Information Fusion 36 (2017) 90–102

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Fusion of multiple diverse predictors in stock market

Sasan Barak[a,b,*], Azadeh Arjmand[c], Sergio Ortobelli[a,b]

[a] *Faculty of Economics, VŠB Technical University of Ostrava, Ostrava 70200, Czech Republic*
[b] *Department of SAEQM, University of Bergamo, Via dei Caniana 2, Bergamo 24127, Italy*
[c] *Faculty of Industrial Engineering, Alzahra University, Tehran, Iran*

A R T I C L E   I N F O

Article history:
Received 3 May 2016
Revised 18 September 2016
Accepted 3 November 2016
Available online 9 November 2016

Keywords:
Classifier fusion
Diversity creation
Machine learning
Fundamental analysis
Stock returns prediction
Risk prediction

A B S T R A C T

Forecasting stock returns and their risk represents one of the most important concerns of market decision makers. Although many studies have examined single classifiers of stock returns and risk methods, fusion methods, which have only recently emerged, require further study in this area. The main aim of this paper is to propose a fusion model based on the use of multiple diverse base classifiers that operate on a common input and a Meta classifier that learns from base classifiers' outputs to obtain more precise stock return and risk predictions. A set of diversity methods, including Bagging, Boosting and AdaBoost, is applied to create diversity in classifier combinations. Moreover, the number and procedure for selecting base classifiers for fusion schemes is determined using a methodology based on dataset clustering and candidate classifiers' accuracy. The results demonstrate that Bagging exhibited superior performance within the fusion scheme and could achieve a maximum of 83.6% accuracy with Decision Tree, LAD Tree and Rep Tree for return prediction and 88.2% accuracy with BF Tree, DTNB and LAD Tree in risk prediction. For feature selection part, a wrapper-GA algorithm is developed and compared with the fusion model. This paper seeks to help researcher select the best individual classifiers and fuse the proper scheme in stock market prediction. To illustrate the approach, we apply it to Tehran Stock Exchange (TSE) data for the period from 2002 to 2012.

1. Introduction

The future status of companies offering stocks is of great importance to stock market practitioners. According to the efficient market theory, it is impossible to predict prices based on historical stock data. This theory also states that the prediction of the classical criteria of risk and return cannot bring advantages to shareholders. There is abundant evidence in the literature, however, that argues against the efficient nature of the market [1]. A precise prediction of companies' future financial status provides investors with the security to make a confident and profitable investment.

To achieve an accurate stock market prediction, the identification of the effective features is crucial. In other words, the representative features of the factors play a key role in prediction efficiency. Technical and fundamental analyses are two essential tools in financial market evaluation. Fundamental analysis can be used to evaluate a firm's performance and financial status over a period of time by carefully analysing the institute's financial statement [2]. Technical analysis (TA), conversely, evaluates securities by means of statistics such as past price and volume that are generated by market activities [3]. The major criticism of TA is that it only considers the price movement and ignores the fundamental factors related to the company. Moreover, TA takes a comparatively short-term approach to analysing the market.

Fundamental analysis seeks to find the essential features of stock and market movements. In fact, the logic behind fundamental analysis is that if a company has a proper fundamental strength, then long term stock investment in the company will be more secure and stable. Thus, the stocks of these fundamentally strong companies, which are making money, gaining profit and growing their businesses, represent an opportunity for a successful investment. For this reason, in this paper, fundamental analysis is applied in order to determine the fundamental features that decide which company is a good bet for a secure investment.

Stock return forecasting is a fascinating endeavour with a long history. From the standpoint of finance practitioners, asset allocation requires real-time forecasts of stock returns and an improved stock return forecast holds the promise of enhancing the investment performance [4]. Many studies address the prediction of stock market returns (see, e.g., [5–7]). For an efficient investment, the return consideration is not sufficient. In fact, the risk and

* Corresponding author at: Faculty of Economics, VŠB Technical University of Ostrava, Sokolská třída 33, Ostrava 70200, Czech Republic.

E-mail addresses: sasan.barak@gmail.com, sasan.barak@vsb.cz (S. Barak), A.arjmand@alzahra.ac.ir (A. Arjmand), sergio.ortobelli@unibg.it (S. Ortobelli).

http://dx.doi.org/10.1016/j.inffus.2016.11.006
1566-2535/© 2016 Elsevier B.V. All rights reserved.

return must be considered simultaneously to create an accurate portfolio evaluation [8]. In this paper, the prediction of stock return and risk are implied concurrently based on fundamental features in order to build a more comprehensive model for stock market analysis.

Although the statistical approaches such as logistic regression and regression analysis are widely applied to forecast the return and risk of stocks, the results of machine learning approaches are generally superior in comparison to statistical methods [9–11]. The multiple classifier ensemble system (MCS), one type of machine learning technique, has recently become the focus of a new methodology for obtaining higher accuracy in predictions. The rationale is that the optimization of a combination of relatively simpler predictors appears more convenient than optimizing the design of a single complex predictor [12,13]. In fact, three fundamental issues are effective for establishing a successful MCS model: accuracy of individual classifiers, diversity among classifiers, and the choice of the fusion methods that will be used. The aim of this combination scheme is to gain increased precision with proper single classifiers and eliminate the uncorrelated individual classifier errors, which are the errors made by individual classifiers on various parts of input space [14].

In fusion methods, multiple dissimilar predictors are used and combined by a fusion algorithm that combines the outputs of the individual predictors. Fusion methods in MCS are generally categorized as linear, non-linear, statistical, and machine learning combination methods. Linear methods are known as the simplest fusion methods. For instance, the sum and average of the individual classifiers' outputs are examples of linear fusion methods [15]. Non-linear methods include rank-based combiners such as Borda Count and majority voting strategies [15,16]. In statistics-based fusion methods, statistical techniques such as regression or Bayesian combination methods are used to combine the outputs of individual classifiers [17]. Finally, different machine learning methods such as decision trees (DT) and support vector machines (SVM) can be used to fuse the base learner.

The second fundamental issue, diversity, refers to the differences existing among decisions made by various classifiers. In classifier combination design, it is believed that the success of combinations not only depends on the individual classifiers' suitability but also on diversity being inherent among them. In fact, classifiers that are strong in different areas are supposed to be diverse. The entire point of fusing multiple classifiers is to balance the weaknesses of the individual classifiers. This balancing requires classifiers that make errors in different areas of the decision space. Diversity creation methods are generally categorized as explicit and implicit methods [12]. Explicit methods generally seek to optimize certain metrics during the diversity creation. Boosting and AdaBoost [18] are examples of explicit diversity methods that directly manipulate the training data distributions in order to make some sort of diversity in the combination procedure. Implicit methods, unlike explicit methods, pay no special attention to diversity metrics. Bagging, as an implicit method, randomly samples from training data in order to train each individual classifier and these samples will be reused to produce diverse combination members [19].

In the present paper, in which we aim to improve the accuracy of the risk and return prediction of stocks, we propose a fusion model framework that relies on the combination of multiple dissimilar and diverse classifiers operating on a common input. In the first phase, cross-validation is applied on the dataset and a specific (but optimum) number of different classifiers sets is learned from the dataset (creating a pool of classifiers). A classifier selection procedure is proposed in which the dataset is first clustered by the *k*-means method, after which the optimum number of clusters is chosen by Streamlined Silhouette Criterion Average (SSCA). The performance of the classifiers on the dataset is then evaluated and the best combinations are selected for the fusion phase. Finally, in the fusion phase, Bagging, Boosting and AdaBoost are applied to the classifiers of the selected combinations of the previous phase and one fusion algorithm as a Meta-classifier learns from their predictions to provide the final prediction of the initial input data.

The contribution of the paper is summarized as follows:

- Designing a fusion model for returns and risk prediction of stocks in financial market.
- Applying various diversity methods in order to achieve more precise predictions.
- Considering the simultaneous risk and return prediction of stocks.
- Developing a base classifier selection procedure from candidate procedures by dataset clustering and considering the accuracy of combined classifiers.
- Developing a wrapper-GA scheme for feature selection and prediction and comparing it with the fusion method.

This paper is divided into six sections and organized as follows: The backgrounds of multi-classifier systems and combination models, as well as diversity creation and fusion methods, are discussed in Section 2. In Section 3, the proposed multi classifier ensemble system (MCS) is generally discussed. Next, in Section 4, the experimental results are provided, and the discussion of real return and risk prediction with the proposed fusion model is presented in Section 5. Moreover, in this section, a new selection scheme is also developed for feature selection part and compared with the hybrid method. Finally, this study's conclusions and future research directions are presented in Section 6.

2. Literature review

2.1. Stock prediction with classifier ensembles

The best way to design of MCS to achieve higher accuracy has become an important research topic in the field of pattern recognition, as stated in a number of related review articles [20–22]. The main idea behind using ensembles is that the combination of classifiers can improve the performance of a pattern recognition system in terms of better generalization with increased efficiency and clearer design [23]. Wolpert [24] believes that every classifier has its own specific competencies over other competing algorithms, and MCS tries to take advantage of each of the available trained classifiers based on their competencies for different parts of the feature space.

Dasarathy and Sheela [25] proposed combining a linear classifier and a *k*-nearest neighbour classifier in which the conflicting feature space regions were first identified by classifiers, after which one classifier works on the features of the conflicting region and the other works on the remaining features. This can be considered as the first study suggesting a classifier selection concept for MCS design. In 1981, Rastrigin and Erenstein [26] further developed the idea by partitioning the feature space into several regions and assigning the individual classifiers with the best accuracy over each region. A survey of multiple classifier systems as hybrid systems can be found in [27].

