The proposed *ADS* method considers all the known historical stock values available (i.e. those on the left side of the

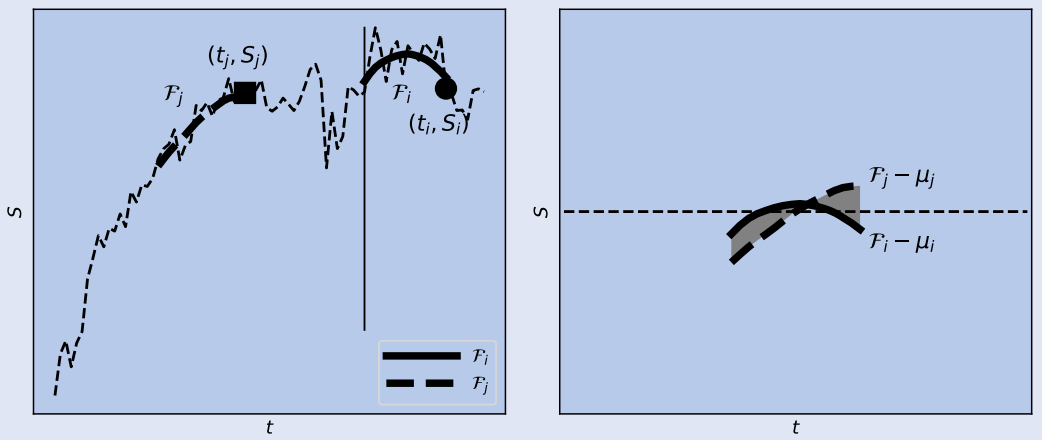

Figure 4. (Left) The features (solid curve) of the test data (circle) and features (dashed curve) of a candidate data (square) from the known historical values (whole dotted curve). (Right) The area of the shaded region between the curves in L_2 norm in (3) is used as a measure for distance.

vertical dotted line in figure 4 (Left)) to select appropriate data as follows. The feature of S_i is compared with that of each point in the historical values for similarity in trends. Note that the raw values (a dotted curve) show oscillations in time and these oscillatory fluctuations may give misleading information when the trends in data patterns are compared. In order to eliminate the noise from such an irregular rising and falling of financial values, the values are smoothened first using w-day moving average:

$$\overline{S}_k = \frac{1}{\mathsf{w}} \sum_{j=0}^{\mathsf{w}-1} S_{k-j}. \tag{1}$$

Then those smoothened values of past N days with the current average $\overline{S}_i$ are used as the feature at t_i. That is, the test data S_i at t_i has the feature $\mathcal{F}_i = \{\overline{S}_{i-\mathsf{N}}, \overline{S}_{i-\mathsf{N}+1} \ldots, \overline{S}_i\}$ (a solid curve in figure 4 (Left)), which will be compared with the feature $\mathcal{F}_j = \{\overline{S}_{j-\mathsf{N}}, \overline{S}_{j-\mathsf{N}+1} \ldots, \overline{S}_j\}$ (a dashed curve in figure 4 (Left)) of S_j for each j.

In order to compare relative trends only and to reduce any other financial or monetary differences in time, those two features are standardized to have zero means:

$$\hat{\mathcal{F}}_i \equiv \mathcal{F}_i - \mu_i = \{\overline{S}_{i-\mathsf{N}} - \mu_i, \ldots, \overline{S}_i - \mu_i\} \tag{2}$$

where μ_i denotes the mean of $\{\overline{S}_{i-\mathsf{N}}, \ldots, \overline{S}_i\}$. That is, the distance between the standardized features $\hat{\mathcal{F}}_i$ and $\hat{\mathcal{F}}_j$ of S_i and S_j, respectively, is measured by L_2 norm

$$d_{ij} = \|\hat{\mathcal{F}}_i - \hat{\mathcal{F}}_j\|_2 = \sqrt{\sum_{k=0}^{\mathsf{N}} \left\{(\overline{S}_{i-k} - \mu_i) - (\overline{S}_{j-k} - \mu_j)\right\}^2}, \tag{3}$$

where μ_i and μ_j are the means of $\{\overline{S}_{i-\mathsf{N}}, \ldots, \overline{S}_i\}$ and $\{\overline{S}_{j-\mathsf{N}}, \ldots, \overline{S}_j\}$, respectively. The value d_{ij} in (3) approximates the area of the shaded region in figure 4 (Right). Here $S_k = S(t_k)$ is a scalar value representing the close price of a stock at time $t = t_k$. However one can directly extend it to a vector, for instance, $\boldsymbol{S}_k = (O_k, H_k, L_k, C_k)^T$ with the open price O_k, high H_k, low L_k, and close price C_k, etc. Then, the two features $\mathcal{F}_i = \{\overline{\boldsymbol{S}}_{i-\mathsf{N}}, \ldots, \overline{\boldsymbol{S}}_i\}$ and $\mathcal{F}_j = \{\overline{\boldsymbol{S}}_{j-\mathsf{N}}, \ldots, \overline{\boldsymbol{S}}_j\}$ become matrices, and each element is a vector of size M. That is, $\overline{\boldsymbol{S}}_k = (\overline{S}_{1,k}, \overline{S}_{2,k}, \ldots, \overline{S}_{\mathsf{M},k})^T$. Then the distance between standardized feature matrices can be measured by *Frobenius* norm

$$d_{ij} = \|\hat{\mathcal{F}}_i - \hat{\mathcal{F}}_j\|_f = \sqrt{\sum_{l=1}^{\mathsf{M}} \sum_{k=0}^{\mathsf{N}} \left\{(\overline{S}_{l,i-k} - \mu_{l,i}) - (\overline{S}_{l,j-k} - \mu_{l,j})\right\}^2},$$

where $\mu_{l,i}$ and $\mu_{l,j}$ are the means of $\{\overline{S}_{l,i-\mathsf{N}}, \ldots, \overline{S}_{l,i}\}$ and $\{\overline{S}_{l,j-\mathsf{N}}, \ldots, \overline{S}_{l,j}\}$, respectively, for $l = 1, 2, \ldots, \mathsf{M}$.

When d_{ij} is smaller than a predefined tolerance, TOL, the proposed Adaptive Data Selection method adds the corresponding candidate data S_j into the training data. Otherwise, the data are discarded and are not included in the training data. The procedure above is repeated for the next available data in the remaining known values until all the available data in the entire set are checked for inclusion as in figure 5. For instance, figure 6 shows the standardized d_{ij} values for Comcast Corporation (CMCSA) in the Communication Services sector of S&P 500 index for 10 years from October 22, 2010 to October 23, 2020, in which the dotted line represents the threshold value of TOL = 1.6 to include 500 small values. The prediction algorithm based on the training data from the ADS construction method can be summarized as in algorithm 1.

2.3. *Inductive ADS*

While the ADS algorithm in section 2.2 has the advantage in training by collecting only similar data, it repeatedly performs collecting appropriate data and training them whenever new

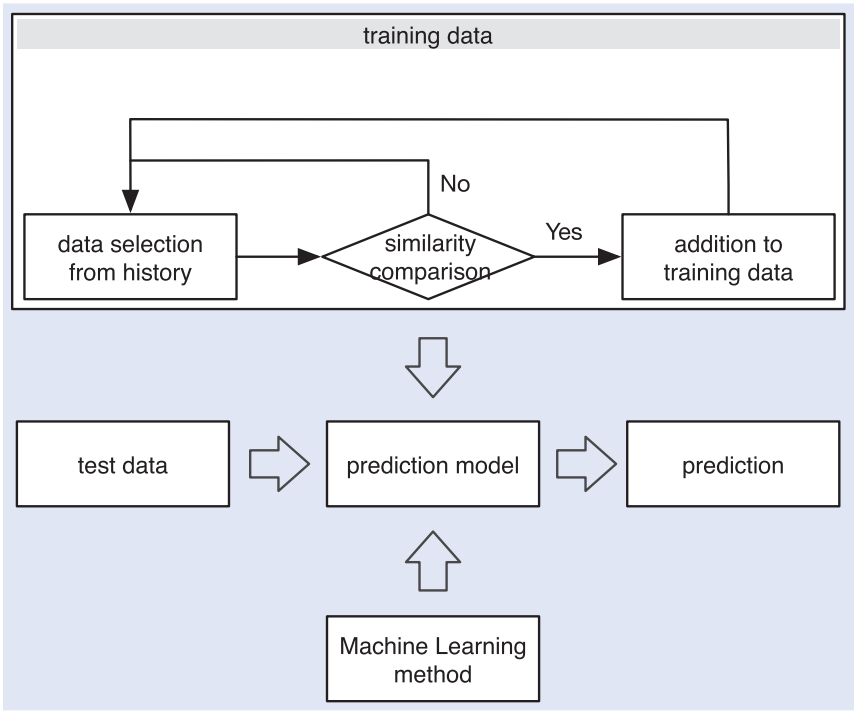

Figure 5. The simple structure of the ADS method.

Algorithm 1 ADS Algorithm

Input: Historical stock prices and parameters w in (1), N in (2) and TOL

Output: Future trend for the test data based on ADS construction method

```
Initialization
while not at end of the training data do
    Smoothen the data using w-day moving average as in (1)
    Standardize the features of the candidate data as in (2)
    Compute the distance d_ij as in (3)
    if d_ij ≤ TOL then
        Add the candidate data x_j to the training data S_i for all i
    end if
end while
Train the prediction classifier with the selected training data
Apply the prediction model to the test data
return Future trend for the test data
```

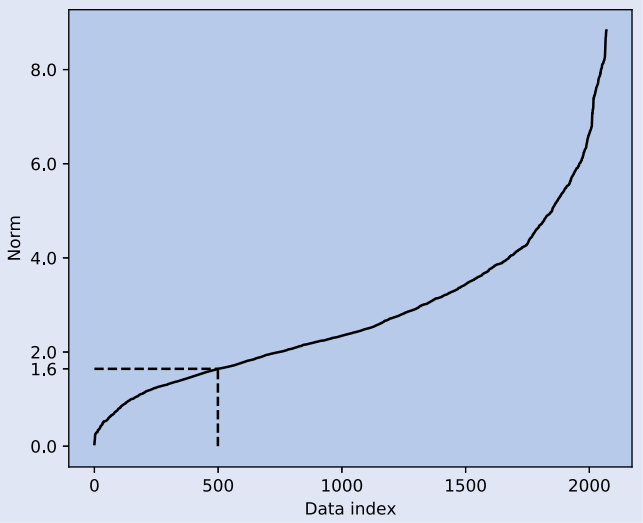

Figure 6. The d_{ij} values in (3) from the majority voting (MV) and ADS for Comcast Corporation (CMCSA) in the Communication Services sector of S&P 500 index for 10 years from October 22, 2010 to October 23, 2020 with the threshold $\mathsf{TOL} = 1.6$ for the 500 smallest values.

data are given so that it may not be practical due to the amount of calculation. Therefore, the above ADS algorithm is modified as in the following process and transformed into an inductive learning method.

