Information Fusion 16 (2014) 3–17

Contents lists available at SciVerse ScienceDirect

Information Fusion

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

A survey of multiple classifier systems as hybrid systems

Michał Woźniak [a,*], Manuel Graña [b], Emilio Corchado [c]

[a] Department of Systems and Computer Networks, Wroclaw University of Technology, Wroclaw, Poland
[b] Computational Intelligence Group, University of the Basque Country, San Sebastian, Spain
[c] Departamento de Informática y Automática, University of Salamanca, Salamanca, Spain

ARTICLE INFO

Article history:
Available online 29 April 2013

Keywords:
Combined classifier
Multiple classifier system
Classifier ensemble
Classifier fusion
Hybrid classifier

ABSTRACT

A current focus of intense research in pattern classification is the combination of several classifier systems, which can be built following either the same or different models and/or datasets building approaches. These systems perform *information fusion* of classification decisions at different levels overcoming limitations of traditional approaches based on single classifiers. This paper presents an up-to-date survey on multiple classifier system (MCS) from the point of view of Hybrid Intelligent Systems. The article discusses major issues, such as diversity and decision fusion methods, providing a vision of the spectrum of applications that are currently being developed.

1. Introduction

Hybrid Intelligent Systems offer many alternatives for unorthodox handling of realistic increasingly complex problems, involving ambiguity, uncertainty and high-dimensionality of data. They allow to use both *a priori* knowledge and raw data to compose innovative solutions. Therefore, there is growing attention to this multidisciplinary research field in the computer engineering research community. Hybridization appears in many domains of human activity. It has an immediate natural inspiration in the human biological systems, such as the Central Nervous System, which is a *de facto* hybrid composition of many diverse computational units, as discussed since the early days of computer science, e.g., by von Neumann [1] or Newell [2]. Hybrid approaches seek to exploit the strengths of the individual components, obtaining enhanced performance by their combination. The famous "*no free lunch*" theorem [3] stated by Wolpert may be extrapolated to the point of saying that there is no single computational view that solves all problems. Fig. 1 is a rough representation of the computational domains covered by the Hybrid Intelligent System approach. Some of them deal with the uncertainty and ambiguity in the data by probabilistic or fuzzy representations and feature extraction. Others deal with optimization problems appearing in many facets of the intelligent system design and problem solving, either following a nature inspired or a stochastic process approach. Finally, classifiers implementing the intelligent decision process are also subject to hybridization by various forms of combination. In this paper, we focus in this specific domain, which is in an extraordinary effervescence nowadays, under the heading of Multi-Classifier Systems (MCS). Referring to classification problems, Wolpert's theorem has an specific lecture: there is not a single classifier modeling approach which is optimal for all pattern recognition tasks, since each has its own domain of competence. For a given classification task, we expect the MCS to exploit the strengths of the individual classifier models at our disposal to produce the high quality compound recognition system overcoming the performance of individual classifiers. Summarizing:

- Hybrid Intelligent Systems (HIS) are free combinations of computational intelligence techniques to solve a given problem, covering al computational phases from data normalization up to final decision making. Specifically, they mix heterogeneous fundamental views blending them into one effective working system.
- Information Fusion covers the ways to combine information sources in a view providing new properties that may allow to solve better or more efficiently the proposed problem. Information sources can be the result of additional computational processes.
- Multi-Classifier Systems (MCS) focus on the combination of classifiers form heterogenous or homogeneous modeling backgrounds to give the final decision. MCS are therefore a subcategory of HIS.

[*] Corresponding author.
E-mail addresses: michal.wozniak@pwr.wroc.pl (M. Woźniak), ccpgrrom@gmail.com (M. Graña), escorchado@usal.es (E. Corchado).

1566-2535/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.inffus.2013.04.006

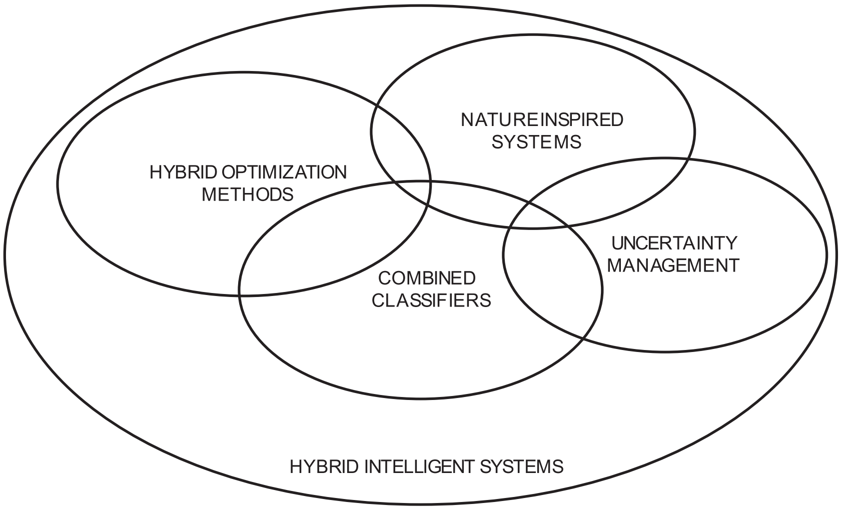

Fig. 1. Domains of hybrid intelligent systems.

