# Performance Evaluation of an SSL Algorithm for Forecasting the Dow Jones Index Stocks

Ioannis E. Livieris*, Andreas Kanavos*, Gerasimos Vonitsanos†, Niki Kiriakidou‡, Anastasios Vikatos*, Konstantinos Giotopoulos§ and Vassilis Tampakas*

*Department of Computer & Informatics Engineering (DISK Lab),
Technological Educational Institute of Western Greece, Hellas, GR 263-34
Email: livieris@teiwest.gr, kanavos@ceid.upatras.gr, vikatos@teiwest.gr, vtampakas@teimes.gr
†Department of Informatics, Ionian University, Corfu, Hellas
Email: c16voni@ionio.gr
‡Department of Mathematics, University of Patras,
Hellas, GR 265-00. Email: kyriakidou@upatras.gr
§Department of Business Administration,
Technological Educational Institute of Western Greece, Hellas, GR 263-34
Email: kgiotop@ceid.upatras.gr

***Abstract*—Semi-supervised learning algorithms have become a hot topic of research as an alternative to traditional classification methods, exploiting the explicit classification information of labeled data with the knowledge hidden in the unlabeled data for building powerful and effective classifiers. In this work, we evaluate the performance of an ensemble semi-supervised learning algorithm for the prediction of stocks movement in the Dow Jones industrial average. Our experimental results indicate that the proposed algorithm outperforms its component semi-supervised learning algorithms, illustrating that reliable and robust prediction models could be developed utilizing a few labeled and many unlabeled data.**

***Index Terms*—Semi-supervised Learning, Self-labeled Methods, Ensemble Learning, Classification, Voting**

## I. Introduction

One of the most significant decision problems in the financial domain concerns the forecasting of stock return or stock index movement which has attracted researchers' attention for many years. Nowadays, the area of financial market analysis has been dramatically changed from a rather qualitative science to a more quantitative science; this kind of science is also based on knowledge extraction from databases. During the last decades, a large volume of data have been maintained and accumulated from daily activity of different resources by the stock markets.

The growing research and developments in technology constitute in the exponential generation of these data in size, dimension and complexity in the future. Hence, even more companies are interested in extracting knowledge out of them, which enables researchers to analyze them and support or critique policy decisions. Nevertheless, stock financial data have non-linear relationships between inputs and outcomes, hindering their analysis and modeling. As a result the process of analyzing these data and producing an insight into them consists an attractive and challenging task for many stock investors which often require huge efforts [24].

978-1-5386-8161-9/18/$31.00 © 2018 IEEE

The Dow Jones Industrial Average (DJIA) is a price-weighted index of 30 blue-chip U.S. companies representing nine economic sectors including financial service, technology, retail, entertainment and consumer goods. The stocks comprising the DJIA constitute some of the largest and most successful publicly traded stocks in the USA. Moreover, the leadership position of the component stocks in the DJIA tends to result in an extremely high correlation of the DJIA to broader U.S. indexes, such as the S&P 500 index providing additional opportunities. Therefore, the ability of forecasting which stocks will be successful based solely on previous data, even at margins slightly above 50%, could prove to be significantly lucrative, leading to successful stock investments with considerable profit and the development of an efficient portfolio.

Machine learning techniques offer a first step in extracting useful information from financial data and gaining a more meticulous view into the prediction of stock index or commodity price. In addition, these techniques identify the major factors of elements which affect their movement. In this context, a number of rewarding research studies have been carried out leading to the implementation of several machine learning techniques. However, the major proportion of these studies evaluate the efficacy of supervised methods, utilizing only labelled data in order to develop an accurate prediction model.

Semi-Supervised Learning (SSL) are prominent machine learning techniques which attempt to achieve strong generalization by efficiently combining the explicit classification information of labeled data with the information in the unlabeled data [40]. The main issue in SSL is how to efficiently exploit the information hidden in the unlabeled data. In the literature, several approaches have also been proposed, each with a different philosophy related to the link between the distribution of labeled as well as unlabeled data [2], [3], [26], [33], [40].

Self-labeled algorithms are considered as the most popular class of SSL algorithms which address the shortage of labeled data via a self-learning process based on supervised prediction models. The main advantages of these algorithms are their simplicity of implementation as well as their wrapper-based philosophy; therefore they have been successfully applied in a variety of real-world classification problems (see [18]–[21], [38]–[40] and the references therein) providing some interesting results.

In this work, we examine and evaluate the performance of an ensemble SSL algorithm, entitled CST-Voting, for the prediction of stocks' price movement in the DJIA. CST-Voting is based on an ensemble philosophy, combining the predictions of three of the most efficient and frequently used self-labeled algorithms, using a majority voting. The efficacy and efficiency of the presented algorithm is demonstrated by a plethora of experiments and confirmed by the statistical non-parametric tests.

The remainder of this paper is organized as follows: Section II presents a survey of recent studies concerning the application of data mining in stock price prediction. Section III defines the semi-supervised classification problem and presents some of the most popular self-labeled algorithms. Section IV presents a detailed description of the data utilized in our study and the ensemble SSL classification algorithm. Section V presents a series of experiments in order to examine and evaluate the accuracy of proposed algorithm compared with the most popular classification algorithms. Finally, Section VI presents the conclusions and directions for future work.