In the stock exchange and financial research area, different machine learning methods such as artificial neural networks (ANN), decision trees (DT) and support vector machines (SVM) are widely applied to establish efficient ensemble systems. Neural network ensemble systems are found to be effective in achieving superior accuracy for stock price forecasting [5,28,29]. In addition to neural networks, algorithms based on decision trees use a greedy search approach and tend to choose a search direction by means of a heuristic attribute evaluation function [30]. This approach, however, does not guarantee finding an optimal solution. Thus, a combined algorithm starting from different initial points in the search space can improve the DT's performance in finding an optimal model. Qian and Rasheed [31] combined several machine learning classifiers, including artificial neural networks, decision trees and *k*-nearest neighbour, to design a hybrid model for stock market prediction. The results show that accuracy of up to 65% is achieved. Tsai et al. [14] combined three classifiers, including multi-layer perceptron (MLP) neural networks, SVM and DT based on a combination of the bagging and boosting methods. Their results showed that DT ensembles utilizing boosting techniques have the best performance, and this superiority is demonstrated by further studies on a Taiwan bankruptcy dataset.

2.2. Fusion in ensembles scheme

Fusion methods refer to the approaches used to obtain classifier ensembles. The motivation behind such methods is to combine predictions so that misclassification is less likely to occur. In other words, a proper fusion method is the one that can exploit the strength of individual classifiers and optimally combine their outputs to provide the final decision of the system [27]. Among early studies regarding fusion techniques for MCS, the majority voting schemes application is widely known [32].

In another category, aggregation methods such as supremum, average, mean or median value perform simple fusion operators and lack any learning procedures [33–35]. The most essential advantage of these methods is that they balance the over-fitting of the individual classifiers. Tumer and Ghosh [36] used a large number of unbiased and independent classifiers and reported the average of the outputs as the final results. These researchers claimed that their method returned the same results as applying the optimal Bayes classifier. Ho et al. [37] used different methods based on decision ranks, such as Borda counts, for the combination function in order to achieve a useful representation for each classifier's decision. A Borda count is categorized as a support function in a fusion system that assigns a score for the decisions taken from each individual classifier.

Machine learning fusion methods are another group of fusers that use the accuracy of individual classifiers as training data and then apply a learner algorithm, e.g., DT, K-NN,…, as a high level classifier (meta classifier) that learns from the accuracy of individual classifiers to obtain higher accuracy [38].

2.3. Diversity and accuracy creation

System diversity is highlighted as a crucial important aspect in MCS design [39–41]. The main purpose of designing MCS is to integrate a set of mutually complementary individual classifiers in order to achieve outputs with higher accuracy and diversity and less correlation. Diversity can generally be achieved by inducing variations in classifier parameters (e.g., weights and topology of a neural network as initial parameters) [42], classifiers' training datasets (e.g., using the learning strategies, such as Bagging and Boosting) [43], and classifier types (e.g., using different types of classifiers as ensemble members).

Variation in classifiers' training dataset (data partitioning) is important for several reasons, such as data privacy or learning procedure requirements over distributed data partitions in different databases. In this category, cross-validation is a well-known approach that minimizes overlap among dataset partitions [44]. Bagging [19] and Boosting [45,46] are both known as the most popular techniques in diversity creation and originate in bootstrapping. In this paper, diversity is achieved by making variations in the training dataset by means of the abovementioned techniques (e.g., cross-validation and Bagging, Boosting) and using different classifier types (e.g., neural network, decision trees, and SVM) as ensemble members.

3. Proposed multi-classifier ensemble system (MCS)

The MCS is generally composed of three main phases: (1) generation, (2) selection, and (3) integration (see [47]). The generation phase focuses on creating a pool of base classifiers composed of the most appropriate candidates for the subsequent classifier selection and integration steps. In the second phase, the best classifiers from the pool are selected for building the MCS, and finally, in the integration phase, the predictions of the selected classifiers are combined in order to make a final decision.

In the generation phase, the aim is to create as many diverse classifiers as possible. Thus, several diversity methods are applied to build a pool of diverse base classifiers. In the second and third phases, a meta-learning approach is proposed [48]. In other words, the selection of the most competent classifiers and final classifications are considered as another classification problem, termed the meta-problem. For a given instance, the base classifiers' outputs and the real class of the sample are passed down to a Meta classifier as its input data, and then the Meta classifier estimates the final class of the given sample.

In the following, three phases of the proposed MCS framework, including the generation of the base classifiers (pool of classifiers), selection of competent classifiers and final fusion phase are widely discussed.

3.1. Phase one: base classifier generation

The main aim of MCS design is to effectively select competent classifiers and combine their predictions. In order to ease and increase the accuracy of the selection procedure, it is preferable to build an initial set of potentially competent base classifiers (pool of classifiers) first and then decide whether each candidate is sufficiently qualified to classify an input instance. A key factor here is to combine the prediction results of those classifiers in which the decision boundaries are widely different. To achieve this, the creation of diversity among classifiers is proposed. A diversification strategy aims to train the classifiers on different (disjoint) input subspaces in order to create a set of different but complementary classifiers. Several diversification approaches that are mainly used for generating diverse classifiers are discussed by Mousavi and Eftekhari [49]. Using unstable classifiers such as decision trees (DT) and neural networks (NN), applying different sets of classifiers including NN, DT, SVM, etc., and utilizing various datasets for training the base classifiers are the diversification methods used in this paper. To create variations in classifiers' training dataset, four methods are applied: Cross-validation, Bagging, Boosting and AdaBoost, each of which is discussed in detail below.

3.1.1. Cross validation

In order to generate variations in the training dataset, data partitioning approaches are used to create various partitions for classifier training. A common methodology in this category is cross-validation, which evaluates the robustness of the predictor and minimizes the overlapping of dataset partition. We utilize a 10-fold cross-validation model, which is proven to be sufficient in the predictor's performance evaluation [50]. In this model, the training dataset is divided equally into 10 subsets. The training procedure is repeated 10 times, and each time, nine out of 10 of the subsets are selected for classifier training while the tenth subset is used as the test set. When the procedure is complete, the best result will be selected.

3.1.2. Bagging

Bootstrap aggregation, simply known as Bagging, is an ensemble-based algorithm and one of the most intuitive and simple to implement methods with extremely good performance. In the Bagging approach, the classifier is trained on different training datasets that are generated by bootstrap method [19]. The Bootstrap method builds k training datasets by randomly re-sampling the original given dataset with replacement. Thus, there are k independent training datasets for the classifier training procedure. When classifier training is complete, the final results should be aggregated via an appropriate method, such as majority voting. Pseudo code for the Bagging method is given as follows:

Algorithm: Bagging

```
Input:
    Training data $S$ with correct labels $\omega_i \in \Omega=\{\omega_1, ..., \omega_C\}$ representing $C$ classes
    Weak learning algorithm WeakLearn
    Integer $T$ specifying number of iterations
    Percent (of fraction) $F$ to create bootstrapped training data
Do: $t=1,...,T$
    1. Take a bootstrapped replica $S_t$ by randomly drawing percent of $S$.
    2. Call WeakLearn with $S_t$ and receive the hypothesis (classifier) $h_t$.
    3. Add $h_t$ to the ensemble, $E$.
End
Test: Simple Majority Voting - Given unlabeled instance $x$
    1. Evaluate the ensemble on $x$.
    2. Let
       $\upsilon_{t,j} = \begin{cases} 1, & \text{if } h_t \text{ picks class } \omega_j \text{ be the vote given to class by classifier.} \\ 0, & \text{otherwise} \end{cases}$
    3. Obtain total vote received by each class $V_j = \sum_{t=1}^{T} \upsilon_{t,j},\ j = 1, \ldots, C$
    4. Choose the class that receives the highest total vote as the final classification.
```

3.1.3. Boosting

In Boosting method, it is believed that finding many prediction rules can be much easier than building one rule with a high level of accuracy. In boosting, unlike bagging, each individual classifier is trained on different k training sets in a sequential and not a parallel and independent way. The algorithm creates an ensemble of classifiers through data resampling in order to provide the most informative training data for each consecutive classifier. Boosting creates three weak classifiers: the first classifier, C_1, is trained by a random sample of the training data. For the second classifier, C_2, the training set is selected as the most informative subset. In other words, half of the training data for C_2 is correctly classified by C_1 and the other half is misclassified by C_1. Finally, the third classifier C_3 is trained on samples that are misclassified by both C_1 and C_2. In fact, in each iteration, the classifier creates a new set of prediction rules and assigns new weights to data so that the classifier will pay more attention to misclassified tuples in subsequent iterations. Finally, after many repetitions, the boosting algorithm combines these rules into one single prediction rule that is expected to be much more accurate than any of the single rules.