Historical perspective. The concept of MCS was first presented by Chow [4], who gave conditions for optimality of the joint decision[1] of independent binary classifiers with appropriately defined weights. In 1979 Dasarathy and Sheela combined a linear classifier and one *k*-NN classifier [6], suggesting to identify the region of the feature space where the classifiers disagree. The *k*-NN classifier gives the answer of the MCS for the objects coming from the conflictive region and by the linear one for the remaining objects. Such strategy significantly decreases the exploitation cost of whole classifier system. This was the first work introducing a classifier selection concept, however the same idea was developed independently in 1981 by Rastrigin and Erenstein [7] performing first a feature space partitioning and, second, assigning to each partition region an individual classifier that achieves the best classification accuracy over it. Other early relevant works formulated conclusions regarding MCS 's classification quality, such as [8] who considered a neural network ensemble, [9] with majority voting applied to handwriting recognition, Turner in 1996 [10] showed that averaging outputs of an infinite number of unbiased and independent classifiers can lead to the same response as the optimal Bayes classifier, Ho [11] underlined that a decision combination function must receive useful representation of each classifier's decision. Specifically, they considered several method based on decision ranks, such as Borda count. Finally, the landmark works devoted introducing bagging [12] and boosting [13,14] which are able to produce strong classifiers [15], in the (*Probably Approximately Correct*) theory [16] sense, on the basis of the weak one. Nowadays MCS, are highlighted by review articles as a hot topic and promising trend in pattern recognition [17–21]. These reviews include the books by Kuncheva [22], Rokach [23], Seni and Edler [24], and Baruque and Corchado [25]. Even leading-edge general machine learning handbooks such as [26–28] include extensive presentations of MCS concepts and architectures. The popularity of this approach is confirmed by the growing trend in the number of publications shown in Fig. 2. The figure reproduces the evolution of the number of references retrieved by the application of specific keywords related to MCS since 1990. The experiment was repeated on three well known academic search sites. The growth in the number of publications has an exponential trend. The last entry of the

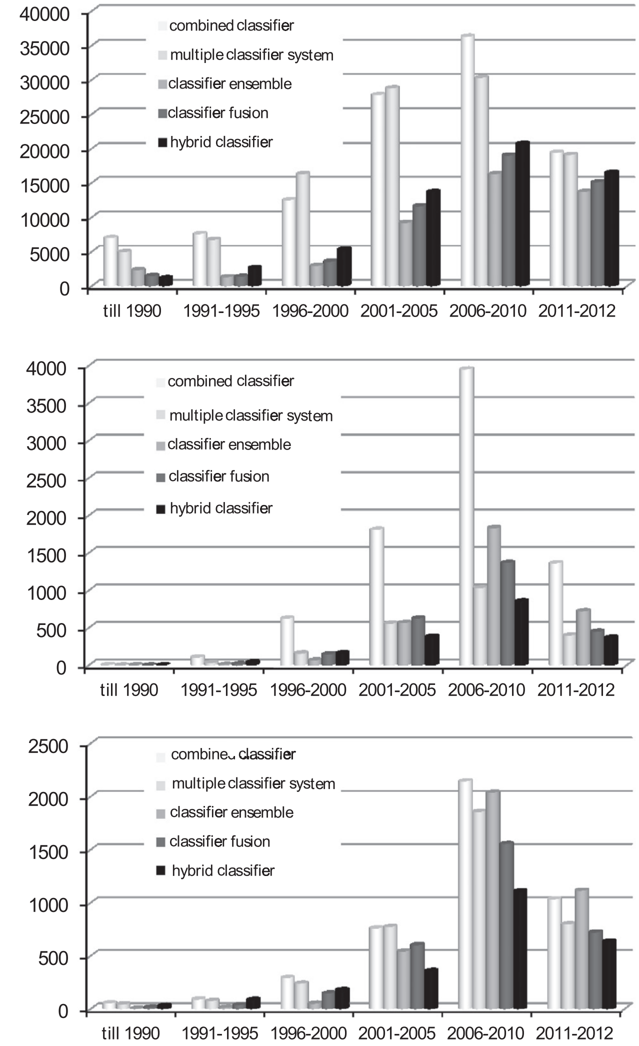

Fig. 2. Evolution of the number of publications *per* year ranges retrieved from the keywords specified in the plot legend. Each plot corresponds to searching site: the top to Google Scholar; the center to the Web of Knowledge, the bottom to Scopus. The first entry of the plots is for publications prior to 1990. The last entry is only for the last 2 years.

[1] We can retrace decision combination long way back in history. Perhaps the first worthy reference is the Greek democracy (meaning *government of the people*) ruling that full citizens have an equal say in any decision that affects their life. Greeks believed in the community wisdom, meaning that the rule of the majority will produce the optimal joint decision. In 1785 Condorcet formulated the *Jury Theorem* about the misclassification probability of a group of independent voters [5]], providing the first result measuring the quality of classifier committee.

plots corresponds to the last 2 years, and some of the keywords give as many references as in the previous 5 years.

Advantages. Dietterich [29] summarized the benefits of MCS: (a) allowing to filter out hypothesis that, though accurate, might be incorrect due to a small training set, (b) combining classifiers trained starting from different initial conditions could overcome the local optima problem, and (c) the true function may be impossible to be modeled by any single hypothesis, but combinations of hypotheses may expand the space of representable functions. Rephrasing it, there is widespread acknowledgment of the following advantages of MCS:

- MCS behave well in the two extreme cases of data availability: when we have very scarce data samples for learning, and when we have a huge amount of them at our disposal. In the scarcity case, MCS can exploit bootstrapping methods, such as bagging or boosting. Intuitive reasoning justifies that the worst classifier would be out of the selection by this method [30], e.g., by individual classifier output averaging [31]. In the event of availability of a huge amount of learning data samples, MCS allow to train classifiers on dataset's partitions and merge their decision using appropriate combination rule [20].
- Combined classifier can outperform the best individual classifier [32]. Under some conditions (e.g., majority voting by a group of independent classifiers) this improvement has been proven analytically [10].
- Many machine learning algorithms are *de facto* heuristic search algorithms. For example the popular decision tree induction method C4.5 [33] uses a greedy search approach, choosing the search direction according to an heuristic attribute evaluation function. Such an approach does not assure an optimal solution. Thus, the combined algorithm, which could start its work from different initial points of the search space, is equivalent to a multi-start local random search which increases the probability of finding an optimal model.
- MCS can easily be implemented in efficient computing environments such as parallel and multithreaded computer architectures [34]. Another attractive area of implementation solutions is distributed computing systems (i.e.: P2P, Grid or Cloud computing) [35,36], especially when a database is partitioned for privacy reasons [37] so that partial solutions must be computed on each partition and only the final decision is available as the combination of the networked decision.
- Wolpert stated that each classifier has its specific competence domain [3], where they overcome other competing algorithms, thus it is not possible to design a single classifier which outperforms another ones for each classification tasks. MCS try to select always the local optimal model from the available pool of trained classifiers.

System structure. The general structure of MCS is depicted in Fig. 3 following a classical pattern recognition [38] application structure. The most informative or discriminant features describing the objects are input to the classifier ensemble, formed by a set of *complementary* and *diverse* classifiers. An appropriate fusion method combines the individual classifier outputs optimally to provide the system decision. According to Ho [39], two main MCS design approaches can be distinguished. On one hand, the so-called *coverage optimization* approach tries to cover the space of possible models by the generation of a set of mutually complementary classifiers whose combination provides optimal accuracy. On the other hand, the so-called *decision optimization* approach concentrates on designing and training an appropriate decision combination function over a set of individual classifier given in advance [40].The main issues in MCS design are:

- System topology: How to interconnect individual classifiers.
- Ensemble design: How to drive the generation and selection of a pool of valuable classifiers.
- Fuser design: How to build a decision combination function (fuser) which can exploit the strengths of the selected classifiers and combine them optimally.

2. System topology

Fig. 4 illustrates the two canonical topologies employed in MCS design. The overwhelming majority of MCS reported in the literature is structured in a parallel topology [22]. In this architecture, each classifier is feed the same input data, so that the final decision of the combined classifier output is made on the basis of the outputs of the individual classifiers obtained independently. Alternatively, in the serial (or conditional) topology, individual classifiers are applied in sequence, implying some kind of ranking or ordering over them. When the primary classifier cannot be trusted to classify a given object, e.g., because of the low support/confidence in its result, then the data is feed to a secondary classifier [41,42], and so on, adding classifiers in sequence. This topology is adequate when the cost of classifier exploitation is important, so that the primary classifier is the computationally cheapest one, and secondary classifiers have higher exploitation cost [43]. This model can be applied to classifiers with the so-called reject option as well [44]. In [45] the first classifier in the pipeline gives an estimation of the certainty of the classification, so that uncertain data samples are sent to a second classifier, specialized in difficult instances. We notice the similarity of such approach to the ordered set of rules [46] or decision list [47], when we consider each rule as the classifier.

A very special case of sequential topology is the Adaboost introduced by Freund and Schapire in 1995 [48], widely applied in data

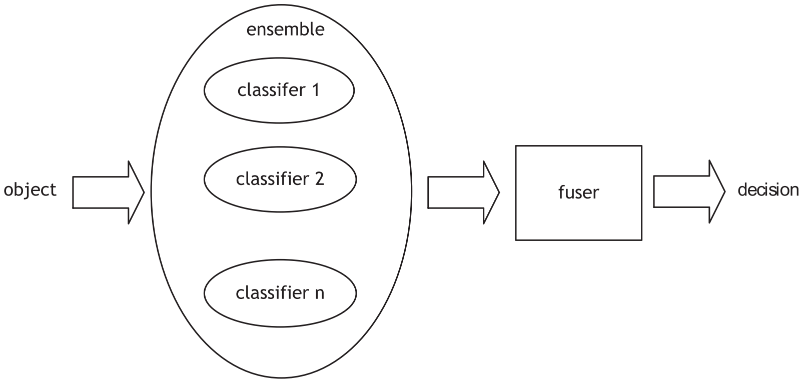

Fig. 3. Overview of multiple classifier system.

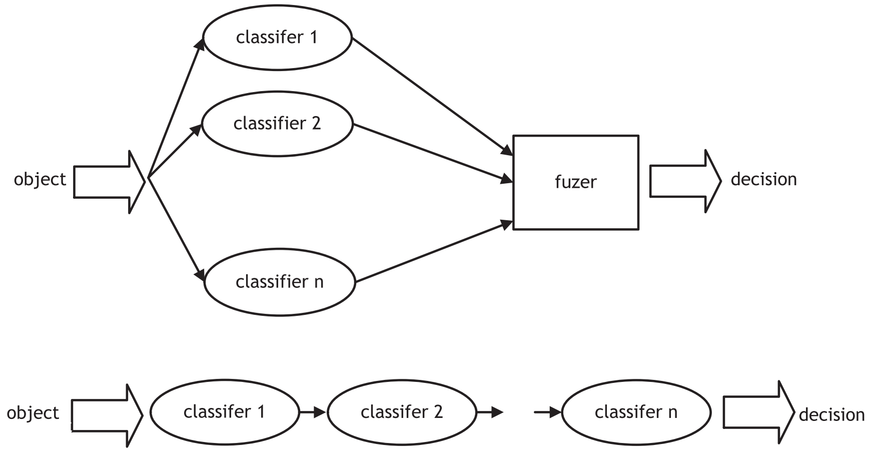

Fig. 4. The canonical topologies of MCSs: parallel (top) and serial (bottom).