## II. Related work

The availability of vast amounts of valuable data generated by the stock markets as well as the rapid advances in technology has enabled the development of decision support systems to assist in complex decision making environments. Therefore, during the past decades, researchers began to apply machine learning and data mining techniques and methodologies to develop intelligent systems for forecasting stock movement and price index.

Nevertheless, despite all this effort, there is still no widely utilized accurate prediction method. Along this line, Hajizadeh et al. [10] presented a survey in which they stated that the stock market is a complex, non stationary, chaotic and non-linear dynamic system; thus at present there are no systems which can accurately forecast stock market. To this end, a number of rewarding studies has being carried out, some of which are briefly described in the next paragraphs.

Enke and Thawornwong [7] investigated the predictive power of several financial and economic variables by adopting the variable relevance analysis technique for forecasting stock market returns. They stated that their proposed technique seems attractive in selecting the variables when the usefulness of the data is unknown, especially when non-linearity exists. Furthermore, they evaluated the efficacy of neural network models for level estimation and classification and went on to introduce a cross-validation, early stopping technique which aims to improve the generalization ability of the prediction models. Finally, their results showed that the trading strategies guided by the neural network classification models generate higher profits under the same risk exposure than those suggested by the other strategies. One has to consider that in their results, the buy-and-hold strategy, as well as the level estimation forecasts of neural network and linear regression models are included.

Senthamarai Kannan et al. [31] evaluated various data mining techniques to predict stock movement. Their proposed method is based on the combination of five algorithms and generates a prediction of whether stock prices will either go up or down in the following day. The authors performed an experimental analysis indicating that their method was able to foresee if the following day's closing price would increase or decrease better than chance ($50\%$) with a high level of significance. Moreover, they stated that their proposed method could be used as a buying or selling decision support system or it could be used to give confidence to a trader's prediction of stock prices.

Nanda et al. [23] presented a methodology for integrating a variety of clustering techniques into portfolio management and build a hybrid system of getting efficient portfolios. All clustering methods were utilized to collect financial stock data from Bombay Stock Exchange which consists of returns for variable period lengths along with the valuation ratios. Their results showed that their recommended technique can considerably reduce a lot of time in the selection of stocks since stocks of similar categories can be easily grouped into a cluster; therefore the best performing stocks from those groups can be selected.

Another work is presented by Patel el al. [27], where authors address the problem of predicting direction of movement of stock and stock price index for the Indian stock markets. They evaluated the performance of various machine learning algorithms utilizing two approaches for the input data. In particular, the first approach involves the computation of ten technical parameters using stock trading data while the second one focuses on representing these technical parameters as trend deterministic data. Their extensive experimental analysis revealed that the performance of all prediction models was improved when these technical parameters are represented as trend deterministic data.

Ng and Khor [24] introduced a stock profiling framework, named StockProF, which can assist investors to build a stock portfolio based on their investment strategies. StockProF detects outliers from a pool of stocks using an outlier detection algorithm in order to identify stocks with superior or poor financial performance. Moreover, it utilizes a clustering algorithm for grouping the remaining stocks enabling the identification of stocks with various financial performances. They utilized 1-year stock price movements to evaluate the performance of the outliers as well as the clusters and their results showed that StockProF is effective as the profiling corresponded to the average capital gain or loss of the plantation stocks.

In more recent works, Kia et al. [13] proposed a hybrid supervised semi-supervised model, called HyS3, for predicting daily direction of movement for everyday in markets across the globe. The graph-based semi-supervised part of HyS3 models the markets global interactions through a network designed with a novel continuous Kruskal-based graph construction algorithm. Furthermore, the supervised part of the model injects results extracted from each market's historical data to the network whenever the hybrid model allows with an innovative conditional mechanism. Based on their numerical experiments the authors concluded that their suggested model utilizing historical market data for each market along with data from other global markets could provide higher accuracy than other existing prediction models.

## III. A Review of Self-labeled Algorithms

This section provides the formal definition and the necessary notations for the semi-supervised classification problem and briefly describes the most popular self-labeled algorithms in the literature.

### A. Semi-supervised classification

Let $(x, y)$ be an example, where $x$ belongs to a class $y$ and a $D$-dimensional space in which $x_i$ is the $i$-th attribute of the instance. Suppose that the training set $L \cup U$ consists of a labeled set $L$ of $N_L$ instances where $y$ is known and of an unlabeled set $U$ of $N_U$ instances where $y$ is unknown with $N_L \ll N_U$. Furthermore, there exists a test set $T$ of $N_T$ unseen instances where $y$ is unknown which has not been utilized in the training stage. Notice that the aim of the semi-supervised classification is to obtain an accurate and robust learn hypothesis with the use of the training set.

Self-labeled techniques are considered a significant family of classification methods which progressively classify unlabeled data based on the most confident predictions without making any specific assumptions about the input data [33].

In the literature, a variety of self-labeled methods has been proposed each following a different methodology on exploiting the information hidden in the unlabeled data.