Ultimately, boosting places heavier weights on the samples that are most often misclassified in every round. In other words, it forces the base learner (i.e., the individual classifier) to focus on the hardest samples that were mostly misclassified by the preceding rules. In order to combine the achieved prediction rules, using a (weighted) majority voting approach is suggested as an efficient method [46]. The pseudo code for the Boosting algorithm is given as follows:

Algorithm: Boosting

```
Input:
    Training data $S$ of the size $N$ with correct labels $\omega_i \in \Omega=\{\omega_1, ..., \omega_C\}$;
    Weak learning algorithm WeakLearn.
Training
    1. Select $N_1 < N$ patterns without replacement from $S$ to create data subset $S_1$.
    2. Call WeakLearn and train with $S_1$ to create classifier $C_1$.
    3. Create dataset as the most informative dataset $S_2$, given $C_1$, such that half of $S_2$ is correctly classified by $C_2$, and the other half is misclassified. To do so:
       a. Flip a fair coin. If Heads, select samples from $S$, and present them to $C_1$ until the first instance is misclassified. Add this instance to $S_2$.
       b. If Tails, select samples from $S$ and present them to $C_1$ until the first one is correctly classified. Add this instance to $S_2$.
       c. Continue flipping coins until no more patterns can be added to $S_2$.
    4. Train the second classifier $C_2$ with $S_2$.
    5. Create $S_3$ by selecting those instances for which $C_1$ and $C_2$ disagree. Train the third classifier $C_3$ with $S_3$.
Test – Given a test instance $x$
    1. Classify $x$ by $C_1$ and $C_2$. If they agree on the class, this class is the final classification.
    2. If they disagree, choose the class predicted by $C_3$ as the final classification
```

3.1.4. AdaBoost

In AdaBoost [51], the most popular boosting algorithm, a series of models are combined; in each model, the dataset is re-sampled and weighted based on their difficulty to be learned and classified [52]. AdaBoost takes the training set $s_n=\{(x_1,y_1), ..., (x_m,y_m)\}$ as input data and calls the base learning algorithm repeatedly for a series of iterations $t=1, ..., T$. The $w_t = \{ w_t^1, w_t^2, \ldots, w_t^n \}$ represents the weight distribution over samples in iteration t and is equally distributed in the first iteration. In each iteration t, AdaBoost maintains the weights on the training sample i denoted as w_t^iso that the weights of misclassified samples will increase and the learner algorithm will pay more attention to these difficult samples in the training set in subsequent iterations. Based on the given w_t in each round t, the base learning algorithm finds the most appropriate classifier h_t and assigns an importance measure α_t to it, set as:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - e_t}{e_t} \right) \tag{1}$$

where e_t represents the mean squared error (MSE) for h_t. The final classifier H will be built by a weighted majority vote of the T base classifiers in which the parameter α_t is assigned as the weight of classifier h_t. The pseudo code of the presented AdaBoost is given as follows [13]:

Algorithm: AdaBoost

```
Input:      Initial training set composed of n samples, denoted as $s_n=\{(x_1,y_1), ..., (x_m,y_m)\}$
Initialize: $w_1^i=1/n$, i.e., $w_1=\{w_1^1, w_1^2, \ldots, w_1^n\}=\{1/n, 1/n, \ldots, 1/n\}$
            For $t=1,2, ..., T$
                Take $R_t$ samples randomly from $s_n$
                Determine the weight distribution $w_t$
                Build a classifier $h_t$ using $R_t$ as the training set
                Compute: $e_t$=MSE for $h_t$ and $\alpha_t = \frac{1}{2}\ln(\frac{1-e_t}{e_t})$
                Update the distribution weight set: $w_{t+1}^i=normalize(w_t^i * \exp(-\alpha_t))$
Output:     The combined classifier: $H = \sum_{t=1}^{T} \alpha_t h_t$
```

3.2. Phase two: classifier selection

In addition to diversity, the accuracy of the base classifiers is another critical factor in base classifier selection [53,54] and can guarantee the effectiveness of the MCS.

The proposed selection phase seeks to discover sets of classifiers (fusion set) that improve the classification accuracy in integration. In order to specify the optimum number of classifiers in fusion sets, it is suggested to determine the optimum number of clusters of the dataset first and then find the sets of classifiers with the same number. To achieve this, the k-means clustering algorithm is first applied on the dataset for different values of k, after which the best value is chosen.

The k-means algorithm is one of the simplest unsupervised learning methodologies, using a simple method to classify a given dataset into a certain number of clusters (k clusters). The aim is to determine k centroids that should be cunningly placed as different locations resulting in different outputs. Thus, it is better to place these centroids as far as possible from each other. Next, each point of the dataset should be associated with the nearest centroid. When no point is left, early grouping is complete. At this point, k new centroids must be recalculated for the centres of clusters built in previous step. A new binding must be carried out on the dataset points toward the new k centroids. This loop continues until no more changes occur in centroids' locations and they do not move any further. For a given set of observations (x_1, x_2, …, x_n), where each observation is a D-dimensional real vector, the best condition in k-means clustering is to partition the n observations into k ($\leq n$) sets $S=\{S_1, S_2, \ldots, S_k\}$ in which the within-cluster sum of squares is minimized:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2, \tag{2}$$

where μ_i is the mean of points in S_i.

The k-means algorithm is then performed for different values of k. In order to specify the optimum number of clusters (k), the streamlined silhouette criterion average (SSCA) is calculated [55]. SSCA evaluates the closeness of each object of a cluster to the objects of nearby clusters. For the ith observation of the kth cluster, the Silhouette index is defined as follows:

$$s_{i,k} = \frac{b_{i,k} - a_{i,k}}{Max(b_{i,k}, a_{i,k})} \tag{3}$$

where $b_{i,k}$ is the minimum average distance between observation i and all other observations to the nearest neighbouring clusters except the kth cluster, and $a_{i,k}$ represents the average distance between observation i and the remaining (n_k-1) observations of the kth cluster. The most effective clustering is the one that results in maximum SSCA, computed as:

$$SSCA = \frac{1}{k} \sum_{j=2}^{k} \frac{1}{n_j} \sum_{i=1}^{n_j} s_{i,j}. \tag{4}$$

The optimum number of clusters (k) resulting from SSCA calculations determines the number of base classifiers in ensemble that create the widest decision boundaries. The proposed classifier selection process is illustrated in Fig. 1. The cross-validation is first performed on training data for diversity creation among base classifiers. In order to increase the base classifiers' accuracy, the heuristic Meta cost method is applied. This method intensifies the error of prediction by multiplying it by an integer number. In other words, if the classifier predicts class A for an input data but the real class is two steps above or below the predicted class, Meta cost penalizes the error by doubling it. Similarly, if the real class is three steps away from the predicted one, the error will be multiplied by three. Table 1 illustrates the error penalty scores of the Meta cost method.

Based on the optimum value of k achieved by SSCA, all possible sets of k classifiers are formed and their outputs, including the real class of the samples, are passed down as inputs to the meta-classifier. For each set of classifiers, the accuracy index is calculated. If it is higher than 75%, the related set is selected; otherwise, it will be removed from the set of classifiers.

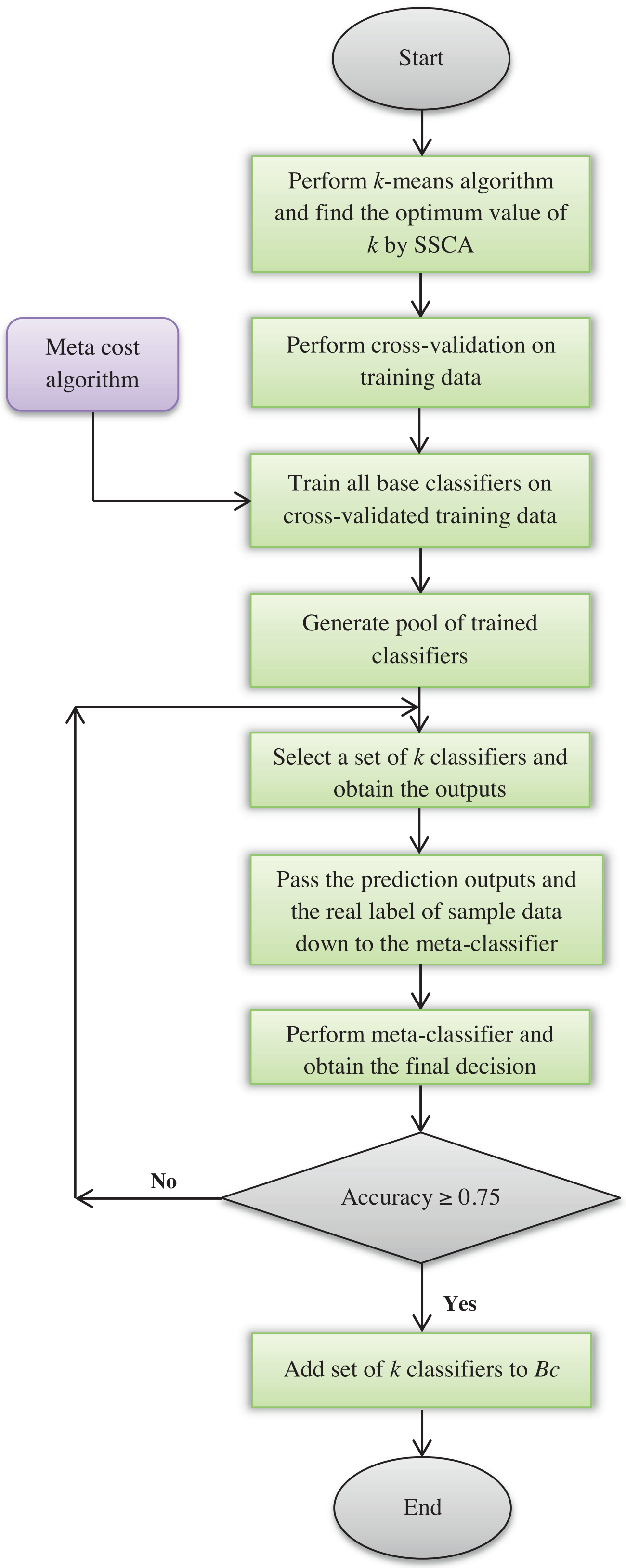
Start
Perform k-means algorithm and find the optimum value of k by SSCA
Perform cross-validation on training data
Meta cost algorithm
Train all base classifiers on cross-validated training data
Generate pool of trained classifiers
Select a set of k classifiers and obtain the outputs
Pass the prediction outputs and the real label of sample data down to the meta-classifier
Perform meta-classifier and obtain the final decision
Accuracy ≥ 0.75
No
Yes
Add set of k classifiers to Bc
End

Fig. 1. Selection phase process.