mining problems [49]. The goal of boosting is to enhance the accuracy of any given learning algorithm, even weak learning algorithms with an accuracy slightly better than chance. Shapire [50] showed that weak learners can be *boosted* into a strong learning algorithm by sequentially focusing on the subset of the training data that is *hardest* to classify. The algorithms performs training of the weak learner multiple times, each time presenting it with an updated distribution over the training examples. The distribution is altered so that *hard* parts of the feature space have higher probability, i.e. trying to achieve a hard margin distribution. The decisions generated by the weak learners are combined into a final single decision. The novelty of Adaboost lies in the adaptability of the successive distributions to the results of the previous weak learners, thus the name **Ada**ptive**Boost**. In the words of Kivinen et al. [51], AdaBoost finds a new distribution that is closest to the old one but taking into consideration the restriction that the new distribution must be orthogonal to the mistake vector of the current weak learner.

3. Ensemble design

Viewing MCS as a case of robust software [52–55], diversity arises as the guiding measure of the design process. Classifier ensemble design aims to include mutually complementary individual classifiers which are characterized by high diversity and accuracy [56]. The emphasis from the Hybrid Intelligent System point of view is in building MCS from components following different kinds of modeling and learning approaches, expecting an increase in diversity and a decrease in classifier output correlation [57]. Unfortunately, the problem of how to measure classifier diversity is still an open research topic. Brown et al. [58] notice that we can ensure diversity using implicit or explicit approaches. Implicit approaches include techniques of independent generation of individual classifiers, often based on random techniques, while explicit approaches focus on the optimization of a diversity metric over a given ensemble line-up. In this second kind of approaches, individual classifier training is performed conditional to the previous classifiers with the aim of exploiting the strengths of valuable members of classifier pool. This section discusses some diversity measures, and the procedures followed to ensure diversity in the ensemble.

3.1. Diversity measures

For regression problems, the variance of the outputs of ensemble members is a convenient diversity measure, because it was proved that the error of a compound model based on a weighted averaging of individual model outputs can be reduced according to increasing diversity [56,59]. Brown et al. [60] showed a functional relation between diversity and individual regressor accuracy, allowing to control the bias-variance tradeoff systematically.

For classification problems such theoretical results have not been proved yet, however many diversity measures have been proposed till now. On the one hand, it is intuitive that increasing diversity should lead to the better accuracy of the combined system, but there is no formal proof of this dependency [61], as confirmed by the wide range of experimental results presented, e.g., in [62]. In [53] authors decomposed the error of the classification by majority voting into individual accuracy, *good* and *bad* diversities. The *good* diversity has positive impact on ensemble error reduction, whereas the *bad* diversity has the opposite effect. Sharkley et al. [55] proposed a hierarchy of four levels of diversity according to the answer of the majority rule, coincident failures, and possibility of at least one correct answer of ensemble members. Brown et al. [58] argue that this hierarchy is not appropriate when the ensemble diversity varies between feature subspaces. They formulated the following taxonomy of diversity measures:

- Pairwise measures averaging a measure between each classifier pair in an ensemble, such as Q-statistic [58], kappa-statistics [63], disagreement [64] and double-fault measure [61,65].
- Non-pairwise diversity measures comparing outputs of a given classifier and the entire ensemble, such as Kohavi–Wolpert variance [66], a measure of inter-rater (inter-classifier) reliability [67], the entropy measure [68], the measure of difficulty [8], generalized diversity [52], and coincident failure diversity [69].

The analysis of several diversity measures [70] relating them to the concept of classifiers' margin, showed their limitations and the source of confusing empirical results. They relate the classifier selection to a NP-complete matrix cover problem, implying that ensemble design in fact a quite difficult combinatorial problem. Diversity measures usually employ the most valuable sub-ensemble in *ensemble pruning* processes [71]. To deal with the high computational complexity of ensemble pruning, several hybrid approaches have been proposed such as heuristic techniques [72,73], evolutionary algorithms [74,75], reinforcement learning [76], and competitive cross-validation techniques [77]. For classification tasks, the cost of acquiring feature values (which could be interpreted as the price for examination or time required to collect the data for decision making) can be critical. Some authors take it into consideration during the component classifier selection step [78,79].

3.2. Ensuring diversity

According to [22,38] we can enforce the diversity of a classifier pool by the manipulation of either individual classifier inputs, outputs, or models.

3.2.1. Diversifying input data

This diversification strategy assumes that classifiers trained on different (disjoint) input subspaces become complementary. Three general strategies are identified:

1. Using different data partitions.
2. Using different sets of features.
3. Taking into consideration the local specialization of individual classifiers.

Data partitions They may be compelled by several reasons, such as data privacy, or the need to learn over distributed data chunks stored in different databases [80–82]. Regarding data privacy, we should notice that using distributed data may come up against legal or commercial constraints which do not allow sharing raw datasets and merging them into a common repository [37]. To ensure privacy we can train individual classifiers on each database independently and merge their outputs using hybrid classifier principles [83]. The distributed data paradigm is strongly connected with the big data analysis problem [84]. A huge database may impede to deliver trained classifiers under specified time constraints, imposing to resort to sampling techniques to obtain manageable dataset partitions. A well known approach is cross-validated committee which requires to minimize overlapping of dataset partitions [56]. Providing individualized train datasets for each classifier is convenient in the case of shortage of learning examples. Most popular techniques, such as bagging [12] or boosting [14,19,64,85], have their origin in bootstrapping [13]. These methods try to ascertain if a set of weak classifier may produce a strong one. Bagging applies sampling with replacement to obtain independent training datasets for each individual classifier. Boosting modifies the input data distribution perceived by each classifier from the results of classifiers trained before, focusing on difficult samples, making the final decision by a weighted voting rule.