### B. Self-labeled Algorithms

*Self-training* probably constitutes the most popular and frequently utilized SSL algorithm, due to its simplicity of implementation and good classification accuracy. According to Ng and Cardie [25] "*self-training is a single-view weakly supervised algorithm*" which is based on its own predictions on unlabeled data to teach itself. This algorithm wraps around a base learner and utilizes its own predictions to assign labels to unlabeled data.

More analytically, in the self-training process, a classifier is initially trained with a small number of labeled examples and at each iteration its training set is augmented gradually with classified unlabeled instances that have achieved a probability value over a defined threshold $c$; these instances are considered sufficiently reliable to be added to the training set. Observe that the way in which the confidence prognostics are measured, is dependant on the type of utilized base learner (see [34]). Nevertheless, this proposed methodology can lead to erroneous predictions in case noisy examples are characterized as confident and can later be incorporated into the labeled training set [40].

Li and Zhou [16] tried to address this difficulty and as a result, they presented the *SETRED* method, which incorporates data editing in the self-training framework in order to learn actively from the self-labeled examples. Their principal improvement in relation to the classical self-training scheme is the establishment of a restriction related to the acceptance or the rejection of the unlabeled examples which the algorithm evaluates as trustworthy. More analytically, a neighboring graph in $D$-dimensional feature space is being built and all the candidate unlabeled examples for being appended to the initial training set are being filtered through a hypothesis test. As a result, any examples which successfully passed that test are finally added to the training set before the end of each iteration.

*Co-training* is a semi-supervised algorithm which can be regarded as a different variant of self-training technique [3]. It is based on the strong assumption that the feature space can be divided in two conditionally independent views, each view being sufficient to train an efficient classifier. In this framework, two learning algorithms are trained separately for each view using the initial labeled dataset and the most confident predictions of each algorithm on unlabeled data are used to augment the training set of the other through an iterative learning process. Following the same concept, Nigam and Gani [26] performed an experimental analysis and concluded that the Co-training outperforms other SSL algorithms when there is a natural existence of two distinct and independent views. Nevertheless, the assumption about the existence of sufficient and redundant views is a luxury hardly met in most real-case scenarios.

Ensemble classifiers can also be used under a semi-supervised scheme for exploiting the power of more than one weak learners [35]. The efficiency of an ensemble method is heavily dependent from the accuracy and diversity of the included classifiers thus, different artificial tactics have been presented for injecting miscellaneousness to a group of classifiers when the original variety does not reach the expected levels [29].

Zhou and Goldman [37] have also adopted the idea of ensemble learning and majority voting and proposed *Democratic co-learning* algorithm which also follows the multiview theory but from another aspect. More specifically, this algorithm utilizes multiple algorithms for producing the necessary information and endorses a voted majority process for the final decision, instead of asking for more than one views of the corresponding data.

Along this line, Li and Zhou [17] proposed another algorithm, in which a number of Random trees are trained on bootstrap data from the dataset, named *Co-Forest*. The main idea of this algorithm is the assignment of a few unlabeled examples to each Random tree during the training process.