Given a set X (called the *universe*) and a family Ω of the subsets of X, a *cover* C is a subfamily $C \subseteq \Omega$ of sets whose union is X. The universe X is defined by the whole data except the test data in this study. Given the universe X, let us consider the following algorithm:

(i) For an arbitrary but fixed point x_1 in X, the distance between x_1 and each point in X is measured and a subset $U_1 \subset X$ is defined by N closest points to x_1 in X. The L_2 norm is used as the distance measure in this study, which can be replaced by other measures. Note that x_1 itself is in U_1 and if there are several values of the same distance, even if the total number of points is N or more, it is included in the cluster.

(ii) For an arbitrary but fixed point x_2 in $U_1^c = X \setminus U_1$, the complement of U_1, the distance between x_2 and each point in X is measured and a subset $U_2 \subset X$ is defined by N closest points to x_2 in X. U_1 and U_2 may not be disjoint because the points in U_2 are chosen from the entire universe X not U_1^c.

(iii) In general, when i sets $\{U_1, U_2, \ldots, U_i\}$ are constructed, for an arbitrary but fixed point x_{i+1} in $(\cup_{k=1}^{i} U_k)^c = X \setminus \cup_{k=1}^{i} U_k$, the distance between x_{i+1} and each point in X is measured and a subset $U_{i+1} \subset X$ is defined by N closest points to x_{i+1} in X. Note that U_{i+1} may not be disjoint with $\cup_{k=1}^{i} U_k$.

(iv) Repeat the process above until the union of the subsets is X. If n subsets $U_1, U_2, \ldots, U_n$ are constructed and the union of these n subsets is X, then the collection $C = \{U_1, U_2, \ldots, U_n\}$ is a cover of X.

(v) Construct n machine learning algorithms, one for each subset $U_k \in C$, where $n = |C|$ is the cardinality of C. Each machine learning algorithm is trained with the data in U_k only as the training data.

The algorithm above represents the training process, which can be completed *before* the test data are given. Then, in the test process, the following steps are repeated for each x in the test data:

(i) The distance between x and the centroid of each subset $U_k \in C$ in the cover is measured to identify the subset with the closest distance

(ii) and (if U_j is the closest subset to x) then x is applied to the machine learning model trained with U_j to get its future prediction.

The above method will be called *Inductive ADS*. Thus, it is an algorithm, which constructs a cover $C = \{U_1, U_2, \ldots, U_n\}$ of the universe X and several machine learning models from one machine learning classifier, one model for each subset in the cover. Note that the above algorithm constructs the cover C with the subsets of the approximately same cardinality, i.e. $|U_1| \approx |U_2| \approx \cdots \approx |U_n| \approx N$.

Note that the algorithm of the Inductive ADS algorithm can be summarized as follows in algorithm 2.

Remark 1 Given a pair (X, Ω) and an integer k, the set covering decision problem checks if there is a subset of Ω of size k or less. In the set covering optimization problem, the input is a pair (X, Ω), and the task is to find a set covering that uses the fewest sets. The decision version of set covering is NP-complete, and the search of optimal set cover is NP-hard. Thus, even though a subcover of C, i.e. a subset of C that still covers X, may reduce the computational cost, finding it is not a simple process. In addition, a subcover may not lead to the improvement of the prediction accuracy because each set in the subcover does not necessarily improve training the pattern in it.

Remark 2 Like ADS, it is possible to configure a cluster by adjusting the distance between data by using TOL as a hyperparameter, and it is logically more suitable. However, in this study, N is set as a hyperparameter for comparison with the

Table 3. Variables considered in empirical tests.

	Variables	Values used in the experiments
Sectors in S&P 500	(Communication Services)	CMCSA, DIS, GOOG, IPG, DISH
	(Consumer Discretionary)	F, HD, LOW, MCD, NKE
	(Consumer Staples)	COST, KO, PEP, WBA, WMT
	(Energy)	APA, COP, CXO, SLB, XOM
	(Financials)	BEN, BLK, BRK-B, JPM, WFC
	(Health Care)	BSX, CVS, MDT, MRK, MYL
	(Industrials)	ALK, FAST, HON, MMM, UNP
	(Information Technology)	AAPL, ADBE, MSFT, NVDA, V
	(Materials)	DD, ECL, LIN, NEM, SHW
	(Real Estate)	AMT, CCI, DLR, EQIX, PLD
	(Utilities)	AES, D, DUK, FE, SO
ML Classifiers	(Single classifier)	LR, DT, KNN, SVM
	(Ensemble method)	MV, RF
Construction methods of the training data		All, Recent, Cluster, ADS

Algorithm 2 Inductive ADS Algorithm

Input: Universe X (historical stock prices) and parameters w in (1), N in (2) and N

Output: Future trend for the test data based on Inductive ADS method

1: Initialization
2: **for** each data in X **do**
3: Smoothen the data using w-day moving average as in (1)
4: Standardize the features of the candidate data as in (2)
5: **end for**
6: $C = \emptyset$
7: **while** $X \setminus \cup_{U \in C} U \neq \emptyset$ **do**
8: Choose an arbitrary but fixed point $x \in X \setminus \cup_{U \in C} U$
9: Measure the distance (3) between x and each point in X
10: Define a subset U of X with N closest points to x
11: Add U to C
12: **end while**
13: **for** each $U \in C$ **do**
14: Train the machine learning classifier with U
15: **end for**
16: **for** each $x \in$ <test data> **do**
17: Measure the distance (3) between x and the centroid of each $U \in C$
18: Find $U \in C$ with the closest distance to x
19: Predict x with the machine learning classifier trained with U
20: **end for**
21: **return** Future trend for the test data

Recent and Cluster methods, and each group is configured to contain approximately N data.

3. Data and measures

3.1. Data and hyperparameters

This section provides the information regarding the data sets and the variables and hyperparameters used in the machine learning methods. Table 3 summarizes the financial data used in the empirical experiments, the methods to construct the training data, and the machine learning methods to train them.

The current study predicts the future movements of the stocks included in S&P 500 index. S&P 500 consists of 11 sectors (Communication Service, Consumer Discretionary, Consumer Staples, Energy, Financials, Health Care, Industrials, Information Technology, Materials, Real Estate, Utilities) and we use five stocks from each sector as in table 3 for 10 years from October 22, 2010 to October 23, 2020.

Figure 7 shows the values of those stocks from each sector for 10 years and table 4 summarizes the statistics of the data in each sector. The test data are defined by the last 400 values and the remaining data are used as the training data. Out of the training data, 25% is used as the validation data for tuning the hyperparameters.

The following classifiers of the machine learning are used for the training and the prediction of financial data as in table 3:

- Single classifiers: Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbors (KNN), Support Vector Machine (SVM)
- Ensemble methods: Majority Voting (MV), Random Forest (RF). MV in this study considers the majority voting of 4 single classifiers, i.e. LR, DT, KNN, and SVM.

First, let's take a brief look at each classifier. The LR model measures the probability that a sample belongs to a specific class. It calculates the sum of the weights of the input features as follows:

$$\hat{p} = \phi\left(\sum_{i=0}^{n} w_i x_i\right) = \phi\left(\mathbf{w^T x}\right)$$

where w_0 is a bias, w_i are the weights of the input features, $x_0 = 1$, x_i is the ith feature value and $\phi(z) = 1/(1+e^{-z})$ is a sigmoid function. Here, the weight vector, $\mathbf{w}$, is determined so that the next cost function is minimized

$$J(\mathbf{w}) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} log(\hat{p}^{(i)}) + (1-y^{(i)})\log(1-\hat{p}^{(i)})\right) + \frac{\lambda}{2}\|w\|^2$$