Table 1
Meta cost penalty score matrix.

		Real class				
		1	2	3	4	5
Predicted class	**1**	0	1	2	2	3
	2	1	0	1	2	2
	3	2	1	0	1	2
	4	2	2	1	0	1
	5	3	2	2	1	0

Table 2
Confusion matrix for a five-class prediction problem.

		Predicted class				
		Very low	Low	Normal	High	Very high
Actual class	Very low	a_1	b_1	c_1	d_1	e_1
	Low	a_2	b_2	c_2	d_2	e_2
	Normal	a_3	b_3	c_3	d_3	e_3
	High	a_4	b_4	c_4	d_4	e_4
	Very high	a_5	b_5	c_5	d_5	e_5

The accuracy of the meta-classifier is defined as the percentage of the correctly predicted sets of tuples on the given dataset. For a five-class prediction problem, the accuracy can be measured by means of a confusion matrix, as shown in Table 2 with the associated formula:

$$Accuracy = \frac{a_1 + b_2 + c_3 + d_4 + e_5}{\sum_{i=1}^{5} a_i + b_i + c_i + d_i + e_i}. \tag{5}$$

The proposed method considers the accuracy and diversity of the candidate classifiers simultaneously and results in a set of competent classifiers for the subsequent integration stage. The entire selection phase is formalized in Algorithm 1 as follows:

Algorithm 1 Classifier selection phase.

```
Input:
Pool of classifiers, denoted as $P = \{c_1, c_2, \ldots, c_M\}$;
Training dataset, denoted as Train_data;
Optimum number of clusters achieved by k-means and SSCA, denoted as $k$ ($k \le M$).
Output: Set of selected base classifiers denoted as Bc.
Algorithm.
1:  $Bc = \emptyset$
2:  Perform 10-fold cross-validation on Train_data.
3:  for all $c_i \in P$ do
4:     Train $c_i$ by Train_data.
5:     Perform Meta cost algorithm.
6:  end for
7:  for $j = 1 : C_k^M$
8:     Form $Set_j$ with $k$ members taken from $P$ without replacement.
9:     Achieve the outputs of classifiers in $Set_j$ and the real class of sample data, and pass them down to the Meta-classifier.
10:      if Average Accuracy > 75%
11:         $Bc = Bc \cup Set_j$
12:      end if
13:   end for
14:   Return Bc
```

3.3. Phase three: fusion

In the generation phase, different diversity algorithms were introduced for the initial generation and training of classifiers. The cross-validation is used in the selection phase and the remaining algorithms, including Bagging, Boosting and AdaBoost, will be used for this final fusion phase. Because different machine learning algorithms have dissimilar errors, the fusion of multiple different classifiers is expected to decrease the overall error rate.

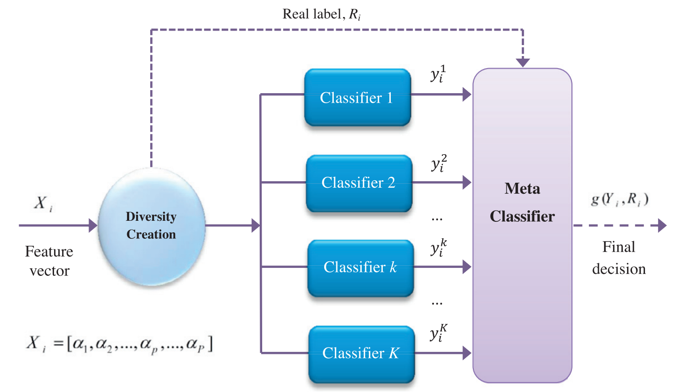

Fig. 2. Generalized flowchart of proposed fusion phase.

3.3.1. Fusion scheme: training and operation

Apart from the initial diversifications, the fusion scheme is similar to the previous selection phase. In order to train the fusion scheme, the three above mentioned diversity algorithms are first used to train the classifiers of every selected set in *Bc*. Then, for a given sample dataset, the classification results, including the real class, are passed down as inputs to the meta classifier. Based on the achievements of base classifiers and the real response, the meta classifier will return the final classification of the given sample data.

Suppose we have a set of k stock classifiers, C_k, with $1 \le k \le K$. The feature vector, X_i, for the ith instance ($1 \le i \le I$) that is used for training all C_k of set j ($1 \le j \le J$) is defined as:

$$X_i = [\alpha_1, \alpha_2, \ldots, \alpha_p, \ldots, \alpha_P], \quad i = 1, 2, \ \ldots, I \tag{6}$$

where α_p is the pth feature ($1 \le p \le P$) of the feature vector X_i.

When all the C_k are trained, they are fed with the test data and the outcomes are the set of individual classifiers' predictions for ith instance, y_i^k, which are defined as:

$$y_i^k = f_{C_k}(X_i) \quad i = 1, 2, \ \ldots, I \tag{7}$$

The vector Y_i, is formed by individual predictions, y_i^k, with the real label of the ith instance, R_i, being used to train the upper level meta classifier.

$$Y_i = \{y_i^k\}^T \quad i = 1, 2, \ \ldots, I \tag{8}$$

Once the meta classifier is trained, the fusion scheme is prepared for further operations.

For each set j of *Bc*, the feature vector of the ith instance of test data, X_i, is given to the k individual classifiers, C_k, as input data. The outputs, y_i^k, form the prediction vector, Y_i, and together with real label, R_i, are used as input data for the meta classifier. The final decision for ith instance is made as given by:

$$D_i = g(Y_i, R_i) \quad i = 1, 2, \ \ldots, I \tag{9}$$

for some real values of D_i. Fig. 2 shows the generalized flowchart of the fusion phase.

4. Experimental results

4.1. Datasets

In this paper, the dataset comprises data from the Tehran Stock Exchange (TSE) from 2002 to 2012 for a total of 1963 records for 400 companies.

According to the literature, negative and positive returns and return trends are widely used for predicting returns [5,56,57]. In order to enhance accuracy, however, and based on a group of experts' opinions, a set of intervals are introduced in this paper for

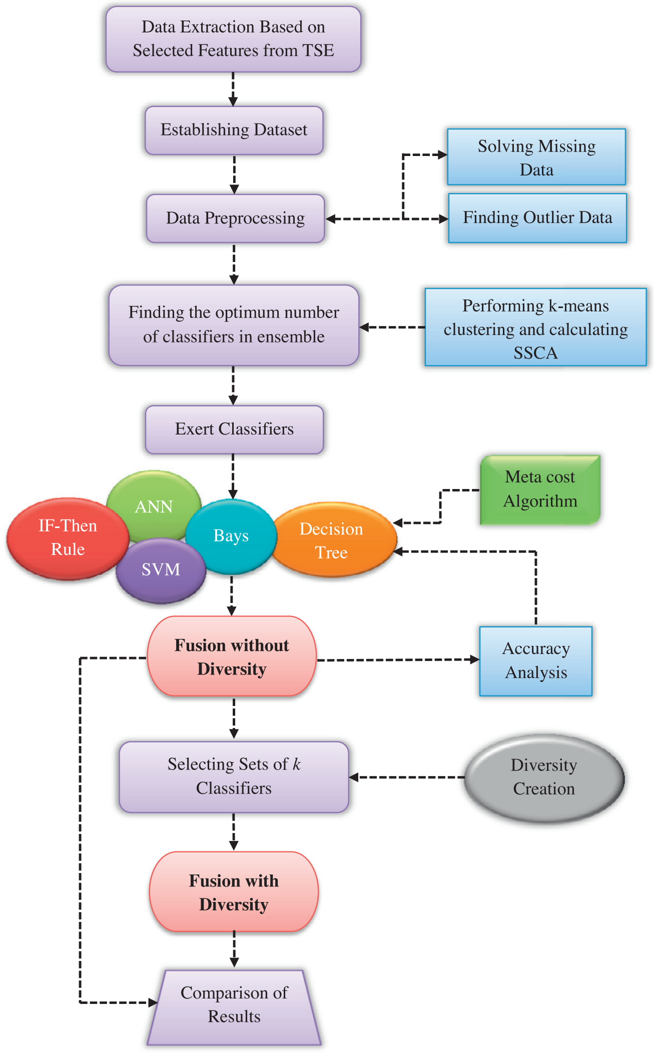

Fig. 3. The proposed process of stock return and risk prediction.

real return and risk prediction. Therefore, more information will be given to the investors for selecting the best optimal portfolio. For real returns, 5 intervals are specified: very high with a range higher than 9.3, high with a range of 4 to 9.3, average with a range of 1.14 to 4, low with a range of −1.3 to 1.14 and very low with a range lower than −1.3. Similarly, 3 intervals are specified for risk: high with a range higher than 15.5, average with a range of 6.3 to 15.5 and low with a range lower than 6.3. Fig. 3 illustrates the entire proposed process of stock return and risk prediction.