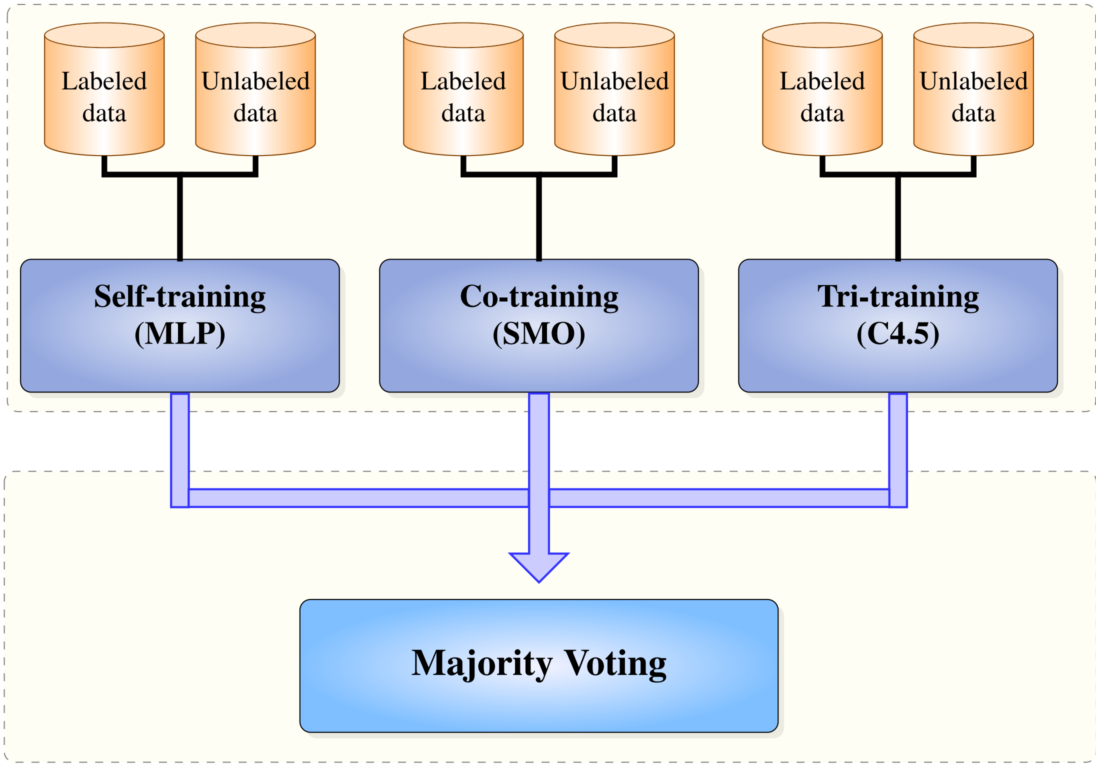


Fig. 1. CST-Voting

Eventually, the final decision is composed by a simple majority voting. Notice that the utilization of Random Tree classifier for random samples of the collected labeled data is the main reason why the behavior Co-Forest is efficient and robust although the number of the available labeled examples is reduced.

A rather representative approach which is based on the ensemble philosophy is the *Tri-training* algorithm. This corresponding algorithm constitutes an improved single-view extension of the Co-training algorithm exploiting unlabeled data without relying on the existence of two views of instances [38]. Tri-training algorithm can be considered as a bagging ensemble of three classifiers which are trained on data subsets generated through bootstrap sampling from the original labeled training set [9]. Subsequently, in each Tri-training round, if two classifiers agree on the labeling of an unlabeled instance while the third one disagrees, then these two classifiers will label this instance for the third classifier. It is worth noticing that the "majority teach minority strategy" serves as an implicit confidence measurement, which avoids the use of complicated time-consuming approaches to explicitly measure the predictive confidence, and hence the training process is efficient [20].

## IV. Methodology

The main goal of the research described in this paper is the development of a prediction model so as to forecast the stocks in the DJIA which will have a net value gain or loss. For this purpose, we adopted a two-stages methodology, where the first stage deploys the proposed semi-trained two-level classification algorithm while the second one concerns dataset utilized in this study.

### A. CST-Voting

Subsequently, we present a brief description of the proposed ensemble SSL algorithm, entitled CST-Voting [14] for the classification of stock price movement.

CST-Voting is based on the idea of generating an ensemble of classifiers by applying different SSL algorithms with a different philosophy and methodology on exploiting the unlabeled data. On this basis, the learning algorithms, which constitute the proposed ensemble are: Co-training, Self-training and Tri-training. These three SSL algorithms are self-labeled algorithms which differ in the mechanism utilized to label unlabeled data. It is worth mentioning that Self-training and Tri-training are single-view methods while Co-training is a multi-view method. Moreover, Self-training uses a single classifier while Co-training and Tri-training use multiple classifiers.

Initially, the classical semi-supervised algorithms, which constitute the ensemble, i.e. Self-training, Co-training and Tri-training, are trained utilizing the same labeled $L$ and unlabeled dataset $U$. In the sequel, the final hypothesis on an example of the test set is based on the individual predictions of the SSL algorithms, utilizing a simple majority voting methodology.

Hence, the output of the ensemble is the one made by at least two of the SSL algorithms. An overview of CST-Voting is depicted in Figure 1.

### B. Dow Jones Index Dataset

The Dow Jones index dataset includes 750 instances from the UCI Machine Learning Repository [22], concerning weekly metrics of every DJIA stock in the first and second financial quarters of 2011.

Table I presents a set of the ten (10) specific attributes utilized in our study. The first four (4) attributes concern the price of the stock at the beginning and at the end of the week and the highest and the lowest price of the stock during the week. The following four (4) attributes concern the number of shares of stock that traded hands in the week, the percentage change in price throughout the week, the percentage change in the number of shares of stock that traded hands for this week compared to the previous week and the number of shares of stock that traded hands in the previous week. The last two (2) attributes are related with the number of days until the next dividend and the percentage of return on the next dividend. Finally, the stocks in the DJIA were classified based on the net value gain or loss over the following week. Notice that the features that reported next week's opening and closing price contained in the original data were removed since these attributes would not be known in real life when attempting to predict stock behavior. The dataset was partitioned into two sets (training/testing) based on financial quarter, thus the 360 examples of the first quarter were utilized as training set and the rest 390 of the second quarter were utilized as testing set as in [4].

| Attribute | Values |
|---|---|
| Opening price | Nominal |
| Closing price | Numeric |
| Weekly high price | Numeric |
| Weekly low price | Numeric |
| Volume | Numeric |
| Percent change in price | Numeric |
| Percent change in volume | Numeric |
| Previous week's volume | Numeric |
| Days to next dividend | Numeric |
| Percent return on next dividend | Numeric |
| Class | {Gain, Loss} |

TABLE I
DATASET DESCRIPTION

## V. NUMERICAL EXPERIMENTS

In this section, we exhibit an evaluation of the classification performance of CST-Voting algorithm using the Dow Jones index dataset. The experiments in our study take place in two distinct parts. In the first part, we evaluate the performance of CST-Voting against Self-training, Co-training and Tri-training; while in the second part we compare the classification performance of CST-Voting against the most popular self-labeled algorithms, namely Self-training with editing (SETRED), Co-Forest and Democratic Co-learning (Demo-Co) against classical supervised algorithms.