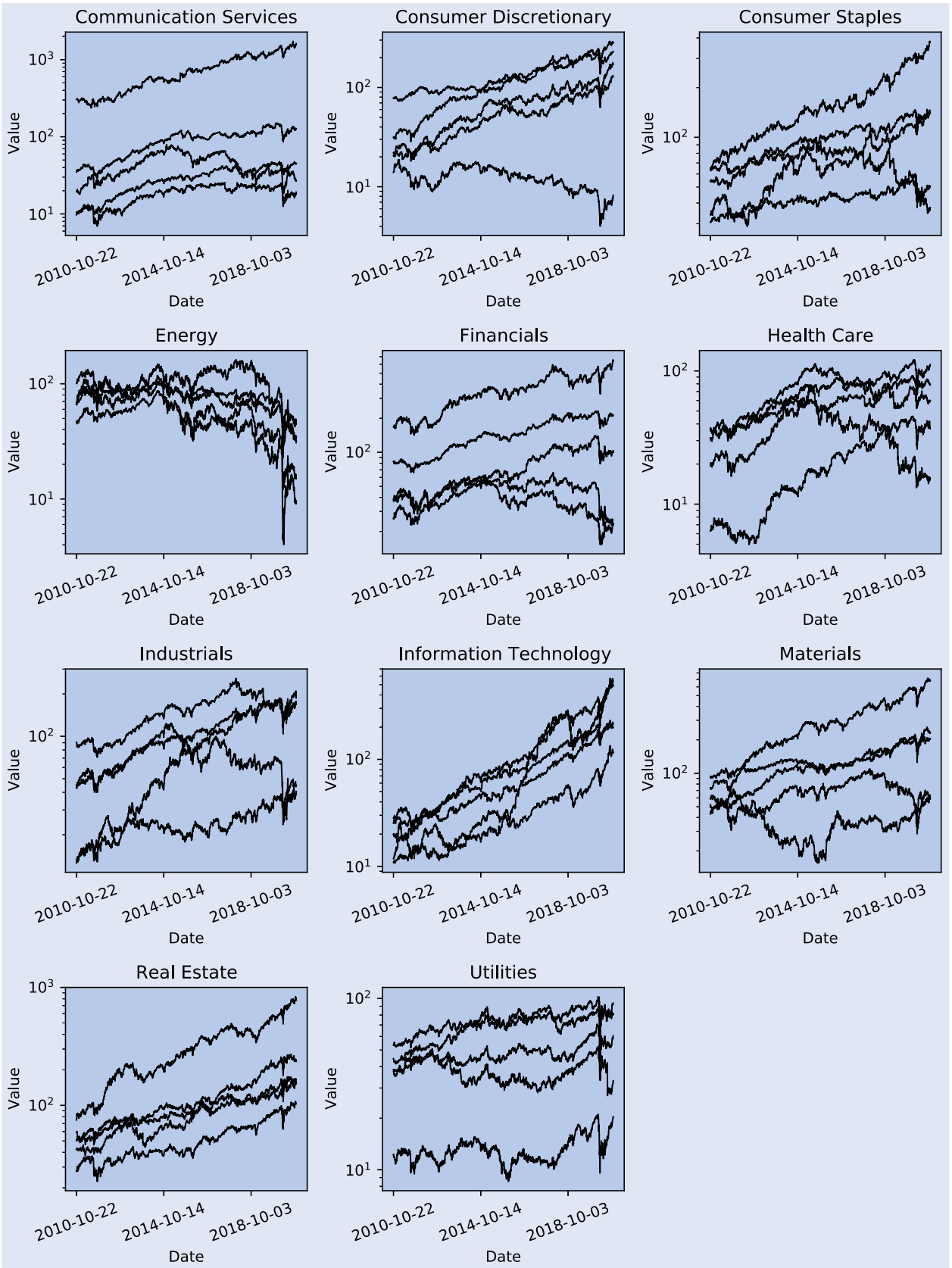

Figure 7. 5 stocks from 11 sectors in S&P 500 in table 3 from October 22, 2010 to October 23, 2020.

Table 4. The statistics of the financial data in each sector of S&P 500 used in this study.

Sectors	Count	Mean	Std	Min	25%	50%	75%	Max
Communication Services	2518	183.60	86.07	60.04	108.08	171.51	258.87	406.48
Consumer Discretionary	2518	77.39	34.02	28.43	49.68	73.24	103.10	169.81
Consumer Staples	2518	88.57	28.02	46.13	67.26	85.30	104.59	166.60
Energy	2518	75.14	20.44	21.24	63.84	75.41	89.75	120.82
Financials	2518	129.98	43.10	54.80	96.19	128.91	164.35	230.69
Health Care	2518	50.18	16.84	21.62	35.39	50.76	60.85	89.71
Industrials	2518	86.83	31.17	35.24	59.69	87.24	109.87	160.32
Information Technology	2518	84.65	71.15	17.05	30.46	52.33	131.68	337.99
Materials	2518	129.09	53.29	49.41	88.43	122.90	163.40	277.94
Real Estate	2518	132.89	61.96	45.14	86.13	118.20	172.18	310.05
Utilities	2518	48.10	7.22	33.14	42.52	48.40	53.26	67.28

where $y^{(i)}$ is the predicted value, 0 if $\hat{p}$ is less than 0.5, and 1 otherwise. The regulatory hyperparameter λ can be used to adjust how well the training data fit while keeping the weights small. While LR is simple, easy to implement, and easy to update, it is sensitive to outliers because it maximizes the conditional likelihood of training data.

The DT is a process of dividing a set of data used for learning into subsets according to appropriate partitioning criteria or partitioning tests. It repeats recursively over each of the divided data subsets, and continues until no more new predicted values are added due to the segmentation, or the node of the subset has the same value as the target variable. The cost function used here is as follows:

$$J(k, t_k) = \frac{m_1}{m} G_1 + \frac{m_2}{m} G_2$$

where $G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$ is a Gini impurity, $p_{i,k}$ is the proportion of samples in class k among the training samples in the ith node and m_i is the number of each sample. The advantage of decision trees compared to other methods is that it is easy to interpret and understand the results, is stable, and works well with large data. On the other hand, it is difficult to know the optimal decision tree, and if the training data are not properly generalized, an overfitting problem that creates too complex decision trees may occur.

The SVM is a model that maximizes the margin defined by the distance between the hyperplane separating the class and the training sample closest to this hyperplane. The larger the margin, the lower the generalization error, so the constraint that maximizes the margin, $2/\|\mathbf{w}\|$ can be written as:

$$y^{(i)} \left(w_0 + \mathbf{w}^T \mathbf{x}^{(i)}\right) \geq 1, \quad \text{for all } i$$

where $y^{(i)} = 1$ for positive class and $y^{(i)} = -1$ for negative class. For soft margin classification, which allows for limited margin errors, use a slack variable, $\zeta^{(i)}$, to minimize the following objective function

$$\frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} \zeta^{(i)}$$

under the constraint $y^{(i)}(w_0 + \mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 - \zeta^{(i)}$, for all i, $\zeta^{(i)} \geq 0$. One can adjust the cost for the classification error through hyperparameter C, the greater the cost for the error. In general, we can use linear SVM and, for nonlinear problems, we can use kernel techniques to find the partitioning hyperplane in high dimensional space. Among the most widely used kernels, RBF(Radial Basis Function) is defined as:

$$\mathcal{K}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left(-\gamma \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2\right)$$

Here the kernel can be interpreted as a function of similarity between samples, and γ can be understood as a hyperparameter that limits the size of the Gaussian sphere. The larger the γ value, the less the influence or range of the support vector is.

The KNN algorithm is instance-based learning that stores a set of training data in memory instead of learning a discriminative function from the training data. Select parameter k and distance metrics, find the k nearest neighbors in the sample, and assign a class label by majority vote. Choosing the right k is important for the right balance between overfitting and underfitting, and the distance measure is defined as:

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(\sum_k \left|x_k^{(i)} - x_k^{(j)}\right|^p\right)^{\frac{1}{p}}.$$

The KNN makes predictions without model training, but the computational cost of the prediction phase is high and it is prone to overfitting due to the curse of the dimensions.

MV and RF are used as the ensemble method. The MV uses LR, DT, KNN, and SVM to determine the label by the principle of majority vote. The RF is the ensemble method for learning multiple decision trees. Trees have slightly different characteristics due to randomness, which de-correlates predictions and improves generalization performance. Also, randomization helps to work well for data containing noise. The RF is not as easy to interpret as DT, but has the advantage of not having to put much effort into tuning the hyperparameter.

Above classifiers and the K-means method are implemented by the *scikit-learn* python package (Raschka 2015). For each machine learning method, the hyperparameters in table 5 are considered and the training and the validation procedures find their optimal hyperparameter values.

Four construction methods in table 2 are used to build the training data:

- *All* uses all the known data available from the past 10 years except 400 values reserved for the test data.
- *Recent* defines the training data with a subset of the All-type training data with respect to the close proximity in time to the test data. In this study, 500 data are used for Recent-type training data.

Table 5. The hyperparameters for each machine learning classifier.

Classifier	Hyperparameters		Values used in the computation
Logistic regression	λ		$10^{-3}, 10^{-2}, \ldots, 10^{4}$
Decision tree	Depth		$1, 2, 3, \ldots, 7$
KNN	p		1, 2, 3, 4
	k		1, 2, 3, 4
SVM	Kernel = linear	C	$10^{-3}, 10^{-2}, \ldots, 10^{2}$
	Kernel = RBF	C	$10^{-3}, 10^{-2}, \ldots, 10^{2}$
		γ	0.3, 0.5, 0.7, 0.9, 1.0
Majority voting	All the parameters above		
Random forest	The number of trees		50, 100, 150, ..., 500

- *Cluster* performs unsupervised learning to partition the All-type training data into a few clusters. Given each value in the test data, the same unsupervised learning method is applied to identify the corresponding cluster and the data in that cluster only are used as the training data for the given test data. 5 clusters are considered in this study.
- *Inductive ADS* introduces a cover of the All-type training data. The cover is the union, $\cup_{j=1}^{N} U_j$, of subsets of the All-type training data defined by the L_2 norm (3) and each subset U_j defines a training data as explained in section 2. Each subset U_j consists of approximately 500 data values in this study.

Many studies consider the prediction of the up-down movement of the stock or index in the next day. In this study, the up-down of the financial data in $p = 20$ days is predicted, which can be applied to the portfolio management with rebalancing in every, for example, 20 days. Section 4.2 shows the empirical results of such portfolio management. The financial values for the $\mathsf{N} = 20$ days are used as the features on each day, and $\mathsf{w} = 20$ is used for smoothing.

3.2. Measures

There are several measures about the accuracy of the prediction. We can count the *true positive* (TP, the number of correct predictions for positives), *true negative* (TN, the number of correct predictions for negatives), *false positive* (FP, the number of wrong predictions for positives), and *false negative* (FN, the number of wrong predictions for negatives). Then the *accuracy* (ACC) is defined by the number of correct predictions divided by the total number of predictions,

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

and alternatively, the *prediction error* (ERR) defined by

$$ERR = \frac{FP + FN}{TP + TN + FP + FN} = 1 - ACC$$

can be used, see Pesaran and Timmermann (1992). The *false positive rate* (FPR) and the *true positive rate* (TPR) are performance metrics defined by

$$FPR = \frac{FP}{TN + FP}, \quad TPR = \frac{TP}{TP + FN}.$$

Another metric *precision* (PRE) is defined by

$$PRE = \frac{TP}{TP + FP}$$

and the *F1-score* is then defined by the harmonic average of PRE and the *recall* (REC)

$$F1 = \frac{2}{\frac{1}{PRE} + \frac{1}{REC}} = 2\frac{PRE \times REC}{PRE + REC}$$

where REC is synonymous to TPR.

Table 6 shows the confusion matrices for various classifiers in table 3 and corresponding accuracies and F1-scores. For instance, TP and TN for LR are 6961 and 3749 whereas its FP and FN are 3339 and 3536 so that the accuracy and F1 score for LR are

$$ACC_{LR} = \frac{6961 + 3749}{6961 + 3749 + 3339 + 3536} = 0.61$$

and

$$F1_{LR} = 2\frac{6961/(6961 + 3339) \times 6961/(6961 + 3563}{6961/(6961 + 3339) + 6961/(6961 + 3563)} = 0.67,$$

Table 6. The confusion matrices with respect to the ML classifiers.