4.1.1. *Data pre-processing*

In order to find the outlier data in the dataset, the distance-based approach was first used to analyse the remote records. The results show certain very large governmental companies as outlier data. The density approach was also used and 12 records were identified as outlier points, 7 of which were large companies that remained in dataset. The remaining 5 records were deleted, mostly because they were not sufficiently accurate for the process. By using the clustering approach, 6 points were identified as outliers because they did not belong to any of the clusters. The input feature of the companies was analysed and no suspected case was detected; therefore all the companies remained in the dataset. Finally, a number of techniques based on deviation were used that found 5 outlier records, all of which were removed. Moreover, because of a lack of information, 12 records out of 1963 were identified as missing data and therefore omitted from the dataset.

Table 3
Selected features for risk and return factors.

Response variable	Selected features
Risk	Return, Beta coefficient, Efficiency, Market return, EPS prediction, Percent of growth EPS, DPS, *P/E*, EPS, Equity ratio, Stock book value, Debt to total asset ratio, Predicted profit margin, *P/S*, Total incomes growth.
Return	Return, Market return, Beta coefficient, Return on asset (ROA), Percent of growth EPS, EPS, Predicted profit margin, EPS coverage percent.

4.2. *Feature selection*

In order to accurately predict the risk and return variables, the most effective features of these variables must be identified first. In fact, the selection of representative features plays a key role in an efficient stock prediction design. Referring to [11] and based on the fundamental approach, features that have the potential to be effective in risk and return predictions are first gathered from the company's financial ratios, stock pricing models and profit and loss reports. Then, a comprehensive procedure is used to sort out the most effective ones as the best representative features for risk and return classification. The procedure is based on a hybrid algorithm of filter and function-based clustering and selects 15 features for return prediction as well as 8 features for risk prediction from a group of 45 different financial features. In our proposed model, the same features shown in Table 3 are used in all classifications. The full list of all determined features is provided in Appendix A.

4.2.1. *Response variables*

In our proposed model, the most important response variables are considered as real return and risk, as follows:

$$R = \sqrt[n]{\left(1+\frac{r_1}{100}\right)\left(1+\frac{r_2}{100}\right)\ldots\left(1+\frac{r_n}{100}\right)} \tag{10}$$

where r_1, r_2,...,r_n represent the real return of 1,...,nth periods.

The standard deviation of the stock return is assumed to be the risk response variable, as follows:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(r_i - E(r)^2)} \tag{11}$$

4.3. *Individual classifiers*

In stock prediction, the complexity of the data necessitates applying models that are capable of defining such intricacy. Different methods, including statistical methods and neural networks, are studied and it is found that the results gained by machine learning and data mining algorithms are much more prominent [56]. In this research, Decision Trees (DT), Artificial Neural Networks (ANN), rule based algorithms and Support Vector Machines (SVM) are used as the individual classifiers.

Decision trees and rule based algorithms are known to be powerful prediction algorithms with outstanding performance in stock return prediction [58]. Rule-based classifiers use a set of IF-THEN rules for classification, which is especially useful when there are specific relationships among input variables. A decision tree is composed of decision rules that separate the independent variables into homogeneous areas and build rules that can be used for the output prediction of a set of input variables. In fact, the rules obtained from this group of classifiers are of importance to investors seeking to make their best portfolio. In this study, the LAD tree,

Table 4
SSCA calculations on different number of clusters.

Number of cluster	2	3	4	5
SSCA	12.27	14.43	11.38	9.74

Cart decision tree, Rep tree, BF tree and certain other popular, rule-based algorithms such as decision tree naive Bayes (DTNB) rule, J RIP rule, RIDOR rule and Part Rule are used.

Artificial Neural Networks (ANNs) are a group of analytical techniques capable of approximating extremely sophisticated non-linear functions. The multi-layer perceptron (MLP) is a popular neural network architecture that is a strong function estimator for prediction or classification problems. The MLP is able to learn complex non-linear functions to an arbitrary level of accuracy.

Support Vector Machines (SVMs) are the third group of classifiers used in this research. SVMs are categorized as generalized linear models that achieve the classification/regression decision by means of features linear combination. The approximated function in SVMs can be either a classification function, which is also used for the data categorization of this research, or a regression function for estimating the numerical value of input data.

4.4. Selection of classifier sets

In second phase, sets of classifiers with size k should be generated. In order to specify the optimum number of classifiers (k), the k-means algorithm is first performed with different values of k. Then, the SSCA is calculated for each cluster based on Euclidean distance. The results in Table 4 shows that if we increase the number of clusters, the maximum amount of SSCA will be achieved in $k=3$, after which its value starts to decrease. Thus, using $k=3$ and based on Algorithm 1, sets of classifiers with three members are generated that are then used in the final fusion phase.

Tables 5 and 6 show the performance of each individual classifier for forecasting the return and risk of stocks on the given dataset [11]. In order to specify the meta-classifiers, the algorithms that provide greater prediction accuracy on the given dataset are chosen as meta-classifiers. Investigations state that trees and rule based algorithms with a denser structure generally display better accuracy than larger ones. Thus, the meta-classifiers are chosen from such dense and accurate algorithms.

The 10-fold cross-validation is performed and the average accuracy of classifier sets with 3 members for stock return and risk predictions are given in Tables 7 and 8, respectively. These selected sets are specified as the best ensembles whose performance with three diversity methods is analysed in the fusion step. The results reveal that although some of algorithms did not individually display an accurate performance (e.g., SVM, Bayes, MLP), they could achieve an acceptable level of accuracy in an ensemble model.

4.5. Stock return and risk predictions with the proposed fusion scheme

In the final phase, as illustrated in Fig. 2, three diversity algorithms (Bagging, Boosting and AdaBoost) are performed separately for each selected set, as shown in Tables 7 and 8, and the final results for risk and return predictions are reported in Tables 9 and 11, respectively. To make more reliable results with more baseline algorithms, Random Forest as well as K-Nearest Neighbours (K-NN), SVM, and Bayes algorithms' set are also implemented.

For return prediction, the highest prediction accuracy is estimated as 83.65%, which is achieved by the Decision Tree, LAD Tree and Rep Tree ensemble with BF Tree as its fusion algorithm and Bagging as its diversity method. The detailed prediction results of the superior ensemble are presented in Table 10.

As shown in Table 11, the highest accuracy of 88.23% in risk prediction is achieved by the BF Tree, DTNB and LAD Tree ensemble with Decision Table as the fusion algorithm and Bagging as the diversity method. The detailed prediction results of this ensemble are presented in Table 12.

Based on the results in Tables 9 and 11, Bagging generally displayed better performance in improving the prediction accuracy of ensembles in comparison with Boosting and AdaBoost; however, it could not effectively improve the prediction accuracy of ensembles with weak base classifiers. In order to predict the real return and risk of stocks, voting methods such as simple averaging and weighted averaging are also used as the meta classifier algorithms, which resulted in maximum accuracy of 80% for real return prediction, although performance was poor for risk prediction.

5. Discussion

A comparison of the findings presented in Tables 7 and 9 shows the significant role of applying Bagging as a diversity method on accuracy improvement in real return prediction. Bagging enhanced the accuracy of almost all ensembles except the ensemble of BF Tree, Bayes and SVM with LAD Tree fusion algorithm. The reason may be the weakness of Bayes and SVM algorithms (in our dataset) used as base classifiers, whose performance could not be improved even with Bagging.

Moreover, based on findings shown in Tables 5 and 9, the performance of almost all stock classifiers has greatly improved in fusion system with the exception of the LAD Tree and Decision Tree, which provided higher accuracy only in concert with the Bagging diversity algorithm.

The results achievements show that the fusion of the strongest individual classifiers has led to more accurate prediction, but this is not necessarily always true. For instance, the accuracy of MLP is 69%, but in fusion with Part Rule and J48 Graph algorithms, the accuracy improved to 78.74%. One reason for this may be that the weakness of some algorithms can be covered by other classifiers used in fusion system. In other words, the algorithms can demonstrate complementary behavior when they are combined in this way.

Among several classification algorithms used in this paper, the SVM and Bayes algorithms demonstrated different behavior. Although their individual accuracies in both real return and risk prediction were improved by the fusion system, the findings revealed that the fusion system without diversity resulted in better accuracy. The high sensitivity of SVM to the missed data individually resulted in a low accuracy of 60% for real return prediction, but the ensemble with the Bayes and BF Tree algorithms diversified by Bagging led to an accuracy of 70.66%; without diversity creation, an accuracy of 78.04% was achieved.

Similar to real returns, the achievements on risk prediction also revealed that the fusion system with diversity successfully improved the performance of individual classifiers in all combinations except for the SVM and Bayes algorithms, whose performance was not improved by diversity creation. The superiority of Bagging in improving accuracy in comparison with the other two diversity algorithms shows its high consistency with the proposed prediction procedure.

Because of the low accuracy, the fusion results of some sets of classifiers with Boosting and AdaBoost methods are not reported in Tables 9 and 11.