Data features May be selected to ensure diversity training of a pool of classifiers. The Random Subspace [86,87] was employed for several types of the individual classifiers such as decision tree (Random Forest) [88], linear classifiers [89], or minimal distance classifier [90,91]. It is worth pointing out the interesting propositions dedicated one-class classifier presented by Nanni [92] or an hierarchical method of ensemble forming, based on feature space splitting and then assigning two-class classifiers (i.e. Support Vector Machines) locally presented in [93,94]. Attribute Bagging [95] is a wrapper method that establishes the appropriate size of a feature subset, and then creates random projections of a given training set by random selection of feature subsets. The classifier ensemble are train on the basis of the obtained set.

Local specialization It is assumed for classifier selection, selecting the best single classifier from a pool of classifiers trained over each partition of the feature space. It gives the MCS answer for all objects included in the partition [7]. Some proposals assume classifier local specialization, providing only locally optimal solutions [38,96–98,72], while others divide the feature space, selecting (or training) a classifier for each partition. Static and dynamic approaches are distinguished:

- Static classifier selection [99]: the relation between region of competence and assigned classifier is fixed. Kuncheva's Clustering and Selection algorithm [100] partitions the feature space by a clustering algorithm, and selects the best individual classifier for each cluster according to its local accuracy. Adaptive Splitting and Selection algorithm in [101] partitions the feature space and assigns classifiers to each partition into one integrated process. The main advantage of AdaSS is that the training algorithm considers an area contour to determine the classifier content and, conversely, that the region shapes adapt to the competencies of the classifiers. Additionally, the majority voting or more sophisticated rules are proposed as combination method of area classifiers [102]. Lee et al. [103] used the fuzzy entropy measure to partition the feature space and select the relevant features with good separability for each of them.
- Dynamic classifier selection: the competencies of the individual classifiers are calculated during classification operation [104–107]. There are several interesting proposals which extend this concept, e.g., by using preselected committee of the individual classifier and making the final decision on the basis of a voting rule [108]. In [109,110] authors propose dynamic ensemble selection based on the original competence measure using classification of so-called random reference classifier.

Both static [111–113] and dynamic [114–116] classifier specialization are widely used for data stream classification.

3.2.2. Diversifying outputs

MCS diversity can be enforced by the manipulation of the individual classifier outputs, so that an individual classifier is designed to classify only some classes in the problem.

The combination method should restore the whole class label set, e.g., a multi-class classification problem can be decomposed into a set of binary classification problems [117,118]. The most popular propositions of two-class classifier combinations are: OAO (*one-against-one*) and OAA (*one-against-all*)[119], where at least one predictor relates to each class. The model that a given object belongs to a chosen class is tested against the alternative of the feature vector belonging to any other class. In the OAA method, a classifier is trained to separate a chosen class from the remaining ones. OAA returns class with maximum support. In more general approaches, the combination of individual outputs is made by finding the closest class, in some sense, to the code given by the outputs of the individual classifiers. ECOC (*Error Correcting Output Codes*) model was proposed by Dieterich and Bakiri [118], who assumed that a set of classifiers produces sequence of bits which is related to code-words during training. The ECOC points at the class with the smallest Hamming distance to its codeword. Passerini et al. showed advantages of this method over traditional ones for the ensemble of support vector machines [120].

Recently several interesting propositions on how to combine the binary classifiers were proposed. Wu et al. [121] used pairwise coupling, Friedman employed Max-Win rule [122], Hüllermeier proposed the adaptive weighted voting procedure [123]. A comprehensive recent survey of binary classifier ensembles is [124]. It worth mentioning the one-class classification model which is the special case of binary classifier trained in the absence of counterexamples. Its main goal is to model normality in order to detect anomaly or outliers from the target class [125]. To combine such classifiers the typical methods developed for binary ones are used [126] but it is worth mention the work by Wilk and Wozniak where authors restored multi-class classification task using a pool of one-class classifiers and the fuzzy inference system [127]. The combination methods dedicated the one-class classifiers still await a proper attention [128].

3.2.3. Diversifying models

Ensembles with individual classifiers based on different classification models take advantage of the different biases of each classifier model [3]. However, the combination rule should be carefully chosen. We can combine the class labels but in the case of continuous outputs we have to normalize them, e.g., using fuzzy approach [127]. We could use the different versions of the same model as well, because many machine learning algorithms do not guarantee to find the optimal classifier. Combining the results of various initializations may give good results. Alternatively, a pool of classifiers can be produced by noise injection. Regarding neural networks [129] it is easy to train pools of networks where each of them is trained starting from randomly chosen initial weights. Regarding decisions tree we can choose randomly the test for a given node among the possible tests according to the value of a splitting criterion.

4. Fuser design

Some works consider the answers from a given Oracle as the reference combination model [130]. The Oracle is an abstract combination model, built such that if at least one of the individual classifiers provides the correct answer, then the MCS committee outputs the correct class too. Some researches used the Oracle in comparative experiments to provide a performance upper bound for classifier committee [10] or information fusion methods [131]. A simple example shows the risks of the Oracle model: assume we have two classifiers for a binary class problem, a random one and the other that always returns the opposite decision; hence the Oracle will always return the correct answer. As a consequence the Oracle model does not fit in the Bayesian paradigm. Raudys [132] noticed that Oracle is a kind of quality measure of a given individual classifier pool. Let us systematize methods of classifier fusion, which on the one hand could use class labels or support function, on the other hand combination rules could be given or be the results of training. The taxonomy of decision fusion strategies is depicted in Fig. 5.