	LR		DT		KNN		SVM		MV		RF	
	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down
Up (prediction)	6961	3339	5517	3939	4413	3201	7268	3974	4972	2714	4648	3332
down (prediction)	3536	3749	4981	3150	6084	3887	3229	3115	5525	4375	5850	3756
ACC	0.61		0.49		0.47		0.59		0.53		0.48	
F1-score	0.67		0.55		0.49		0.67		0.55		0.50	

Table 7. The confusion matrices with respect to the construction methods of the training data.

	All		Recent		Cluster		ADS	
	Up	Down	Up	Down	Up	Down	Up	Down
Up (prediction)	8616	5122	7553	5152	6673	4965	10938	5262
Down (prediction)	7131	5511	8193	5481	9074	5669	4808	5372
ACC	0.54		0.49		0.47		0.62	
F1-score	0.58		0.53		0.49		0.68	

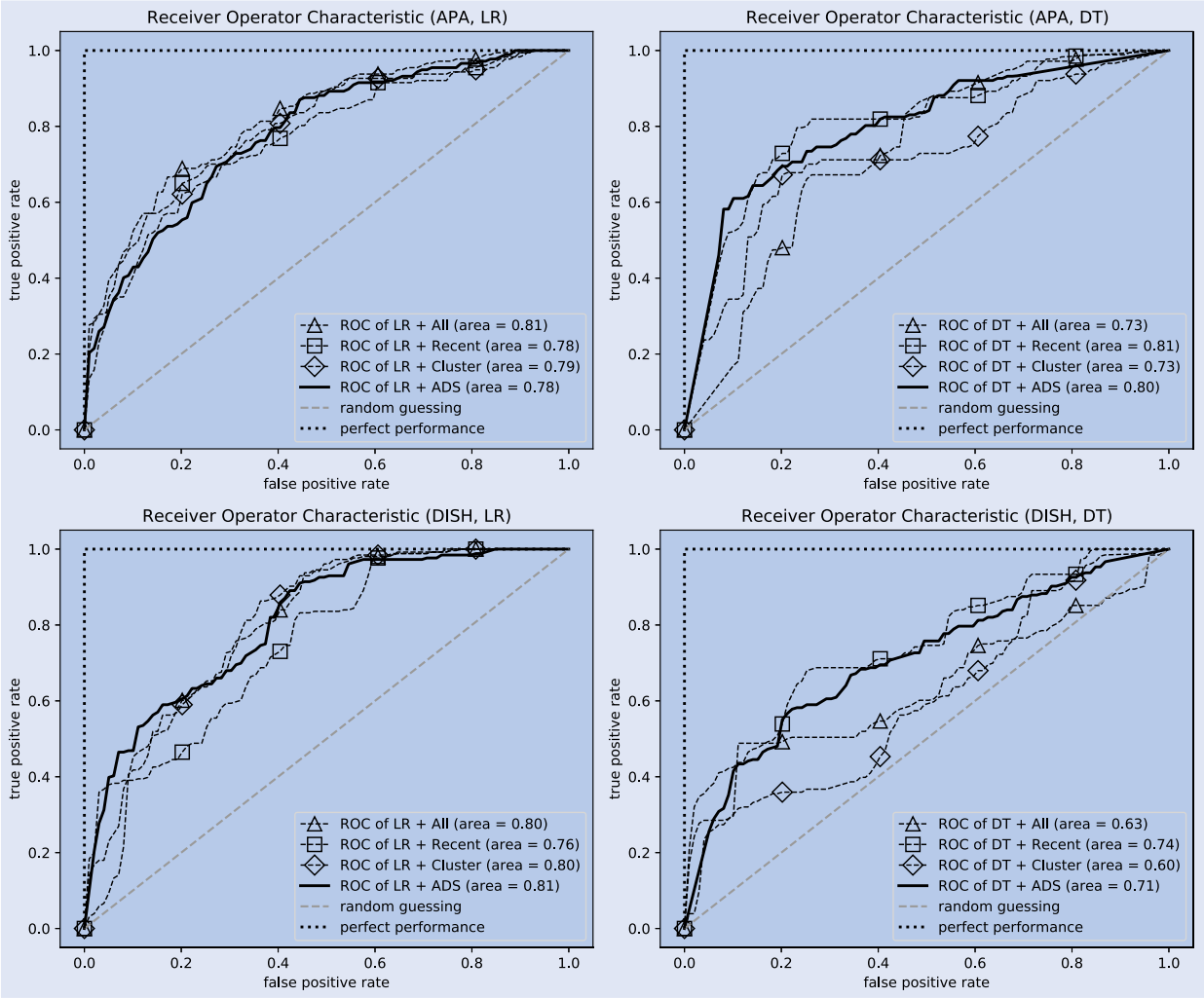

Figure 8. Receiver Operator Characteristic curves for APA (two upper pictures) and DISH (two lower pictures) when LR and DT are trained in various ways.

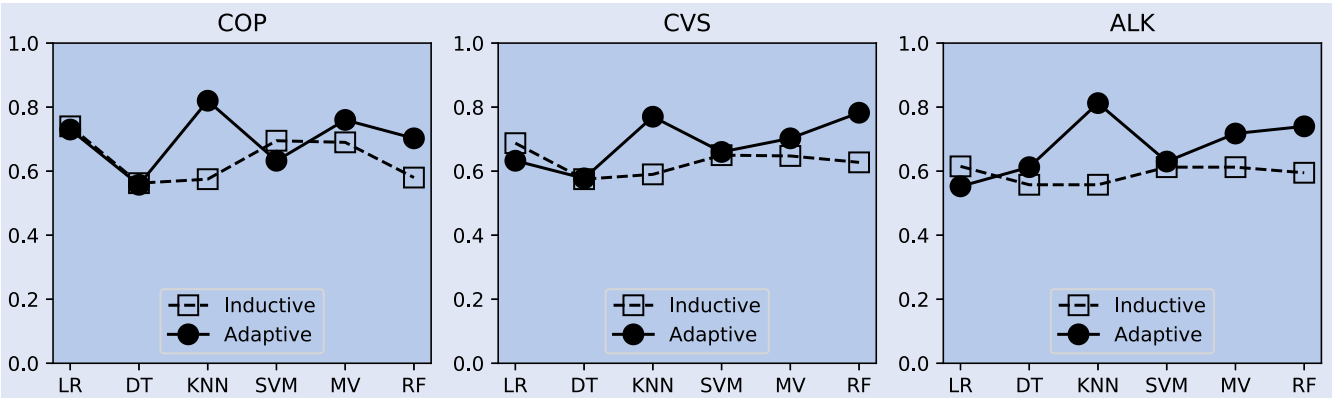

Figure 9. The original Adaptive Data Selection (filled circle) vs Inductive ADS (square) for COP, CVS, ALK from the Energy, Health Care and Industrial sectors, respectively.

Table 8. The prediction accuracies for 11 sectors in S&P 500 from six machine learning classifiers based on four construction methods of training data.

		All	Recent	Cluster	ADS	Mean
LR	Communication Services	0.70	0.59	0.60	0.69	0.64
	Consumer Discretionary	0.73	0.52	0.44	0.71	0.60
	Consumer Staples	0.64	0.51	0.46	0.66	0.56
	Energy	0.64	0.50	0.53	0.72	0.59
	Financials	0.65	0.64	0.55	0.69	0.63
	Health Care	0.64	0.54	0.60	0.70	0.62
	Industrials	0.60	0.56	0.51	0.63	0.58
	Information Technology	0.78	0.66	0.38	0.77	0.65
	Materials	0.72	0.65	0.52	0.70	0.65
	Real Estate	0.68	0.43	0.47	0.68	0.57
	Utilities	0.63	0.56	0.55	0.68	0.60
DT	Communication Services	0.55	0.50	0.54	0.61	0.55
	Consumer Discretionary	0.41	0.40	0.40	0.55	0.44
	Consumer Staples	0.38	0.37	0.38	0.55	0.42
	Energy	0.46	0.48	0.46	0.52	0.48
	Financials	0.54	0.51	0.45	0.57	0.52
	Health Care	0.49	0.49	0.56	0.57	0.53
	Industrials	0.49	0.58	0.45	0.56	0.52
	Information Technology	0.32	0.49	0.51	0.67	0.50
	Materials	0.58	0.35	0.44	0.57	0.49
	Real Estate	0.47	0.46	0.41	0.61	0.49
	Utilities	0.48	0.49	0.45	0.55	0.49
KNN	Communication Services	0.51	0.52	0.49	0.58	0.52
	Consumer Discretionary	0.37	0.39	0.37	0.54	0.42
	Consumer Staples	0.39	0.44	0.39	0.54	0.44
	Energy	0.45	0.46	0.45	0.55	0.48
	Financials	0.48	0.49	0.49	0.54	0.50
	Health Care	0.49	0.48	0.50	0.57	0.51
	Industrials	0.47	0.49	0.47	0.56	0.50
	Information Technology	0.32	0.33	0.32	0.62	0.40
	Materials	0.44	0.44	0.44	0.54	0.46
	Real Estate	0.44	0.43	0.44	0.57	0.47
	Utilities	0.46	0.48	0.47	0.54	0.49
SVM	Communication Services	0.64	0.61	0.58	0.65	0.62
	Consumer Discretionary	0.65	0.47	0.45	0.67	0.56
	Consumer Staples	0.61	0.53	0.38	0.63	0.54
	Energy	0.65	0.48	0.50	0.70	0.58
	Financials	0.58	0.49	0.54	0.62	0.56
	Health Care	0.60	0.58	0.50	0.66	0.58
	Industrials	0.59	0.59	0.52	0.62	0.58
	Information Technology	0.72	0.67	0.46	0.73	0.65
	Materials	0.71	0.64	0.56	0.68	0.65
	Real Estate	0.67	0.50	0.52	0.68	0.59
	Utilities	0.62	0.55	0.51	0.67	0.59
MV	Communication Services	0.57	0.54	0.53	0.65	0.57
	Consumer Discretionary	0.47	0.39	0.39	0.61	0.46
	Consumer Staples	0.42	0.39	0.38	0.61	0.45
	Energy	0.62	0.48	0.49	0.67	0.56
	Financials	0.60	0.54	0.48	0.63	0.56
	Health Care	0.59	0.53	0.53	0.66	0.58
	Industrials	0.53	0.59	0.49	0.62	0.56
	Information Technology	0.47	0.48	0.32	0.70	0.49
	Materials	0.69	0.48	0.48	0.63	0.57
	Real Estate	0.53	0.42	0.43	0.64	0.50
	Utilities	0.53	0.52	0.46	0.60	0.53
RF	Communication Services	0.53	0.51	0.53	0.59	0.54
	Consumer Discretionary	0.35	0.40	0.36	0.55	0.42
	Consumer Staples	0.38	0.38	0.38	0.55	0.42
	Energy	0.44	0.46	0.43	0.55	0.47
	Financials	0.51	0.52	0.49	0.56	0.52
	Health Care	0.51	0.46	0.51	0.58	0.51
	Industrials	0.47	0.48	0.46	0.57	0.49
	Information Technology	0.35	0.40	0.34	0.64	0.43

(*Continued*).