As stated above, Bagging significantly outperformed the Boosting and AdaBoost methods. An important question to ask is why Bagging outperforms other methods and what exactly differentiates it from Boosting and AdaBoost. One reason for its superior performance may be the level of "sensitivity" displayed by each of these algorithms towards the degree of clearance in the dataset.

Table 5
Individual classifier comparisons for real return prediction.

Algorithm	Accuracy	Number of rules	Tree size	Number of leaves
LAD Tree	78.00	–	31	15
Cart Decision Tree	76.50	–	13	7
DTNB rule	76.00	998	–	–
Decision table	75.50	56	–	–
Rep Tree	75.00	–	33	17
BF Tree	74.50	–	9	5
Part Rule	72.60	104	–	–
J48 Graph	71.50	–	1619	810
Neural Net (MLP)	69.00	–	–	–
Bays	60.00	–	–	–
SVM	60.00	–	–	–

Table 6
Individual classifier comparisons for risk prediction.

Algorithm	Accuracy	Number of rules	Tree size	Number of leaves
LAD Tree	78.24	–	31	20
DTNB rule	77.41	426	–	–
Decision table	76.57	297	–	–
BF Tree	76.15	–	109	55
Part Rule	73.64	55	–	–
Rep Tree	72.8	–	77	39
Neural Net (MLP)	59.00	–	–	–
Bays	55.65	–	–	–

Table 7
Average accuracy percentage of selected sets for stock return prediction (without diversity).

Set Number	Base classifiers			Fusion algorithm	Accuracy (%)
1	Decision Tree	LAD Tree	BF Tree	DTNB	78.69
2	BF Tree	Bays	SVM	LAD Tree	78.04
3	Decision Tree	LAD Tree	Rep Tree	BF Tree	77.34
4	Decision Table	DTNB	Rep Tree	Decision Tree	76.23
5	MLP	Part Rule	J48 Graf	BF Tree	75.84

Table 8
Average accuracy percentage of selected sets for stock risk prediction (without diversity).

Set Number	Base classifiers			Fusion algorithm	Accuracy (%)
1	Decision Tree	LAD Tree	DTNB	Decision Table	78.94
2	DTNB	LAD Tree	BF Tree	Decision Table	78.34
3	Part Rule	Decision Tree	MLP	BF Tree	77.10
4	Decision Tree	LAD Tree	BF Tree	DTNB	77.01
5	Rep Tree	LAD Tree	BF Tree	Decision Table	76.90
6	Rep Tree	Decision Table	DTNB	LAD Tree	75.43
7	Bays	SVM	Part Rule	LAD Tree	75.16

The existence of missing values and outlier data is inevitable in problems with large datasets. Although the pre-processing functions decrease the effects of such deficiencies, they are not capable of completely resolving it. The findings reveal that Boosting and AdaBoost are more sensitive to the robustness of the dataset whereas Bagging is scarcely affected by such imperfections.

Another reason for the significant outperformance of Bagging may be related to the instability of the prediction procedure. As Breiman [19] notes, Bagging is highly capable of improving the accuracy of unstable prediction procedures where a small change in training data can result in large changes in the predictor/classifier structure. Neural networks, classification and regression trees are specified as unstable methods in which Bagging works well. The achievements also show the superiority of Bagging to the other two algorithms in similar circumstances.

We will now shift our focus to a new selection scheme that will be compared with the hybrid method proposed by Barak and Modarres [11]. In this paper, a Wrapper-GA algorithm was also developed for feature selection part. In this algorithm, a population is generated from candidate solutions, and in each iteration a new set of individuals is generated by means of mutation and crossover functions. The fitness of individuals in the current population is evaluated by a Decision Tree algorithm and the best individuals are selected as the next generation. The overview of the proposed Wrapper framework is presented in Appendix B [3].

In this methodology, after the pre-processing stage, the GA selects a set of features and then tests the prediction error of the selected features using the CART decision tree algorithm. Ten-fold cross-validation is applied in order to generate the training and test data. GA uses a uniform mutation with a probability of 0.1, tournament algorithm for selecting children (crossover) with rate of 0.8, and a population size of 60. The Decision Table is set as the fitness function. Additionally, the GA was run with varying parameter values (population size, mutation and crossover rates, etc.) in order to enhance the accuracy. Because the Decision Table algorithm is sensitive to missing data, these missed values are substituted with the average value of that column.

Table 13 gives the selected features with the best accuracy by Wrapper-GA for real return prediction.

Table 9
Average accuracy percentage of stock return prediction with diversity creation.

Base classifiers			Diversity	Fusion algorithm	Accuracy (%)
Decision Tree	LAD Tree	BF Tree	Bagging	DTNB	**80.05**
Decision Tree	LAD Tree	BF Tree	Boosting	DTNB	66.55
Decision Tree	LAD Tree	BF Tree	AdaBoost	DTNB	64.50
Decision Tree	LAD Tree	Rep Tree	Bagging	BF Tree	**83.65**
Decision Tree	LAD Tree	Rep Tree	Boosting	BF Tree	70.02
Decision Tree	LAD Tree	Rep Tree	AdaBoost	BF Tree	68.12
Decision Table	DTNB	Rep Tree	Bagging	Decision Tree	**82.23**
Decision Table	DTNB	Rep Tree	Boosting	Decision Tree	79.32
Decision Table	DTNB	Rep Tree	AdaBoost	Decision Tree	77.72
MLP	Part Rule	J48 Graph	Bagging	BF Tree	78.74
BF Tree	Bays	SVM	Bagging	LAD Tree	70.66
K-NN	SVM	Bayes	Bagging	Decision Tree	64.27
	Random Forest		–	–	73.05

Table 10
Prediction results of the best ensemble for real return prediction.

Accuracy: 83.65%	True very low	True high	True very high	True normal	True low	Class precision (%)
Pred. very low	14	1	0	1	3	73.16
Pred. high	0	48	2	3	0	90.57
Pred. very high	1	2	19	2	1	76.00
Pred. normal	0	9	1	56	1	83.67
Pred. low	2	1	1	1	31	86.11
Class recall	82.00%	78.69%	82.60%	88.88%	86.11%	

Table 11
Average accuracy percentage of stock risk prediction with diversity creation.

Base classifiers			Diversity	Fusion algorithm	Accuracy (%)
BF Tree	DTNB	LAD Tree	Bagging	Decision Table	**88.23**
BF Tree	DTNB	LAD Tree	Boosting	Decision Table	78.90
BF Tree	DTNB	LAD Tree	AdaBoost	Decision Table	80.27
BF Tree	Decision Tree	LAD Tree	Bagging	DTNB	**81.34**
BF Tree	Decision Tree	LAD Tree	Boosting	DTNB	78.56
BF Tree	Decision Tree	LAD Tree	AdaBoost	DTNB	80.12
BF Tree	Rep Tree	LAD Tree	Bagging	Decision Table	**82.46**
BF Tree	Rep Tree	LAD Tree	Boosting	Decision Table	66.51
BF Tree	Rep Tree	LAD Tree	AdaBoost	Decision Table	70.55
DTNB	Decision Tree	LAD Tree	Bagging	Decision Table	79.14
DTNB	Decision Table	Rep Tree	Bagging	LAD Tree	76.43
Part Rule	Bays	SVM	Bagging	LAD Tree	68.66
MLP	Decision Table	Part Rule	Bagging	BF Tree	80.56
K-NN	SVM	Bayes	Bagging	Decision Tree	69.71
	Random Forest		–	–	74.36

Table 12
Prediction results of the best ensemble for risk prediction.

Accuracy: 88.23%	true High	true Low	true Normal	Class precision (%)
Pred. high	246	3	8	95.72
Pred. low	14	388	32	89.40
Pred. normal	65	54	685	85.20
Class recall	75.69%	87.19%	94.48%	

The accuracy of real return prediction with the selected features provided in Table 13 was estimated as78.86 ± 0.94%. The findings reveal that if we use all of the specified features (given in Appendix A) for return prediction with the Decision Tree, the accuracy on test data is estimated as 76.5%. The reason for this improvement may be the diversity created by several iterations of the Decision Table with various types of features. Diversity creation as a critical factor in enhancing the accuracy of prediction models is widely discussed in Section 2.3. Fig. 4 illustrates the GA trend diagram with the Decision Table fitness function.

The proposed Wrapper-GA algorithm is also applied for feature selection in risk prediction. Although various fitness and error functions were used in GA, surprisingly, the achievements did not show improvements in prediction accuracy above 70%. Fig. 5 displays the improvement points of GA for risk prediction when the Decision Table and LAD Tree are used as GA evaluation function and error assessment of selected features, respectively.

Table 14 provides the selected features with the best accuracy of 68.26 ± 2.55 via Wrapper-GA for risk prediction.

Although the Wrapper-GA feature selection algorithm could improve the prediction accuracy compared to the case in which all the features are involved, the feature selection ensemble model developed by Barak and Modarres [11] outperformed Wrapper-GA in accuracy enhancement. Thus, its related features reported in Section 4.2 were used for the proposed hybrid prediction method of this paper.

A comparison between our method and similar studies is shown in Table 15. Different hybrid methods that have excellent

Table 13
Selected features for real return prediction with Wrapper-GA algorithm.

Average payment period	Current assets turnover	Predicted profit margin	Equity ratio
Quick ratio	EPS prediction	Profit margin growth rate	β coefficient
Return on equity (ROE)	Debt ratio	EPS growth percentage	percentage of net profit to sale
Prediction difference Percentage of EPS with the Real amount	Return on asset (after tax) ROA	Long-term debt to equity ratio	P/E
EPS cover			

Table 14
Selected features for risk prediction with Wrapper-GA algorithm.