All SSL algorithms were evaluated by deploying as base learners the Naive Bayes (NB) [6], the Multilayer Perceptron (MLP) [30], the Sequential Minimum Optimization (SMO) [28], the 3NN algorithm [1], the rule-learning algorithm RIPPER (JRip) [5] and the decision tree algorithm Logistic Model Tree (LMT) [15]. These algorithms probably constitute the most effective and popular machine learning algorithms for classification problems [36].

The implementation code was written in JAVA, making use of the WEKA Machine Learning Toolkit [11]. The configuration parameters for all SSL algorithms, utilized in our experiments, are presented in following Table II. Moreover, in order to minimize the effect of any expert bias, instead of attempting to tune any of the algorithms to the specific datasets, all base learners were used with their default parameter settings included in the WEKA software. Similar to Blum and Mitchell [3], a limit to the number of iterations of all SSL algorithms is established. This implementation strategy has also been adopted by many researchers [18]–[21], [32]–[34]. In order to study the influence of the amount of labeled data, three different ratios ($R$) of the training data were used, i.e. $10\%$, $20\%$ and $30\%$.

| SSL algorithm | Parameters |
|---|---|
| Self-training | MaxIter $= 40$.<br>$c = 0.95$. |
| Co-training | MaxIter $= 40$.<br>Initial unlabeled pool $= 75$. |
| Tri-training | No parameters specified. |
| Democratic-Co learning | Classifiers = 3NN, C4.5, NB. |
| SETRED | MaxIter $= 40$.<br>Threshold $= 0.1$. |
| Co-Forest | Number of Random Forest classifiers $= 6$.<br>Threshold $= 0.75$. |

TABLE II
PARAMETER SPECIFICATION FOR ALL THE SSL METHODS EMPLOYED IN OUR EXPERIMENTS

To evaluate the performance of the classification algorithms, the following three performance metrics are considered, namely Sensitivity (*Sen*), Specificity (*Spe*) and Accuracy (*Acc*):

$$Sen = \frac{T_P}{T_P + F_N}$$

$$Spe = \frac{T_N}{T_N + F_P}$$

$$Acc = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

where $T_P$ stands for the number of stocks which were correctly identified to have a net value gain, $T_N$ stands for the number of stocks which were correctly identified to have a net value loss, $F_P$ (type $I$ error) stands for the number of stocks which were misidentified to have a net value gain and $F_N$ (type $II$ error) stands for the number of stocks which misidentified to have a net value loss.

It is worth mentioning that Sensitivity of classification is the proportion of actual positives that are predicted as positive; Specificity represents the proportion of actual negatives that are predicted as negative, while Accuracy is the ratio of correct predictions of a classification model.

### A. First Part

In the sequel, we focus our interest on the experimental analysis for evaluating the classification performance of CST-Voting algorithm against Self-training, Co-training and Tri-training.

Tables III, IV and V present the accuracy of each compared SSL algorithm, relative to the performance metrics *Sen*, *Spe* and *Acc*, respectively. Notice that the highest classification accuracy is highlighted in bold. Clearly, CST-Voting exhibits the best performance regarding the two out of three performance metrics *Sen* and *Acc*, reporting the highest classification accuracy in almost all cases. Relative to the *Spe* performance metric, Self-training and Co-training present the best average performance, slightly outperforming CST-Voting and Tri-training. Moreover, it is worth mentioning that CST-Voting, in contrast to the rest SSL algorithms, improves its classification accuracy as the labeled ratio increases.

| Classifier | Ratio | Self-Train | Co-Train | Tri-Train | CST-Voting |
|---|---|---|---|---|---|
| NB | 10% | 61.4% | **72.3%** | 70.7% | 71.2% |
| | 20% | 71.7% | **72.3%** | 66.8% | **72.3%** |
| | 30% | 66.3% | 72.3% | **72.8%** | 71.7% |
| SMO | 10% | 61.4% | 58.2% | 64.1% | **65.8%** |
| | 20% | 71.2% | 75.5% | 74.5% | **78.8%** |
| | 30% | 71.7% | 78.3% | 79.9% | **81.0%** |
| MLP | 10% | 77.2% | 66.3% | 73.4% | **78.8%** |
| | 20% | 76.1% | 72.3% | 69.6% | **78.8%** |
| | 30% | 77.2% | 72.8% | 75.0% | **85.9%** |
| 3NN | 10% | 56.0% | 55.4% | 53.3% | **58.7%** |
| | 20% | 51.6% | 52.2% | 53.3% | **56.5%** |
| | 30% | 55.1% | 54.3% | 53.4% | **60.3%** |
| JRip | 10% | 74.5% | 74.5% | 75.0% | **77.2%** |
| | 20% | 78.3% | 81.0% | 79.3% | **81.5%** |
| | 30% | 70.7% | 79.9% | 78.8% | **83.7%** |
| LMT | 10% | 78.8% | 63.6% | 73.4% | **81.5%** |
| | 20% | 80.4% | 74.5% | 82.1% | **84.2%** |
| | 30% | 82.6% | 77.2% | **83.7%** | 83.2% |