Table 8. Continued.

		All	Recent	Cluster	ADS	Mean
	Materials	0.45	0.44	0.46	0.56	0.48
	Real Estate	0.43	0.43	0.43	0.61	0.47
	Utilities	0.45	0.48	0.46	0.56	0.49
Mean	Communication Services	0.58	0.54	0.55	0.63	0.58
	Consumer Discretionary	0.50	0.43	0.40	0.61	0.48
	Consumer Staples	0.47	0.44	0.39	0.59	0.47
	Energy	0.54	0.47	0.48	0.62	0.53
	Financials	0.56	0.53	0.50	0.60	0.55
	Health Care	0.55	0.51	0.53	0.62	0.56
	Industrials	0.52	0.55	0.48	0.60	0.54
	Information Technology	0.49	0.51	0.39	0.69	0.52
	Materials	0.60	0.50	0.49	0.61	0.55
	Real Estate	0.54	0.45	0.45	0.63	0.52
	Utilities	0.53	0.51	0.48	0.60	0.53

respectively. In table 6, LR and SVM show better results than the others.

Table 7 shows the confusion matrices for the construction methods in table 3 and corresponding metrics. Note that the ACC and F1 score of ADS are larger than those of the others by 8–15% and 10–19%, respectively.

A *receiver operator characteristic* (ROC) plot is another measure to compare classifiers with respect to FPR and TPR. The diagonal of an ROC plot represents naive random guessing. Classifiers that fall below the diagonal are considered as worse than the naive guess. A perfect classifier would be located at the top left corner with TPR 1 and FPR 0. For example, figure 8 shows the ROC curves for APA from the Energy sector (two upper plots) and DISH from the Communication Services sector (two lower plots) when the classifiers LR and DT are trained in various ways. For APA, all the methods give similar results for LR and the ADS method outperforms the other methods for DT. For DISH, the ADS results are better than the others for both LR and DT. The area under the ROC curve, called AUC, characterizes the performances of classifiers. The analyses derived from the ACC value (or equivalently ERR) and ROC AUC value are consistent with each other in most cases and the ACC value is used as the measure of accuracy in this study.

4. Empirical results

The current study introduces ADS and Inductive ADS in sections 2.2 and 2.3. In most cases, the original ADS slightly outperforms the Inductive ADS or both methods show similar accuracies. Figure 9 shows 3 examples: COP (Energy sector), CVS (Health Care sector), and ALK (Industrial sector).

Even though there are situations where the original ADS can be applicable, the original ADS is not an inductive learning so the results of the Inductive ADS only are presented in the empirical tests. Thus, ADS below in this section represents the Inductive ADS.

4.1. Prediction accuracies

Table 8 shows the prediction accuracies for 11 sectors in S&P 500 from six machine learning classifiers (LR, DT, KNN, SVM, MV, RF) based on four construction methods of training data (All, Recent, Cluster, ADS). It is observed that the superiority of ADS-based predictions over other methods regardless of the sector or the machine learning method, as will be explained below.

Figure 10 shows the best training scores from machine learning vs the corresponding prediction accuracies for each of 11 sectors in S&P 500 with respect to the construction method of the training data. Each plot represents the results from 5 stocks in each sector. ADS (filled circle) results in the accuracy of about 60% in every sector, while All (triangle), Recent (square), and Cluster (diamond) result in the accuracies below the probability of naive guessing (dotted line) in some sectors, for example, Consumer Discretionary and Consumer Staples sectors. In addition, even when all the construction methods produce the accuracies greater than a half, the value from ADS is larger than the others in most sectors.

Figure 11 represents figure 10 with respect to the way how the training data are constructed. Each plot represents the results from 55 stocks in 11 sectors. As observed in figure 10, the prediction accuracies of All, Recent and Cluster are even below 0.5 (dotted line) for some sectors. In particular, the

Table 9. The prediction accuracies from six machine learning classifiers based on four construction methods of training data.

	LR	DT	KNN	SVM	MV	RF	Mean
All	0.67	0.47	0.44	0.64	0.55	0.44	0.54
Recent	0.56	0.46	0.45	0.55	0.49	0.45	0.49
Cluster	0.51	0.46	0.44	0.50	0.45	0.44	0.47
ADS	0.69	0.58	0.56	0.66	0.64	0.57	0.62
Mean	0.61	0.49	0.47	0.59	0.53	0.48	0.53

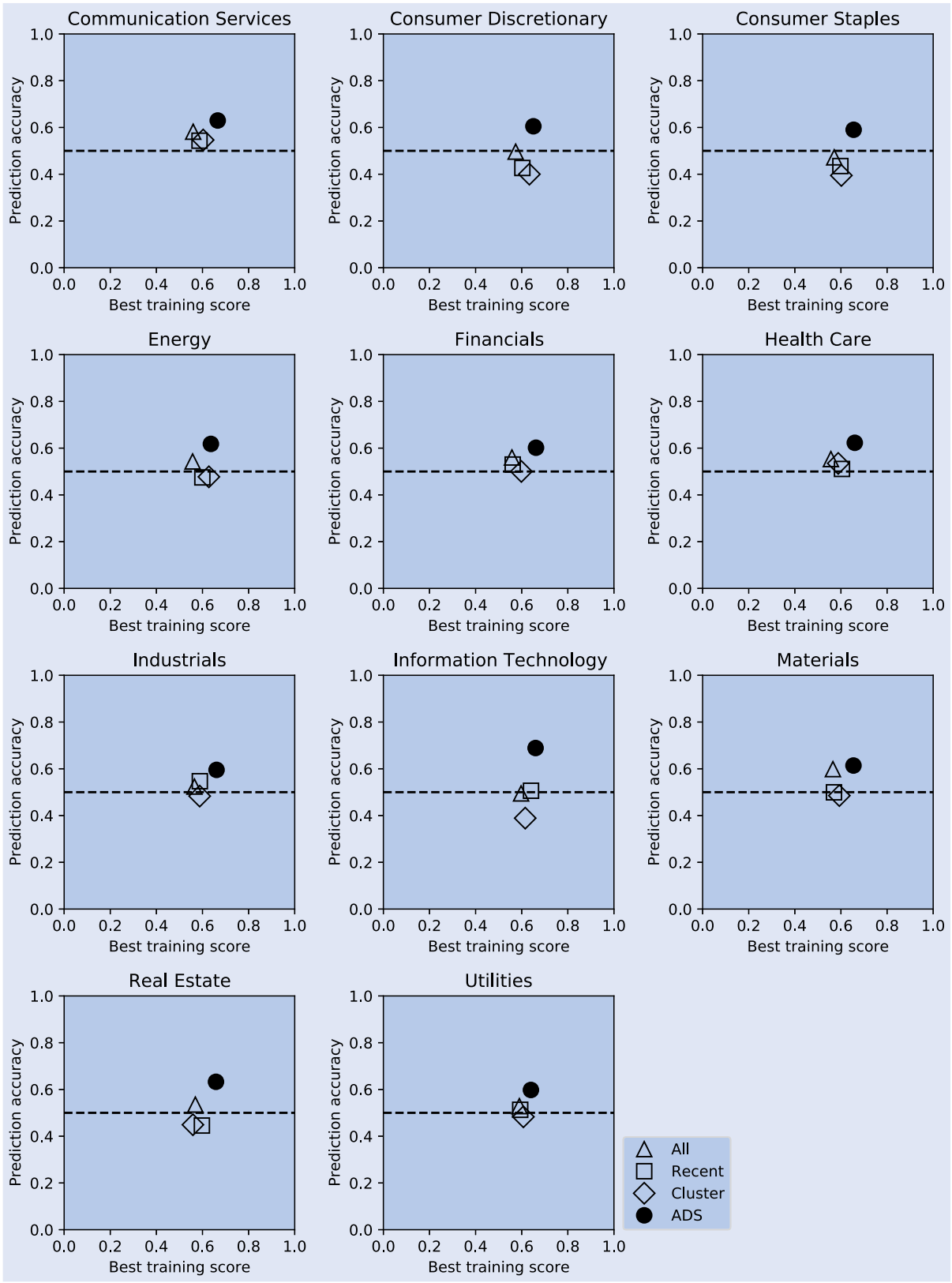

Figure 10. The best training scores from machine learning vs the corresponding prediction accuracies for each of 11 sectors in S&P 500 with respect to the construction method of the training data.

prediction accuracies for 6 sectors out of 11 are below 0.5 in case of Recent or Cluster. On the other hand, the accuracies of ADS are above 0.5 for all 11 sectors, which is consistent with the results in figure 10.