Average payment period	Equity ratio	Quick ratio	Percentage of net profit to gross profit
Asses the loan usefulness	market return	EPS prediction	β coefficient
P/E	EPS cover percentage	Total income growth percentage	Book value
Efficiency	Debt coverage ratio	Total asset turnover	Fixed asset turnover
Current asset turnover	EPS growth percentage	long-term debt to equity ratio	return on equity (ROE)
Fixed asset return percentage	Current ratio	Percentage of net profit to sale	

Table 15
Comparison of the proposed fusion scheme versus other studies.

Author/year	Stock exchange	Input data	Base classifier	Feature selection	Hybrid model	The best accuracy (%)
Tsai, Lin, Yen and Chen [1]	Electronic Industry in Taiwan	19 Financial ratios and 11 Macroeconomic indicators	MLP- Cart - Logistic Regression	–	Bagging -Voting	66.67
Huang [2]	30 special companies in Taiwan	14 Financial ratios	SVR- GA	–	–	85- 76.71
Cheng, Chen and Lin [3]	Taiwan	10 Technical Indexes and 8 Macroeconomic indicators	PNN- C4.5- Rough Set	–	Hybrid	76
Huang, Yang and Chuang [4]	South-Korea and Taiwan	23 Technical Indexes	SVM- K-NN- Cart- Logistic Regression- Back Propagation	Wrapper	Voting	76.06 80.28
Tsai and Hsiao [5]	Taiwan	8 Fundamental Index and 11 Macroeconomic indicators	–	GA-PCA-Cart	Back Propagation	79
Tsai, Lu and Yen [6]	Taiwan	61 intangible assets value variable	MLP	PCA- Stepwise Regression- decision trees- association rules- GA	MLP	75
Barak and Modarres [7]	Return Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	Cart, Rep Tree, LAD Tree, ...	Function based Clustering	Hybrid	80.24
Barak and Modarres [7]	Risk Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	DTNB, BF Tree, LAD Tree, ...	Function based clustering	Hybrid	79.01
Current paper	Return Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	Cart, Rep Tree, LAD Tree, ...	Wrapper-GA	Hybrid	78.86
Current paper	Risk Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	DTNB, BF Tree, LAD Tree, ...	Wrapper-GA	Hybrid	70
Current paper	Return Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	Cart, Rep Tree, LAD Tree, ...	Function based clustering with diversity	Fusion	**83.65**
Current paper	Risk Forecasting in TSE-Iran	44 Financial ratios and Fundamental Index	DTNB, BF Tree, LAD Tree, ...	Function based clustering with diversity	Fusion	**88.23**

accuracy in return forecasting in different national stock exchanges were compared based on input data, base classifier, feature selection, hybrid prediction model and degree of accuracy, as follows: [5,2,59,60,61,62]

6. Conclusions and future research

In this paper, a study of fusion models based on the use of multiple diversity classifiers is presented for stock return and risk

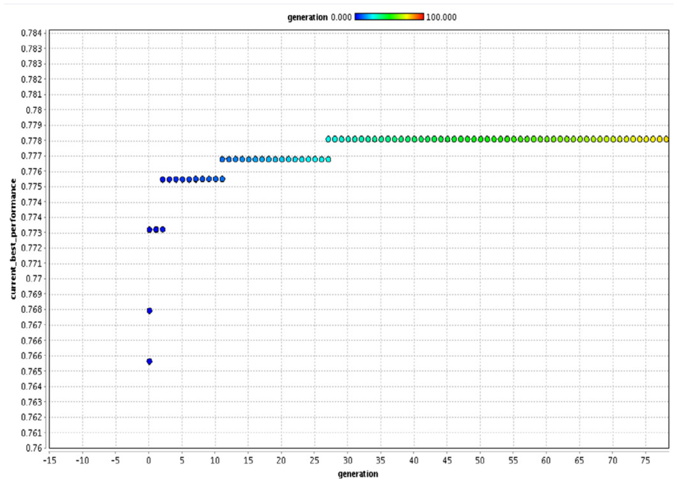

Fig. 4. Improvement points of GA with Decision Table fitness function for real return prediction.

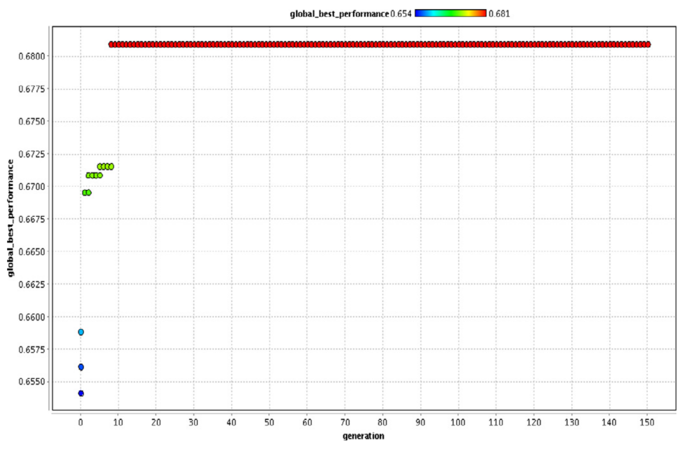

Fig. 5. Improvement points of GA with LAD Tree fitness function for risk prediction.

prediction. Bagging, Boosting and AdaBoost were applied as three diversity algorithms for generating a pool of classifiers. An empirical study was later undertaken on the Tehran Stock Exchange that compared the performance of diversity algorithms with different sets of classifiers in a fusion system. Bagging consistently outperformed the other two algorithms, regardless of the type of individual classifiers employed.

Almost all of the fusion strategies provided statistically significant improvements in performance over the best individual classifiers. Bagging performed well with Decision Trees in fusion systems that are stated as being unstable prediction methods. The limitation of this method is that collecting all of the fundamental data and information may be difficult for certain cases.

Future research directions of the paper include but are not limited to

1. Optimizing the parameters of classification algorithms using metaheuristics algorithms to improve the prediction results.
2. Predicting other important response variables (in addition to risk and return) such as liquidity [8].
3. Using technical features and textual information, in addition to fundamentals features, in order to use more comprehensive features and be able to predict the short term situations of stocks.
4. Customizing the proposed approach for the prediction of risk and return in a particular industry or investigating the accuracy of the procedure using data from other popular stock markets, such as the US stock market, which may result in new dimensions for this procedure.

Acknowledgments

The research was supported through the Czech Science Foundation (GACR) under project 15-23699S and through SP2016/11, a SGS research project of VSB-TU Ostrava, and furthermore by the European Social Fund in the framework of CZ.1.07/2.3.00/20.0296.

Appendix A: The full list of features for real return and risk prediction [11]

Table A.1
The full list of features for real return and risk prediction.

Category		Features
Financial ratio	Liquidity ratios	Current ratio, Quick ratio, Current assets ratio, Net working capital, Liquidity ratios
	Activity ratio	Average payment period, Current assets turn over, Fixed asset turnover, Total asset turnover
	Capitalization ratio	Equity ratio, Debt coverage ratio, Debt to total assets ratio, Debt to equity ratio, Long-term debt to equity ratio, Current debt to equity ratio.
	Profitability ratio	Percentage of net profit to sale, Percentage of operating profit to sale, Percentage of Gross profit to sale, Percentage of net profit to Gross profit, Return on asset (after tax) ROA, Return on equity (after tax) ROE, Working capital return percentage, Fixed assets return percentage, Assessment of loan usefulness
Stock pricing models	Capital Asset Pricing Model	r = return ration without risk β= stock beta coefficient (systematic risk) r_m = expected return from market
	Gordon Model	EPS, DPS, EPS prediction, EPS cover, Prediction difference percentage of EPS with the real amount, EPS growth ration in compare to the previous fiscal year.
	Campbell–Shiller Model	P/E, P/S
	Walter model	Stock cumulative profit
	Fama–French Model	Company's capital(investment), Stock book value, Stock market value
Company's loss and profit reports	Total predicted income (last income prediction in the current fiscal year),Total income growth % (total real income / (total real income - total predicted income)), Predicted Profit margins (last profit ratio / company's income in the current fiscal year), Profit margin growth rate (real profit margin / (real profit margin – predicted profit margin)) and Efficiency (Percent of daily trading volume / company's daily value in the before period).	

Appendix B. The Wrapper framework

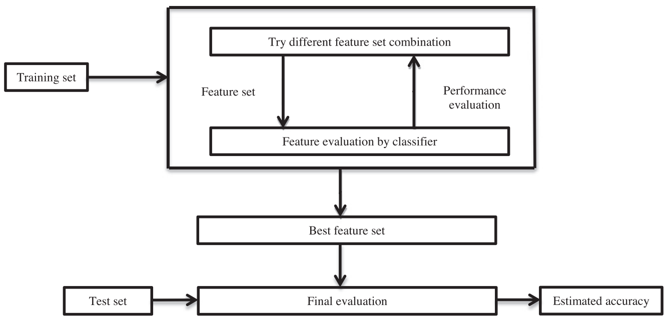

Fig. B1. The Wrapper framework.