TABLE III
ACCURACY OF THE SSL ALGORITHMS BASED ON $Sen$ PERFORMANCE METRIC FOR EACH LABELED RATIO

In machine learning, the statistical comparison of multiple algorithms over multiple datasets is fundamental and it is usually carried out by means of a statistical test [20]. Therefore, we utilize Friedman Aligned-Ranks (FAR) test [12] in order to conduct a complete performance comparison between all corresponding algorithms for all the different labeled ratios. Its application will allow us to highlight the existence of significant differences between the evaluated SSL algorithms and estimate the rejection of the hypothesis that all the classifiers perform equally well for a given level. Notice that FAR test is considered to be one of the most well-known

| Classifier | Ratio | Self-Train | Co-Train | Tri-Train | CST-Voting |
|---|---|---|---|---|---|
| NB | 10% | **45.7%** | 37.0% | 38.0% | 38.6% |
| | 20% | 35.3% | 37.0% | **38.6%** | 38.0% |
| | 30% | **41.3%** | 37.0% | 37.0% | 39.7% |
| SMO | 10% | **32.6%** | 21.7% | 27.2% | 31.0% |
| | 20% | 35.3% | 22.8% | **32.1%** | 28.8% |
| | 30% | **37.0%** | 36.4% | 34.8% | 33.7% |
| MLP | 10% | 24.5% | **32.1%** | 27.7% | 27.2% |
| | 20% | 24.5% | **31.0%** | **31.0%** | 28.3% |
| | 30% | 22.3% | **31.0%** | 29.3% | 29.3% |
| 3NN | 10% | 47.3% | 51.6% | **58.7%** | 54.3% |
| | 20% | 51.1% | 51.1% | **58.7%** | 58.2% |
| | 30% | **63.6%** | 54.3% | 58.2% | 58.7% |
| JRip | 10% | 21.2% | **26.6%** | 21.7% | 22.8% |
| | 20% | **29.3%** | 27.2% | 23.9% | 26.1% |
| | 30% | **31.0%** | 28.8% | 25.0% | 27.2% |
| LMT | 10% | 17.9% | **22.8%** | **22.8%** | 21.7% |
| | 20% | 21.2% | 19.0% | **25.5%** | 23.9% |
| | 30% | 22.8% | 25.0% | 22.8% | **26.1%** |

TABLE IV
ACCURACY OF THE SSL ALGORITHMS BASED ON $Spe$ PERFORMANCE METRIC FOR EACH LABELED RATIO

| Classifier | Ratio | Self-Train | Co-Train | Tri-Train | CST-Voting |
|---|---|---|---|---|---|
| NB | 10% | 50.5% | 51.5% | 51.3% | **51.8%** |
| | 20% | 50.5% | 51.5% | 49.7% | **52.1%** |
| | 30% | 50.8% | 51.5% | 51.8% | **52.6%** |
| SMO | 10% | 44.4% | 37.7% | 43.1% | **45.6%** |
| | 20% | 50.3% | 46.4% | 50.3% | **50.8%** |
| | 30% | 51.3% | 54.1% | 54.1% | **54.1%** |
| MLP | 10% | 47.9% | 46.4% | 47.7% | **50.0%** |
| | 20% | 47.4% | 48.7% | 47.4% | **50.5%** |
| | 30% | 46.9% | 49.0% | 49.2% | **54.4%** |
| 3NN | 10% | 48.7% | 50.5% | 52.8% | **53.3%** |
| | 20% | 48.5% | 48.7% | 52.8% | **54.1%** |
| | 30% | 51.3% | 51.3% | 52.6% | **56.2%** |
| JRip | 10% | 45.1% | **47.7%** | 45.6% | 47.2% |
| | 20% | **50.8%** | 51.0% | 48.7% | **50.8%** |
| | 30% | 47.9% | 51.3% | 49.0% | **52.3%** |
| LMT | 10% | 45.6% | 40.8% | 45.4% | **48.7%** |
| | 20% | 47.9% | 44.1% | 50.8% | **51.0%** |
| | 30% | 49.7% | 48.2% | 50.3% | **51.5%** |

TABLE V
ACCURACY OF THE SSL ALGORITHMS BASED ON $Acc$ PERFORMANCE METRIC FOR EACH LABELED RATIO

tools for multiple statistical comparison tests when comparing more than two methods [8]. Furthermore, the Finner test is applied as a post hoc procedure to find out which algorithms present significant differences.

Tables VI, VII and VIII present the information of the statistical analysis performed by nonparametric multiple comparison procedures over $10\%$, $20\%$ and $30\%$ of labeled data respectively. The best (e.g. lowest) ranking obtained in each FAR test determines the control algorithm for the post hoc test. CST-Voting demonstrates the best overall performance, as it outperforms the rest SSL algorithms. This is due to the fact that it reports the highest probability-based ranking by statistically presenting better results, relative to all labeled ratio.

| SSL Algorithm | Aligned Friedman Ranking | Finner post-hoc test $p$-value | Finner post-hoc test Null Hypothesis |
|---|---|---|---|
| CST-Voting | 4.6667 | - | - |
| Tri-training | 12.9167 | 0.043297 | rejected |
| Self-training | 14.75 | 0.020204 | rejected |
| Co-training | 17.6667 | 0.004346 | rejected |