Figure 12 shows the prediction accuracies of the machine learning classifiers for each sector in S&P 500. In each sector, ADS (filled circle) ranges mostly between 0.6 and 0.8 for all the classifiers and it is above the values from All

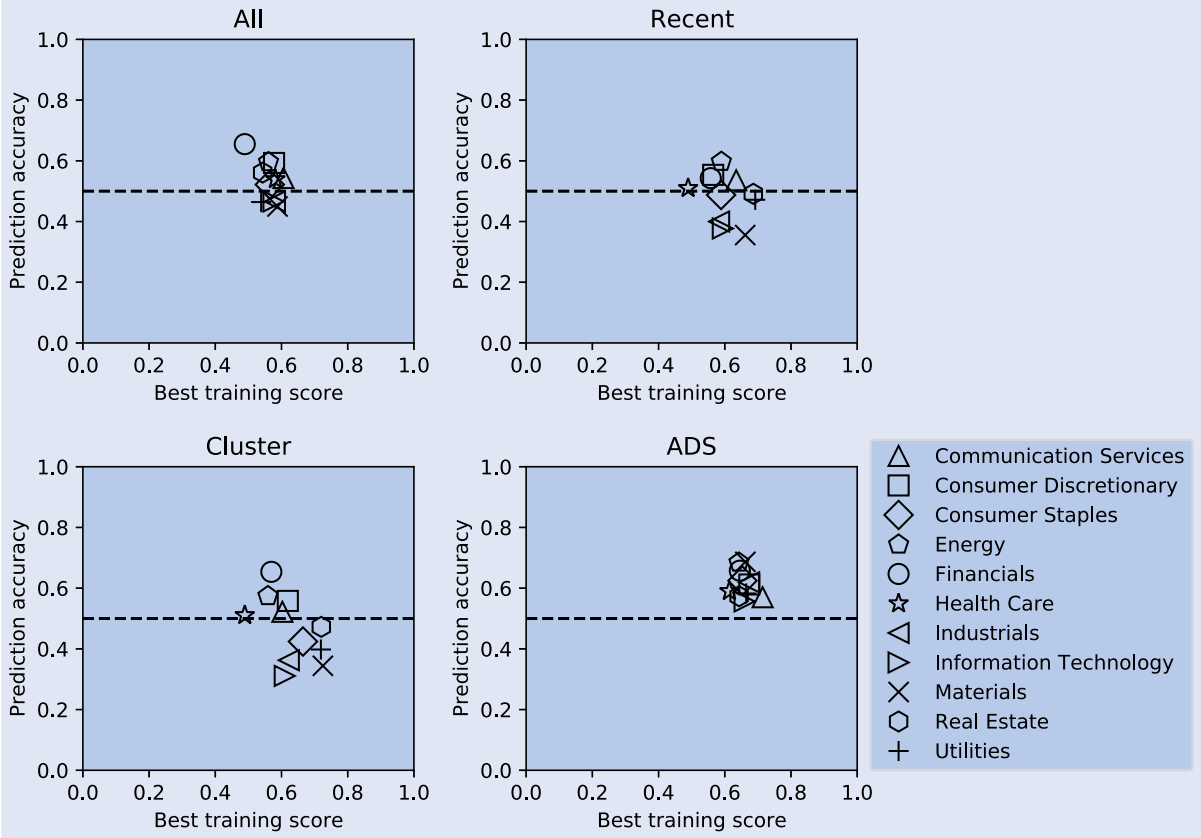

Figure 11. The best training scores from machine learning vs the corresponding prediction accuracies with respect to the construction method of the training data.

(triangle), Recent (square) and Cluster (diamond) regardless of the machine learning classifier. All, Recent and Cluster methods show large variations and their accuracies are low for some sectors such as Consumer Discretionary, Consumer Staples, Information Technology sectors. Some produce accuracies even lower than a half. The ADS, on the other hand, is superior in the sense that the lower bounds for the accuracies are always greater than the probability of naive random guessing for any financial data or any machine learning classifier.

Figure 13 shows the ranges of the prediction accuracy for various machine learning classifiers and construction methods of the training data. In financial risk management, the worst case probability is very important for securing individual assets and the lower bounds of the ranges represent the worst-case accuracies. Note that the lower bounds for All, Recent and Cluster are mostly between 30% and 40%, which implies that their predictions may be worse than naive random guessing, and that the accuracies of the ADS are, on the other hand, bounded below by 0.5 for all machine learning methods.

Table 9 shows the prediction accuracies from 6 machine learning classifiers (LR, DT, KNN, SVM, MV, RF) based on 4 construction methods of the training data (All, Recent, Cluster, ADS). The ADS method results in 62% of accuracy in average while the other three methods result in 47–54% only, among which the All-type construction method is better than Recent- or Cluster-type methods by 5–7%. Figure 13 and table 9 confirm again the superiority of ADS over the other methods. Note that the machine learning classifiers in table 9 show that LR, SVM and MV methods seem to be more reliable than DT, KNN and RF regardless of the construction method of the training data. For instance, the LR classifier when the training data are constructed by All-type method gives 67% of prediction accuracy while the DT, KNN and RF classifiers with the same training data gives less than 47% of accuracies. When the ADS method is used, the accuracies are overall improved, but the distinction among different classifiers still exists. The DT classifier with the ADS-type training data gives 58% of accuracy, which is better than 46–47% from All-, Recent-, or Cluster-type training data, but it is still worse than 69% from the LR classifier or 66% from the SVM classifier. Based on these observations, we consider the results with respect to the machine learning classifiers in figure 14 through figure 16 before we consider the results from the portfolio management.

Figure 14 shows the prediction accuracies of 11 sectors in S&P 500 for single classifiers(LR) and ensemble methods (MV) in machine learning. All (triangle), Recent (square), Cluster (diamond) are mostly below ADS (filled circle) in both cases, which implies that the prediction of ADS outperforms the others in most sectors. In addition, Cluster shows large variations. For instance, in case of LR, even though Cluster makes good prediction for the Health Care sector, it

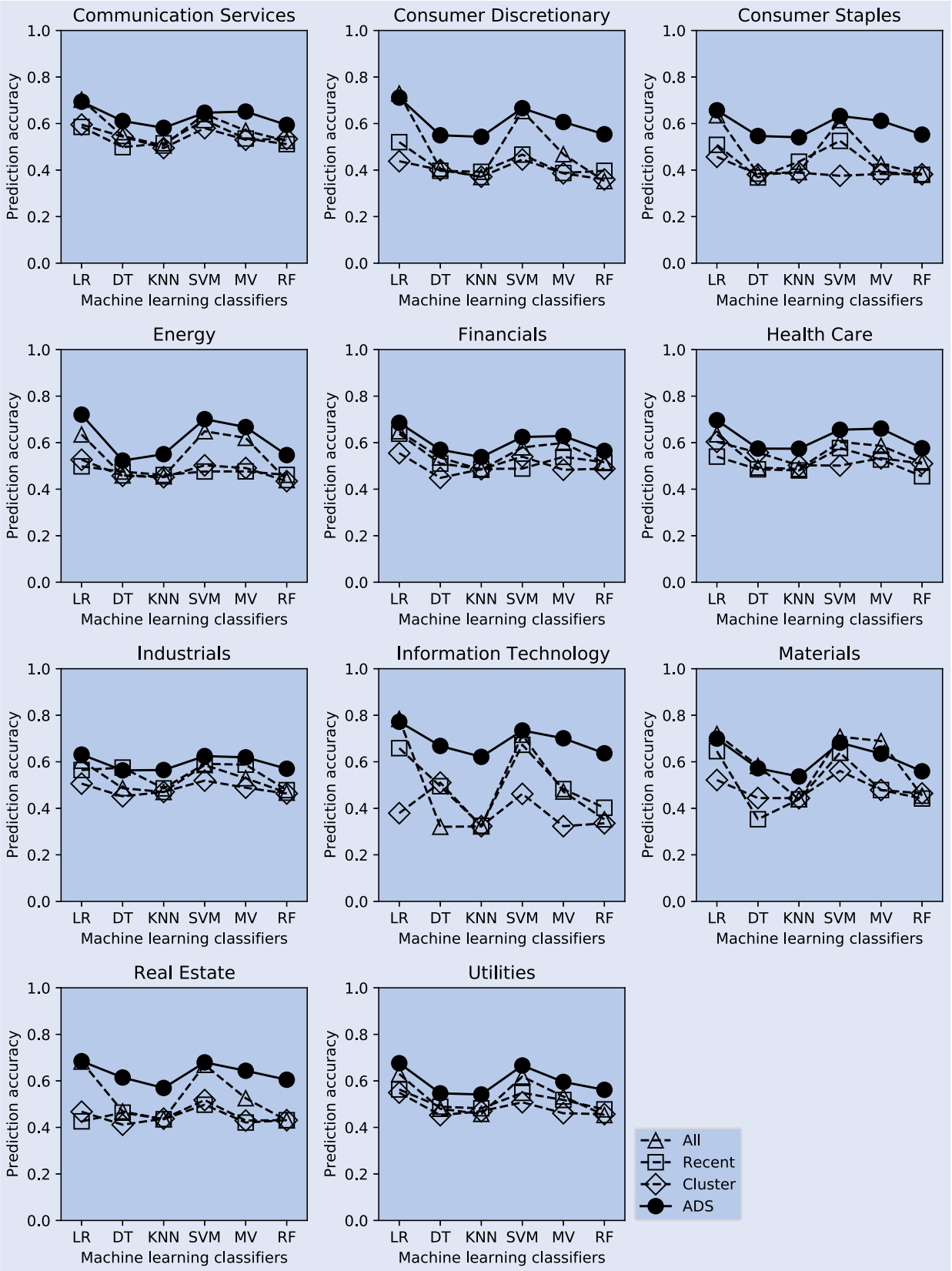

Figure 12. Prediction accuracies of the machine learning classifiers for each sector in S&P 500.

makes poor prediction for the Information Technology sector. ADS, on the other hand, shows good prediction results for all sectors for both LR and MV.