4.1. Class label fusion

Early algorithms performing fusion of classifier responses [9,10,61] only implemented majority voting schemes in three main versions [22]:

- unanimous voting, so that the answer requires that all classifiers agree,
- simple majority, so that the answer is given if majority is greater than half the pool of classifiers,
- majority voting, taking the answer with the highest number of votes.

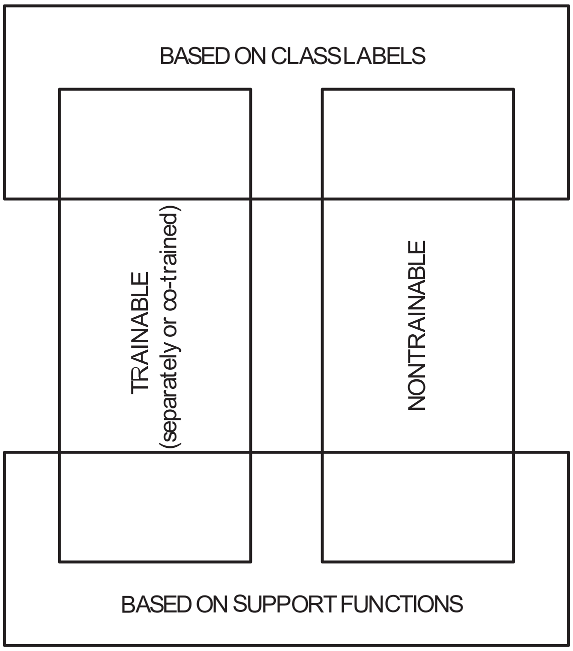
BASED ON CLASS LABELS
TRAINABLE (separately or co-trained)
NONTRAINABLE
BASED ON SUPPORT FUNCTIONS

Fig. 5. A taxonomy of fusing strategies for the combination of MCS individual decisions.

The expected error of majority voting (for independent classifiers with the same quality) was estimated in 1794 according to Bernoulli's equation, proven as the Condorcet Jury Theorem [5]. Later works focused on the analytically derived classification performance of combined classifiers hold only when strong conditions are met [8] so that they are not useful from practical point of view. Alternative voting methods weight differently the decisions coming from different committee members [22,133]. The typical architecture of combined classifier based on class labels is presented in the left diagram of Fig. 6. In [134] authors distinguished the types of weighted voting according to the classifier, both to the classifier and the class, and, finally, to features values, the classifier and the class. Anyway, no one of these models can improve over the Oracle. To achieve that we need additional information, such as the feature values [132,135,136] as depicted in the right diagram of Fig. 6.

4.2. Support function fusion

Support function fusion system architecture is depicted in Fig. 7. Support functions provide a score for the decision taken by an individual classifier. The value of a support function is the estimated likelihood of a class, computed either as a neural network output, *a posteriori* probability, or fuzzy membership function. First to be mentioned, the Borda count [11] computes an score for each class on the basis of its ranking by each individual classifier. The most popular form of support function is the *a posteriori* probability [26], produced by the probabilistic models embodied by the classifiers [137–139]. There are many works following this approach, such as the optimal projective fuser of [140], the combination of neural networks outputs according to their accuracy [141], and Naïve Bayes as the MCS combination method [142].

Some analytical properties and experimental evaluations of aggregating methods were presented in [10,31,143,144]. The aggregating methods use simple operators such as supremum or the mean value. They do not involve learning. However, they have little practical applicability because of the hard conditions imposed by them [145]. The main aggregating advantage is that it counteracts over-fitting of individual classifiers. According to [134], the following types of weighted aggregation can be identified depending on: (a) only the classifier id, (b) the classifier and the feature vector, (c) on the classifier and the class, and (d) on the classifier, the class, and the feature vector. For two-class recognition problems only the last two types of aggregation allow to produce compound classifier which may improve the Oracle. For many-class problems, it is possible to improve the Oracle [131] using any of these aggregation methods. Finally, another salient approach is the *mixture of experts* [146,147] which combines classifier outputs using so-called input dependent gating function. Tresp and Taniguchi [148] proposed a linear function for this fuser model, and Cheeseman [149] proposed a mixture of Gaussian.

4.3. Trainable Fuser

Fuser weight selection can be treated as a specific learning process [31,136]. Shlien [150] used Dempster and Shafer's theory to reach a consensus on the weights to combine decision trees. Wozniak [151] trained the fuser using perceptron-like learning, evolutionary algorithm [152,153]. Zheng used data envelopment

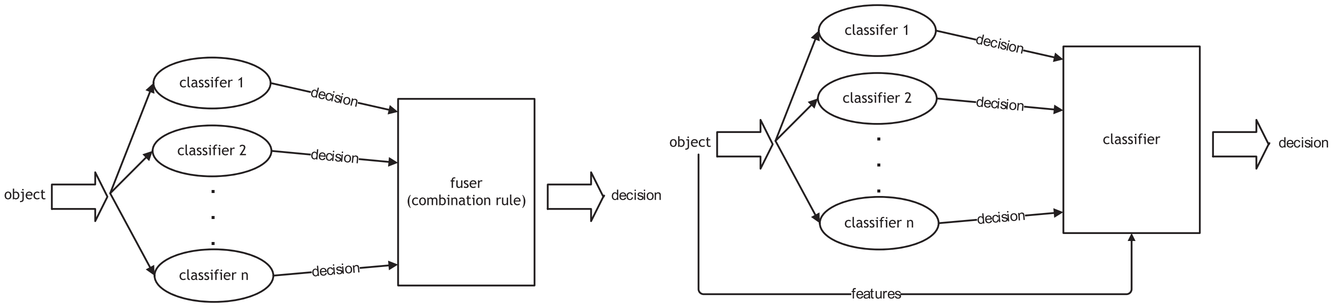

Fig. 6. Architecture of the MCS making decision on the basis of class label fusion only (left diagram). The right diagram corresponds to a MCS using additional information from the feature values.