TABLE VI
FAR TEST AND FINNER POST HOC TEST (LABELED RATIO $10\%$)

| SSL Algorithm | Aligned Friedman Ranking | Finner post-hoc test $p$-value | Finner post-hoc test Null Hypothesis |
|---|---|---|---|
| CST-Voting | 5.4167 | - | - |
| Tri-training | 12.1667 | 0.098248 | accepted |
| Self-training | 15.25 | 0.023920 | rejected |
| Co-training | 17.1667 | 0.011952 | rejected |

TABLE VII
FAR TEST AND FINNER POST HOC TEST (LABELED RATIO $20\%$)

| SSL Algorithm | Aligned Friedman Ranking | Finner post-hoc test $p$-value | Finner post-hoc test Null Hypothesis |
|---|---|---|---|
| CST-Voting | 4 | - | - |
| Tri-training | 12.6667 | 0.033763 | rejected |
| Co-training | 13.75 | 0.025285 | rejected |
| Self-training | 19.5833 | 0.000405 | rejected |

TABLE VIII
FAR TEST AND FINNER POST HOC TEST (LABELED RATIO $30\%$)

### *B. Second Part*

Subsequently, we evaluate the classification performance of the CST-Voting algorithm against some other state-of-the-art self-labeled algorithms such as SETRED, Co-Forest and Democratic-Co learning [33]. Notice that CST-Voting utilizes $NB$ and $LMT$ as base learners, which exhibited the highest accuracy, relative to the performance metrics.

Table IX presents the classification accuracy of each tested algorithm using $10\%$, $20\%$ and $30\%$ as labeled ratio. As above mentioned, the accuracy measure of the best performing algorithm is highlighted in bold. CST-Voting illustrated the best performance independent of the utilized labeled ratio. More analytically, CST-Voting exhibited the outstanding results relative to the performance metric *Spe*, using $NB$ as base learner. Furthermore, CST-Voting reported the highest classification accuracy using $3NN$ as base learner, regarding the performance metrics *Sen* and *Acc*.

| | $\mathbf{R = 10\%}$ | | | $\mathbf{R = 20\%}$ | | | $\mathbf{R = 30\%}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Sen** | **Spe** | **Acc** | **Sen** | **Spe** | **Acc** | **Sen** | **Spe** | **Acc** |
| SETREG | 76.6% | 24.5% | 47.7% | 75.0% | 25.5% | 47.4% | 77.2% | 26.1% | 48.7% |
| Co-Forest | 76.6% | 22.3% | 46.7% | 67.4% | 34.2% | 47.9% | 66.3% | 38.6% | 49.5% |
| Demo-Co | 65.2% | 34.2% | 46.9% | 69.0% | 34.8% | 49.0% | 70.7% | 38.0% | 51.3% |
| CST (NB) | 58.7% | **54.3%** | **53.3%** | 56.5% | **58.2%** | **54.1%** | 60.3% | **58.7%** | **56.2%** |
| CST (3NN) | **81.5%** | 21.7% | 48.7% | **84.2%** | 23.9% | 51.0% | **83.2%** | 26.1% | 51.5% |

TABLE IX
CLASSIFICATION ACCURACY OF SSL ALGORITHM FOR EACH LABELED RATION

Finally, in order to illustrate the classification performance of the CST-Voting, we evaluate its performance for each base learner using $R = 30\%$ of the training set against the corresponding supervised algorithms trained with $100\%$ of the training set. The results presented in Table X illustrate that CST-Voting is comparatively better than the respective supervised algorithms, regarding all base learners that were used.

| | **Supervised** | | | **CST-Voting** | | |
|---|---|---|---|---|---|---|
| | **Sen** | **Spe** | **Acc** | **Sen** | **Spe** | **Acc** |
| NB | 60.3% | 44.2% | 51.8% | 71.7% | 39.7% | 52.6% |
| MLP | 75.7% | 25.7% | 49.3% | 81.0% | 33.7% | 54.1% |
| SMO | 79.9% | 19.9% | 48.2% | 85.9% | 29.3% | 54.4% |
| 3NN | 56.3% | 47.8% | 52.3% | 60.3% | 58.7% | 56.2% |
| JRip | 66.8% | 31.6% | 48.2% | 83.7% | 27.2% | 52.3% |
| LMT | 78.9% | 20.0% | 47.7% | 83.2% | 26.1% | 51.5% |

TABLE X
PERFORMANCE COMPARISON OF CST-VOTING WITH CLASSICAL SUPERVISED ALGORITHMS

## VI. CONCLUSIONS AND FUTURE WORK

In this work, we evaluated the performance of an ensemble SSL algorithm, entitled CST-Voting, for the prediction of stocks in the DJIA which will have a net value gain or loss. The proposed CST-Voting combines the individual predictions of three of the most efficient and popular SSL algorithms, i.e. Co-training, Self-training and Tri-training, utilizing a simple voting methodology. The efficacy of CST-Voting was illustrated by a plethora of experiments and confirmed by the Friedman Aligned Ranks nonparametric test as well as the Finner post hoc test.