Figure 15 shows the effects of the construction method of the training data. Each symbol in the plots represents the mean of the results from all 11 sectors. Points below 0.5 from All (triangles), Recent (squares) and Cluster (diamonds) for DT, KNN and MV imply that the corresponding predictions are inferior to naive random guessing. Note that ADS (filled circles) is superior in the sense that the lower bound

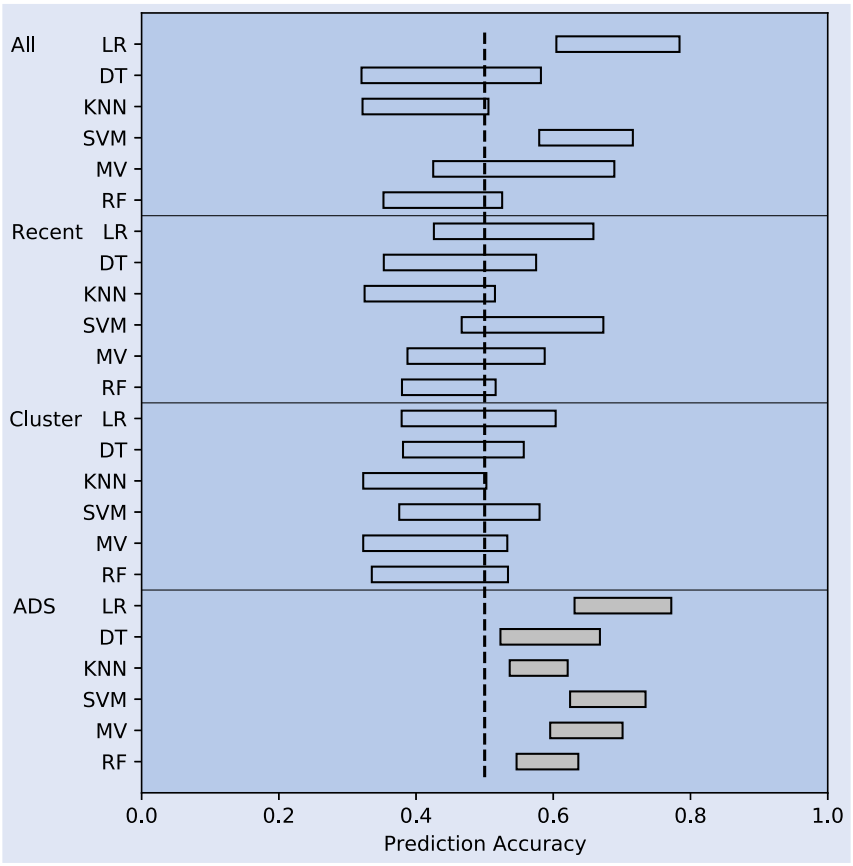

Figure 13. The ranges of prediction accuracies for various machine learning classifiers and construction methods of the training data.

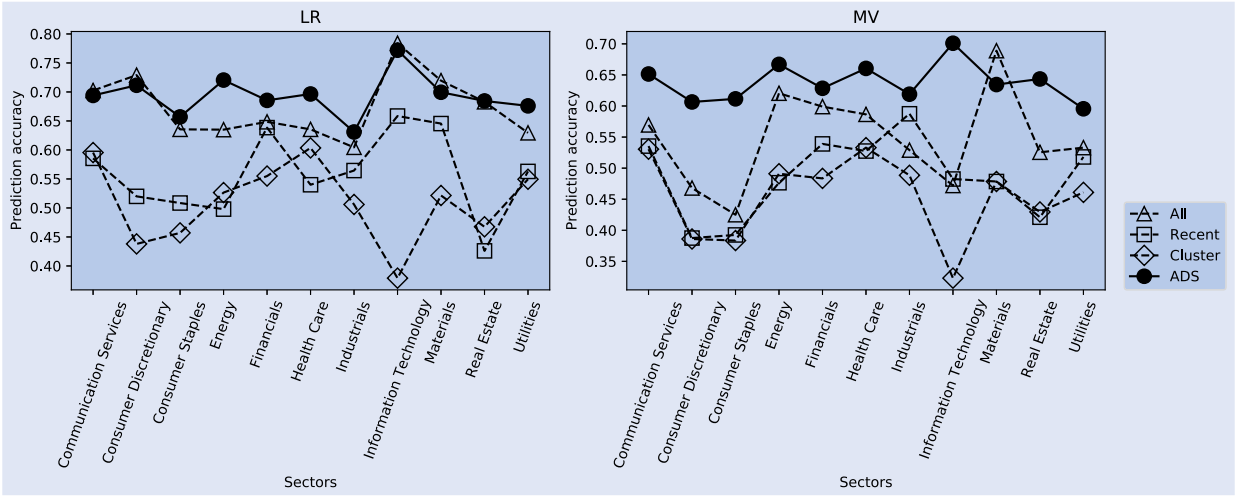

Figure 14. Prediction accuracies of 11 sectors in S&P 500 for LR and MV.

for the prediction is always greater than the random guessing regardless of the classifier.

Figure 16 represents figure 15 with respect to the input data type, which confirms the superiority of ADS again. ADS shows the accuracy of about 60% or more while the other methods may produce the accuracies below 50%.

4.2. Portfolio

Let us consider the portfolio management for 400 days in the test data. One first makes predictions of the up-down movements of 55 stocks in 11 sectors of S&P 500 and constructs the portfolio with only sectors which are expected to increase in 20 days. The shares of the stocks included in the portfolio are kept to be the same and the initial shares are determined from the assumption that $10 is initially invested to the whole portfolio. The portfolio is rebalanced every 20 days. If all 11 sectors are expected to decrease, the stocks in the portfolio will be exchanged with cash.

Figure 17 shows the values of the portfolio when the up-down prediction is made by one of the six machine learning

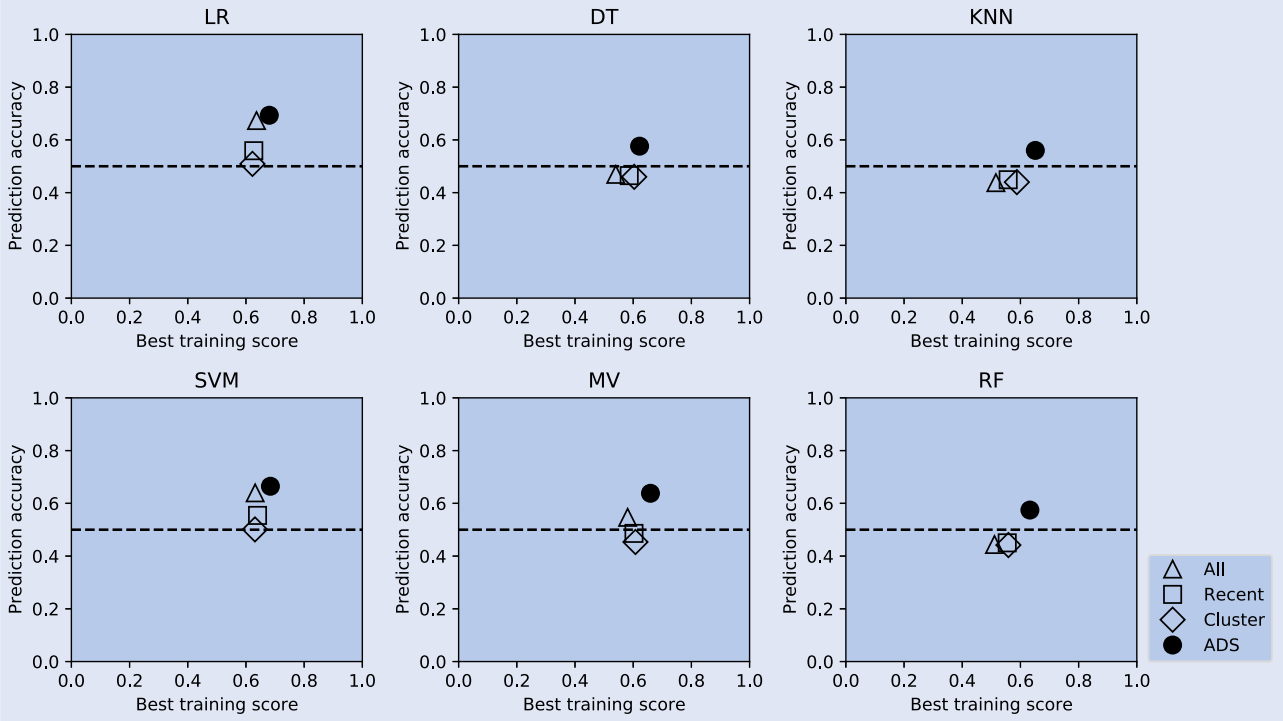

Figure 15. Prediction accuracies vs best training scores for each machine learning classifier.

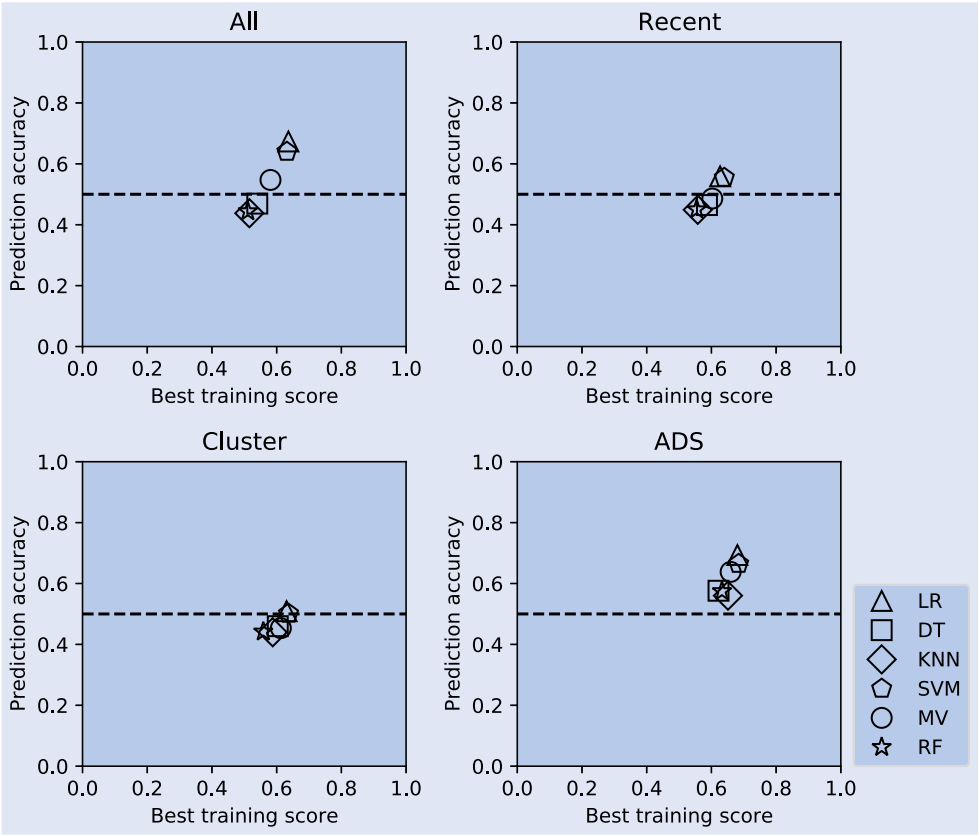

Figure 16. Prediction accuracies with respect to the input data type.

classifiers. The last one is the value of the portfolio when all the machine learning classifiers are used for the prediction. Note that superior prediction accuracy of ADS leads to higher portfolio returns than the other methods.