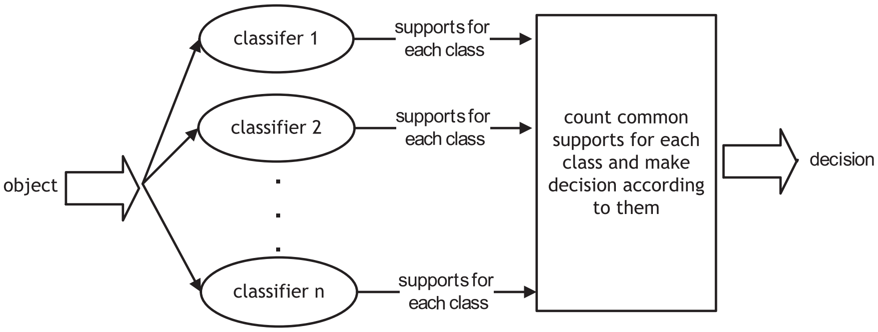

Fig. 7. Architecture of the MCS which computes the decision on the basis of support function combination.

analysis [154]. Other fuser trainable methods may be strictly related to ensemble pruning methods, when authors use some heuristic search algorithm to select the classifier ensemble, as [72,141] according to the chosen fuser.

We have to mention the group of combination methods built from pools of heterogenous classifiers, i.e. using different classification models, such as stacking [155]. This method trains combination block using individual classifier outputs presented during classification of the whole training set. Most of the combination methods do not take into consideration possible relations among individual classifiers. Huang and Suen [156] proposed Behavior-Knowledge Space method which aggregates the individual classifiers decision on the basis of the statistical approach.

5. Concept Drift

Before entering the discussion of practical applications we consider a very specific topic of real life relevance which is known as Concept Drift in knowledge engineering domains, or non-stationary processes in signal processing and statistics domains. Most of the conventional classifiers do not take into consideration this phenomenon. Concept Drift means that the statistical dependencies between object features and its classification may change in time, so that future data may be badly processed if we maintain the same classification, because the object category or its properties will be changing. Concept drift occurs frequently in real life [157]. MCS are specially well suited to deal with Concept Drift.

Machine learning methods in security applications (like spam filters or IDS/IPS) [158] or decision support systems for marketing departments [159] require to take into account new training data with potentially different statistical properties [116]. The occurrence of Concept Drift decreases the true classification accuracy dramatically. The most popular approaches are the Streaming Ensemble Algorithm (SEA) [111] and the Accuracy Weighted Ensemble (AWE) [160]. Incoming data are collected in data chunks, which are used to train new models. The individual classifiers evaluation is done on their accuracy on the new data. The best performing classifiers are selected to constitute the MCS committee in the next time epoch. As the decision rule, the SEA uses a majority voting, whereas the AWE uses a weighted voting strategy. Kotler et al. present the Dynamic Weighted Majority (DWM) algorithm [114] which modifies the decision combination weights and updates the ensemble according to number of incorrect decisions made by individual classifiers. When a classifier weight is too small, then it is removed from the ensemble, a new classifier is trained and added to the ensemble in its place.

A difficult problem is drift detection, which is the problem of deciding that the Concept Drift has taken place. The current research direction is to propose an additional binary classifier giving the decision to rebuild the classifiers. The drift detector can be based on changes in the probability distribution of the instances [161–163] or classification accuracy [164,165]. Not all classification algorithms dealing with concept drift require drift detection, because they can adjust the model to incoming data [166][?].

6. Applications

Reported applications of classifier ensembles have grown astoundingly in the recent years due to the increase in computational power allowing training of large collections of classifiers in practical application time constraints. A recent review appears in [18]. Sometimes the works combine diverse kinds of classifiers, so-called heterogeneous MCS. Homogeneous MCS, such as Random Forest (RF), are composed of classifiers of the same kind. In the works revised below, basic classifiers are Multi-Layer Perceptron (MLP), k-Nearest Neighbor (kNN), Radial Basis Function (RBF), Support Vector Machines (SVM), Probabilistic Neural Networks (PNNs), and Maximum Likelihood (ML) classifiers.

We review in this section recent applications to remote sensing data, computer security, financial risk assessment, fraud detection, recommender systems, and medical computer aided diagnosis.

6.1. Remote sensing

The main problems addressed by MCS in remote sensing domains are the land cover mapping and change detection. Land cover mapping consists in the identification of materials that are in the surface of the area being covered. Depending on the application, a few general classes may be identified, i.e. vegetation, water, buildings, roads, or a more precise classification can be required, i.e. identifying tree or crop types. Applications include agriculture, forestry, geology, urban planning, infrastructure degradation assessment. Change detection consists in the identification of places where the land cover has changed in time, it implies the computation over time series of images. Change detection may or may not be based on previous or separate land cover maps. Remote sensing classification can be done on a variety of data sources, sometimes performing fusion of different data modalities. Optical data has better interpretability by humans, but land is easily occluded by weather conditions, i.e. cloud formations. Hyperspectral sensing provides high-dimensional data at each image pixel, with high spectral resolution. Synthetic Aperture Radar (SAR) is not affected by weather or other atmospheric conditions, so that observations are better suited for continuous monitoring of seasonally changing land covers. SAR can provide also multivariate data from varying radar frequencies. Other data sources are elevation maps, and other ancillary information, such as the measurements of environmental sensors.