References

[1] R. Cervelló-Royo, F. Guijarro, K. Michniuk, Stock market trading rule based on pattern recognition and technical analysis: forecasting the DJIA index with intraday data, Expert Syst. Appl. 42 (2015) 5963–5975.
[2] C.-F. Huang, A hybrid stock selection model using genetic algorithms and support vector regression, Appl. Soft Comput. 12 (2012) 807–818.
[3] S. Barak, J.H. Dahooie, T. Tichý, Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick, Expert Syst. Appl. 42 (2015) 9221–9235.
[4] D.E. Rapach, G. Zhou, Forecasting stock returns, Handbook of Economic Forecasting, 2, 2013, pp. 328–383.
[5] C.-F. Tsai, Y.-C. Lin, D.C. Yen, Y.-M. Chen, Predicting stock returns by classifier ensembles, Appl. Soft Comput. 11 (2011) 2452–2459.
[6] D. Enke, S. Thawornwong, The use of data mining and neural networks for forecasting stock market returns, Expert Syst. Appl. 29 (2005) 927–940.
[7] T. Hyup Roh, Forecasting the volatility of stock price index, Expert Syst. Appl. 33 (2007) 916–922.
[8] S. Barak, M. Abessi, M. Modarres, Fuzzy turnover rate chance constraints portfolio model, Eur. J. Oper. Res. 228 (2013) 141–147.
[9] S.-H. Cheng, A hybrid predicting stock return model based on bayesian network and decision tree, in: M. Ali, J.-S. Pan, S.-M. Chen, M.-F. Horng (Eds.), Modern Advances in Applied Intelligence, Springer International Publishing, 2014, pp. 218–227.
[10] S.-H. Cheng, A hybrid predicting stock return model based on logistic stepwise regression and CART algorithm, in: N.T. Nguyen, B. Trawiński, R. Kosala (Eds.), Intelligent Information and Database Systems, Springer International Publishing, 2015, pp. 624–633.
[11] S. Barak, M. Modarres, Developing an approach to evaluate stocks by forecasting effective features with data mining methods, Expert Syst. Appl. 42 (2015) 1325–1339.
[12] M.S. Haghighi, A. Vahedian, H.S. Yazdi, Creating and measuring diversity in multiple classifier systems using support vector data description, Appl. Soft Comput. 11 (2011) 4931–4942.
[13] S. Barak, S.S. Sadegh, Forecasting energy consumption using ensemble ARIMA–ANFIS hybrid algorithm, Int. J. Electr. Power Energy Syst. 82 (2016) 92–104.
[14] C.-F. Tsai, Y.-F. Hsu, D.C. Yen, A comparative study of classifier ensembles for bankruptcy prediction, Appl. Soft Comput. 24 (2014) 977–984.
[15] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[16] A.J.C. Sharky, N.E. Sharky, Combining diverse neural nets, Knowl. Eng. Rev. 12 (1997) 231–247.
[17] F. Pereira, T. Mitchell, M. Botvinick, Machine learning classifiers and fMRI: a tutorial overview, Neuroimage 45 (2009) S199–S209.
[18] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, machine learning, in: Proceedings of the Thirteenth International Conference, 1996, pp. 48–56.
[19] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[20] N.C. Oza, K. Tumer, Classifier ensembles: select real-world applications, Inf. Fusion 9 (2008) 4–20.
[21] L. Rokach, Taxonomy for characterizing ensemble methods in classification tasks: a review and annotated bibliography, Comput. Stat. Data Anal. 53 (2009) 4046–4072.
[22] S. Finlay, Multiple classifier architectures and their application to credit risk assessment, Eur. J. Oper. Res. 210 (2011) 368–378.
[23] D.F. de Oliveira, A.M.P. Canuto, M.C.P. de Souto, Use of multi-objective genetic algorithms to investigate the diversity/accuracy dilemma in heterogeneous ensembles, in: Proceedings of International Joint Conference on Neural Networks, IJCNN 2009, 2009, pp. 2339–2346.
[24] D. Wolpert, The supervised learning no-free-lunch theorems, in: Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications, 2001, pp. 25–42.
[25] B.V. Dasarathy, B.V. Sheela, A composite classifier system design: CONCEPTS and methodology, Proc. IEEE 67 (1979) 708–713.
[26] L. Rastrigin, R.H. Erenstein, Method of Collective Recognition, Energoizdat, Moscow, 1981.
[27] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17.
[28] L. Wang, J. Wu, Neural network ensemble model using PPR and LS-SVR for stock market forecasting, in: D.-S. Huang, Y. Gan, V. Bevilacqua, J. Figueroa (Eds.), Advanced Intelligent Computing, Springer, Berlin, Heidelberg, 2012, pp. 1–8.
[29] S. Lahmiri, Ensemble with radial basis function neural networks for Casablanca stock market returns prediction, in: Proceedings of the 2014 Second World Conference on Complex Systems (WCCS), 2014, pp. 469–474.
[30] J.R. Quinlan, C4. 5: programs for machine learning, Elsevier, 2014.
[31] B. Qian, K. Rasheed, Stock market prediction with multiple classifiers, Appl. Intell. 26 (2007) 25–33.
[32] L. Kuncheva, C. Whitaker, C. Shipp, R. Duin, Limits on the majority vote accuracy in classifier fusion, Pattern Anal. Appl. 6 (2003) 22–31.
[33] G. Fumera, F. Roli, A theoretical and experimental analysis of linear combiners for multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 942–956.
[34] M. Wozniak, Experiments on linear combiners, in: E. Pietka, J. Kawa (Eds.), Information Technologies in Biomedicine, Springer, Berlin, Heidelberg, 2008, pp. 445–452.
[35] N. Kourentzes, D.K. Barrow, S.F. Crone, Neural network ensemble operators for time series forecasting, Expert Syst. Appl. 41 (2014) 4235–4244.
[36] K. Tumer, J. Ghosh, Analysis of decision boundaries in linearly combined neural classifiers, Pattern Recognit. 29 (1996) 341–348.
[37] T. Ho, J.J. Hull, S. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 66–75.
[38] S. Ferreiro, B. Sierra, I. Irigoien, E. Gorritxategi, Data mining for quality control: burr detection in the drilling process, Comput. Ind. Eng. 60 (2011) 801–810.
[39] L. Kuncheva, C. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2003) 181–207.
[40] E.K. Tang, P.N. Suganthan, X. Yao, An analysis of diversity measures, Mach. Learn. 65 (2006) 247–271.
[41] A. Tsymbal, M. Pechenizkiy, P. Cunningham, Diversity in search strategies for ensemble feature selection, Inf. Fusion 6 (2005) 83–98.
[42] T. Windeatt, Diversity measures for multiple classifier system analysis and design, Inf. Fusion 6 (2005) 21–36.
[43] L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[44] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, Adv. Neural Inf. Process. Syst. 7 (1995) 231–238.
[45] Y. Freund, Boosting a weak learning algorithm by majority, Inf. Comput. 121 (1995) 256–285.
[46] R.E. Schapire, The boosting approach to machine learning: an overview, in: Proceedings of MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA, USA, 2001.
[47] J. Alceu S. Britto, R. Sabourin, L.E.S. Oliveira, Dynamic selection of classifiers – a comprehensive review, Pattern Recognit. 47 (2014) 3665–3680.
[48] C. Giraud-Carrier, Metalearning-a tutorial, in: Proceedings of the 7th International Conference on Machine Learning and Applications, 2008.
[49] R. Mousavi, M. Eftekhari, A new ensemble learning methodology based on hybridization of classifier ensemble selection approaches, Appl. Soft Comput. 37 (2015) 652–666.
[50] T.M. Mitchell, Machine Learning, McGraw-Hill, 1997.
[51] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997) 119–139.
[52] J. Cao, S. Kwong, R. Wang, A noise-detection based AdaBoost algorithm for mislabeled data, Pattern Recognit. 45 (2012) 4451–4465.
[53] T. Woloszynski, M. Kurzynski, P. Podsiadlo, G.W. Stachowiak, A measure of competence based on random classification for dynamic ensemble selection, Inf. Fusion 13 (2012) 207–213.
[54] T. Woloszynski, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, Pattern Recognit. 44 (2011) 2656–2668.
[55] T.F. Covões, E.R. Hruschka, Towards improving cluster-based feature selection with a simplified silhouette filter, Inf. Sci. 181 (2011) 3766–3782.
[56] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Syst. Appl. 42 (2015) 259–268.
[57] H. Yu, R. Chen, G. Zhang, A SVM stock selection model within PCA, Procedia Comput. Sci. 31 (2014) 406–412.
[58] P. Ou, H. Wang, Prediction of stock market index movement by ten data mining techniques, Mod. Appl. Sci. 3 (2009) 28–42.
[59] J.-H. Cheng, H.-P. Chen, Y.-M. Lin, A hybrid forecast marketing timing model based on probabilistic neural network, rough set and C4. 5, Expert Syst. Appl. 37 (2010) 1814–1820.
[60] C.-J. Huang, D.-X. Yang, Y.-T. Chuang, Application of wrapper approach and composite classifier to the stock trend prediction, Expert Syst. Appl. 34 (2008) 2870–2878.
[61] C.-F. Tsai, Y.-C. Hsiao, Combining multiple feature selection methods for stock prediction: union, intersection, and multi-intersection approaches, Decis. Support Syst. 50 (2010) 258–269.
[62] C.-F. Tsai, Y.-H. Lu, D.C. Yen, Determinants of intangible assets value: the data mining approach, Knowl.-Based Syst. 31 (2012) 67–77.