5. Conclusions

This paper proposes an adaptive method to construct appropriate training data for machine learning based on the

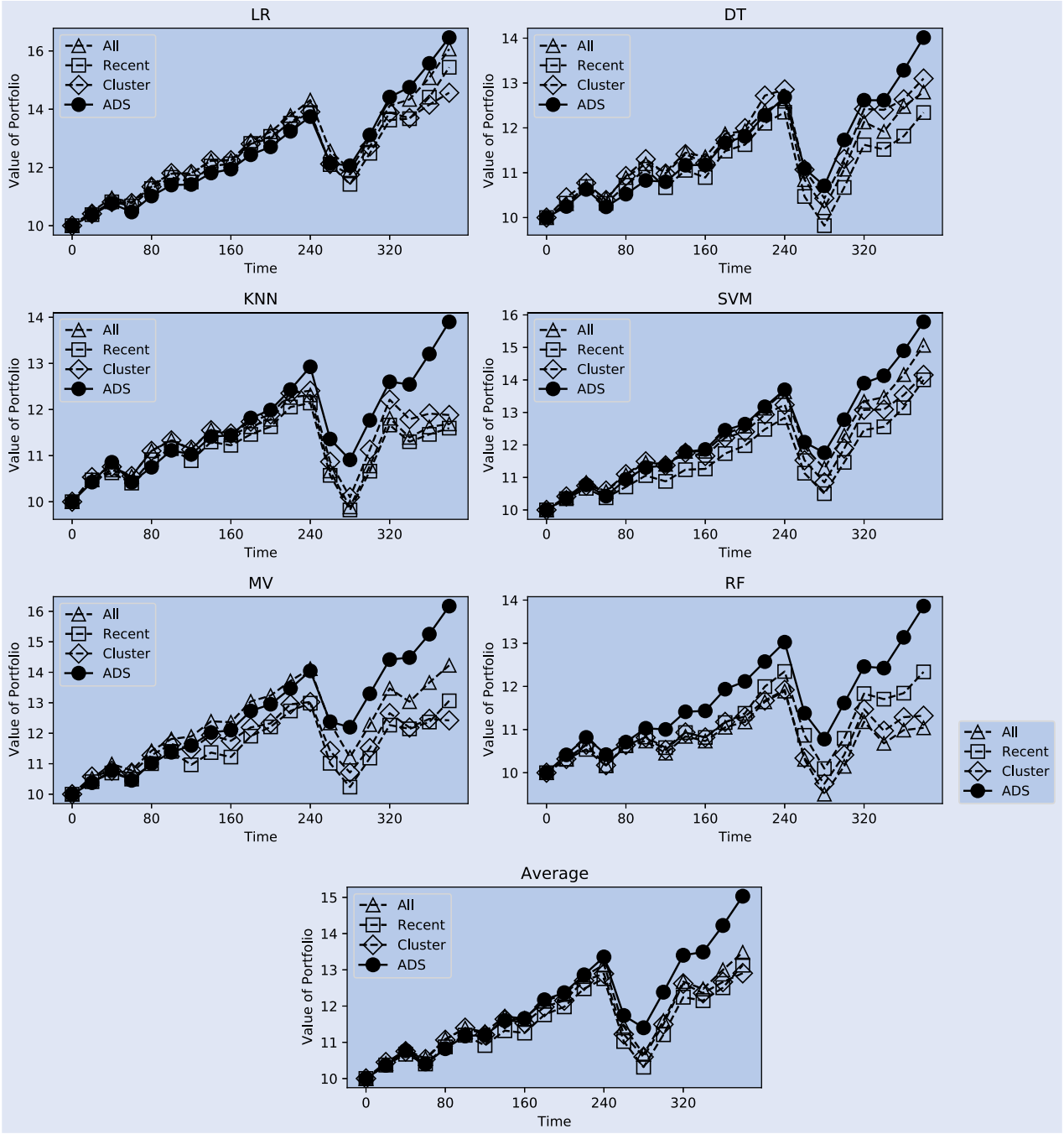

Figure 17. Portfolio values for 400 days for each construction method of the training data from (Top) one of the six machine learning classifiers and (Bottom) all six classifiers.

mathematical distance to the trend of the test data. The proposed construction method improves the prediction accuracy of the machine learning algorithms, which does not depend on the type of machine learning algorithm or data sets. The experiments with various machine learning algorithms on the sectors in S&P 500 for 10 years show that the proposed algorithm is robust, presenting significant improvement in prediction capability. In addition, such superior contribution lead to higher returns in the portfolio management.

Note that the proposed method is not limited to financial time series and can be easily applicable to other time series analysis as well, such as voice recognition, biological evolution, signal processing and pattern recognition. In addition, the distance is measured by the L_2 norm in this study and it will be generalized to Dynamic Time Warping (DTW) in future research.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean government (No. 2021R1F1A1054766 and NRF-2018R1D1A1B07050046).

ORCID

Hongjoong Kim http://orcid.org/0000-0001-9235-0929
Kyoung-Sook Moon http://orcid.org/0000-0003-3023-2358

References

Aggarwal, C.C. and Reddy, C.K., *Data Clustering: Algorithms and Applications*, Data Mining and Knowledge Discovery, 2013 (Chapman and Hall/CRC).

Atsalakis, G.S. and Valavanis, K.P., Surveying stock market forecasting techniques – part II: Soft computing methods. *Expert. Syst. Appl.*, 2009, **36**, 5932–5941.

Ballings, M., Poel, D.V.D., Hespeels, N. and Gryp, R., Evaluating multiple classifiers for stock price direction prediction. *Expert. Syst. Appl.*, 2015, **42**, 7046–7056.

Biau, G. and Devroye, L., *Lectures on the Nearest Neighbor Method*, Data Sciences, 2015 (Springer).

Campbell, J.Y., Lo, A.W.C. and MacKinlay, A.C., *The Econometrics of Financial Markets*, 1997 (Princeton University Press: Princeton, NJ).

Chen, S. and Ge, L., Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction. *Quant. Finance*, 2019, **19**, 1507–1515.

Chen, Y. and Hao, Y., A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert. Syst. Appl.*, 2017, **80**, 340–355.

Fischer, T. and Krauss, C., Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.*, 2018, **270**, 654–669.

Friedman, J.H., Tibshirani, R. and Hastie, T., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2013 (Springer: New York).

Hadavandi, E., Shavandi, H. and Ghanbari, A., Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. *Knowl. Based. Syst.*, 2010, **23**, 800–808.

Harrell, F.E., *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*, 2015 (Springer: New York).

Hsu, S.H. and Hsu, K.C., A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. *Expert. Syst. Appl.*, 2009, **36**, 7947–7951.

Huang, W., Nakamori, Y. and Wang, S., Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.*, 2005, **32**, 2513–2522.

Huang, S.F., Guo, M. and Chen, M.R., Stock market trend prediction using a functional time series approach. *Quant. Finance*, 2020, **20**, 69–79.

James, G., Witten, D., Hastie, T. and Tibshirani, R., *An Introduction to Statistical Learning with Applications in R*, 2013 (Springer: New York).

Jensen, M.C., Some anomalous evidence regarding market efficiency. *J. Financ. Econ.*, 1978, **6**, 95–101.

Kaastra, I. and Boyd, M., Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 1996, **10**, 215–236.

Lai, R.K., Fan, C.Y., Huang, W.H. and Chang, P.C., Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert. Syst. Appl.*, 2009, **36**, 3761–3773.

Law, T. and Shawe-Taylor, J., Practical Bayesian support vector regression for financial time series prediction and market condition change detection. *Quant. Finance*, 2017, **17**, 1403–1416.

Michie, D., Spiegelhalter, D.J. and Taylor, C., *Machine Learning, Neural and Statistical Classification*, Ellis Horwood Series in Artificial Intelligence, 1994 (Prentice Hall).

Moon, K.S., Jun, S. and Kim, H., Speed up of the majority voting ensemble method for the prediction of stock price directions. *Econ. Comput. Econ. Cyb.*, 2018, **52**, 215–228.

Oztekin, A., Kizilaslan, R., Freund, S. and Iseri, A., A data analytic approach to forecasting daily stock returns in an emerging market. *Eur. J. Oper. Res.*, 2016, **253**, 697–710.

Pai, P.F. and Lin, C., A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, 2005, **33**, 497–505.

Patel, J., Shah, S., Thakkar, P. and Kotecha, K., Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert. Syst. Appl.*, 2015, **42**, 259–268.

Pesaran, M.H. and Timmermann, A., A simple nonparametric test of predictive performance. *J. Bus. Econ. Stat.*, 1992, **10**, 461–465.

Qiu, M., Song, Y. and Akagi, F., Application of artificial neural network for the prediction of stock market returns: the case of the Japanese stock market. *Chaos Solitons Fract.*, 2016, **85**, 1–7.

Raschka, S., *Python Machine Learning*, 2015 (Packt Publishing Ltd: Birmingham).

Rokach, L. and Maimon, O.Z., *Data Mining With Decision Trees: Theory and Applications*, Machine Perception and Artificial Intelligence, 2014 (World Scientific Publishing Company).

Scholz, M., Nielsen, J.P. and Sperlich, S., Nonparametric prediction of stock returns based on yearly data: The long-term view. *Insur. Math. Econ.*, 2015, **65**, 143–155.

Schoneburg, E., Stock price prediction using neural networks: A project report. *Neurocomputing*, 1990, **2**, 17–27.

Teixeira, L.A. and Oliveira, A.L., A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert. Syst. Appl.*, 2010, **37**, 6885–6890.

Timmermann, A.G. and Granger, C.W.J., Efficient market hypothesis and forecasting. *Int. J. Forecast.*, 2004, **20**, 15–27.

Vapnik, V.N., *The Nature of Statistical Learning Theory*, 2013 (Springer: New York).

Zhong, X. and Enke, D., A comprehensive cluster and classification mining procedure for daily stock market return forecasting. *Neurocomputing*, 2017, **267**, 152–168.