Our future work is concentrated on enlarging our experiments and on applying further CST-Voting on several financial datasets. Another interesting aspect is the development of a parallel implementation of CST-Voting since with the implementation of each component based learners in parallel machines, a huge amount of data can be processed in significantly less computational time.

## References


[1] D. Aha. *Lazy Learning*. Dordrecht: Kluwer Academic Publishers, 1997.

[2] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *8th International Conference on Machine Learning (ICML 2001)*, pages 19–26, 2001.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[4] M. Brown, M. Pelosi, and H. Dirska. Dynamic-radius species-conserving genetic algorithm for the financial forecasting of Dow Jones index stocks. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 27–41. Springer, 2013.

[5] W. Cohen. Fast effective rule induction. In *International Conference on Machine Learning*, pages 115–123, 1995.

[6] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[7] D. Enke and S. Thawornwong. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29:927–940, 2005.

[8] S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064, 2010.

[9] M. F. A. Hady and F. Schwenker. Combining committee-based semi-supervised learning and active learning. *Journal of Computer Science and Technology*, 25(4):681–698, 2010.

[10] E. Hajizadeh, H. D. Ardakani, and J. Shahrabi. Application of data mining techniques in stock markets: A survey. *Journal of Economics and International Finance*, 2(7):109–118, 2010.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: An update. *SIGKDD Explorations Newsletters*, 11:10–18, 2009.

[12] J. Hodges and E. Lehmann. Rank methods for combination of independent experiments in analysis of variance. *The Annals of Mathematical Statistics*, 33(2):482–497, 1962.

[13] A. N. Kia, S. Haratizadeh, and S. B. Shouraki. A hybrid supervised semi-supervised graph-based model to predict one-day ahead movement of global stock markets and commodity prices. *Expert Systems with Applications*, 105:159–173, 2018.

[14] G. Kostopoulos, I. E. Livieris, S. Kotsiantis, and V. Tampakas. CST-Voting - A semi-supervised ensemble method for classification problems. *journal of Intelligent & Fuzzy Systems*, 2018.

[15] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. *Machine learning*, 59(1-2):161–205, 2005.

[16] M. Li and Z. Zhou. SETRED: Self-training with editing. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 611–621. Springer, 2005.

[17] M. Li and Z. Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):1088–1098, 2007.

[18] C. Liu and P. Yuen. A boosted co-training algorithm for human action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(9):1203–1213, 2011.

[19] I. E. Livieris, K. Drakopoulou, V. Tampakas, T. Mikropoulos, and P. Pintelas. An ensemble-based semi-supervised approach for predicting students' performance. In *Research on e-Learning and ICT in Education*. 2018.

[20] I. E. Livieris, K. Drakopoulou, V. Tampakas, T. Mikropoulos, and P. Pintelas. Predicting secondary school students' performance utilizing a semi-supervised learning approach. *Journal of Educational Computing Research*, 2018.

[21] I. E. Livieris, A. Kanavos, V. Tampakas, and P. Pintelas. An ensemble ssl algorithm for efficient chest x-ray image classification. *Journal of Imaging*, 4(7), 2018.

[22] P. Murphy and D. Aha. UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, 1994.

[23] S. Nanda, B. Mahanty, and M. T. *. Clustering indian stock market data for portfolio management. *Expert Systems with Applications*, 37:8793–8798, 2010.

[24] K. H. Ng and K. Khor. StockProF: A stock profiling framework using data mining approaches. *Information Systems E-Business Management*, 15(1):139–158, 2016.

[25] V. Ng and C. Cardie. Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 94–101, 2003.

[26] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *ACM International Conference on Information and Knowledge Management*, pages 86–93, 2000.

[27] J. Patel, S. Shah, P. Thakkar, and K. Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, 42(1):259–268, 2014.

[28] J. Platt. Using sparseness and analytic QP to speed training of support vector machines. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in neural information processing systems*, pages 557–563, 1999.

[29] L. Rokach. *Pattern classification using ensemble methods*. World Scientific Publishing Company, 2010.

[30] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362, 1986.

[31] K. Senthamarai Kannan, P. Sailapathi Sekar, M. Mohamed Sathik, and P. Arumugam. Financial stock market forecast using data mining techniques. In *Proceedings of the International Multi-conference of Engineers and Computer Scientists*, volume 1–5, pages 991–994, 2010.

[32] J. Tanha, M. van Someren, and H. Afsarmanesh. Semi-supervised selftraining for decision tree classifiers. *International journal of Machine Learning Cybernetics*, 8:355–370, 2015.

[33] I. Triguero, S. García, and F. Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2):245–284, 2015.

[34] I. Triguero, J. Sáez, J. Luengo, S. García, and F. Herrera. On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, 132:30–41, 2014.

[35] S. Wang, L. Wu, L. Jiao, and H. Liu. Improve the performance of co-training by committee with refinement of class probability estimations. *Neurocomputing*, 136:30–40, 2014.

[36] X. Wu, V. Kumar, J. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z. Zhou, M. Steinbach, D. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[37] Y. Zhou and S. Goldman. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 594–602. IEEE, 2004.

[38] Z. Zhou and M. Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[39] X. Zhu. Semi-supervised learning. In *Encyclopedia of Machine Learning*, pages 892–897. Springer, 2011.

[40] X. Zhu and A. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009.