6.1.1. Land cover mapping

Early application of MCS to land cover mapping consisted in overproducing a large set of classifiers and searching for the optimal subset [38,65,167]. To avoid the combinatorial complexity, the approach performs clustering of classifier error, aggregating similar classifiers. The approach was proven to be optimal under some conditions on the classifiers. Interestingly, testing was performed on multi-source data, composing the pixel's feature vector of joining multi-spectral with radar data channels, to compute the land cover map. The MCS was heterogenous, composed of MLP, RBF, and PNN.

The application of RF to processing remote sensing data has been abundant in the literature. It has been applied to estimate land cover on Landsat data over Granada, Spain [168] and multi-source data in a Colorado mountainous area [169]. Specifically, Landsat Multi-Spectral, elevation, slope and aspect data are used as input features. The RF approach is able to successfully fuse these inhomogeneous informations. Works on hyperspectral images acquired by the HyMap sensor have been addressed to build vegetation thematic maps [170], comparing RF and decision tree-based Adaboost, as well as two feature selection methods: the out-of-bag and a best-first search wrapper feature subset selection method. Diverse feature subsets are tested, and the general conclusion is that tree ecotopes are better discriminated than grass ecotopes. Further work with RF has been done assessing the uncertainty in modeling the distribution of vegetation types [171], performing classification on the basis of environmental variables, in an approach that combines spatial distribution modeling by spatial interpolation, using sequential Gaussian simulation and the clustering of species into vegetation types. Dealing with labeled data scarcity, there are methods [172] based on the combination of RF and the enrichment of the training dataset with artificially generated samples in order to increase classifier diversity, which is applied to Landsat multispectral data. Artificial data is generated from the Gaussian modeling of the data distribution. The application of RF to SAR multitemporal data aims to achieve season invariant detection of several classes of land cover, i.e. grassland, ceral, forest, etc. [173]. RF performed best, with lowest spatial variability. Images were coregistered and some model portability was tested, where the model trained on one SAR image was applied on other SAR images of the same site obtained at different times. The success of RF for remote sensing images has prompted the proposal of an specific computational environment [174].

Ensambles of SVM have been also applied to land cover map. Indeed, the ground truth data scarcity has been attacked by an active learning approach to semi-supervised SVM training [175]. The active learning approach is based on the clustering of the unlabeled data samples according to the clustering of the SVM outputs on the current training dataset. Samples with higher membership coefficient are added to the corresponding class data, and the classifier is retrained in an iterative process. These semi-supervised SVM are combined in a majority voting ensemble and applied to the classification SPOT and Landsat optical data. Land cover classification in the specific context of shallow waters has the additional difficulties of the scattering, refraction and reflection effects introduced by the water cover. A robust process combines a parallel and a serial architecture [176], where initial classification results obtained by SVM are refined in a second SVM classifier and the final result is given by a linear combination of two ensembles of SVM classifiers and a minimum distance classifier. Besides, the system estimates the water depth by a bathymetry estimation process. The approach is applied to Landsat images for the estimation of coral population in coastal waters. Polarimetric SAR data used for the classification of Boreal forests require an ensemble of SVM [177]. Each of the SVM is specifically tuned to a class, with specific feature selection process. Best results are obtained when multitemporal data is used, joining two images from two different seasons (summer and winter) and performing the feature selection and training on the joint data vectors.

6.1.2. Change detection

Early application of MCS to land cover change detection was based on non-parametric algorithms, specifically MLP, k-NN, RBF, and ML classifiers [178,179], where classifier fusion was performed either by majority voting, Bayesian average and maximum *a posteriori* probability. Testing data were Thematic Mapper multispectral images, and the Synthetic Aperture Radar (SAR) of Landsat 5 satellite. Recent works on change detection in panchromatic images with MCS follow three different decision fuser strategies: majority voting, Dempster-Shafer evidence theory, and the Fuzzy Integral [180]. The sequential process of the images previous to classification includes pan-sharpening of the multi-temporal images, coregistration, raw radiometric change detection by image subtraction and automatic thresholding, and a final MCS decision computed on the multi-spectral data and the change detection data obtained from the various pan-sharpening approaches.

6.2. Computer security

Computer security is at the core of most critical services nowadays, from universities, banking, companies, communication. Secure information processing is a growing concern, and the machine learning approaches are trying to provide predictive solutions that may allow to avoid the negative impact of such attacks. Here we introduce some of the problems, with current solutions proposed from the MCS paradigm.

6.2.1. Distributed denial of service

Distributed denial of service (DDoS) are among the most threatening attacks that an Internet Service Provider may face. Distributed service providers, such as military applications, e-healthcare and e-governance can be very sensitive to this type of attacks, which can produce network performance degradation, service unavailability, and revenue loss. There is a need for intelligent systems able to discriminate legitimate flash crowds from an attack. A general architecture for automatic detection of DDoS attacks is needed where the attack detection may be performed by a MCS. The MCS constituent classifiers may be ANNs trained with robust learning algorithms, i.e. Resilient Back Propagation (RBP). Specifically, a boosting strategy is defined on the ensemble of RBP trained ANNs, and a Neyman Pearson approach is used to make the final decision [181]. This architecture may be based on Sugeno Adaptive Neuro-Fuzzy Inference Systems (ANFIS) [182]. A critical issue of the approach is the need to report validation results, which can only be based on recorded real life DDoS attacks. There are some public available datasets to perform and report these results. However, results reported on these datasets may not be informative of the system performance on new attacks which may have quite different features. This is a pervasive concern in all security applications of machine learning algorithms.

6.2.2. Malware

Malicious code, such as trojans, virus, spyware, detection by anti-virus approaches can only be performed after some instance of the code has been analyzed finding some kind of signature, therefore some degree of damage has already been done. Predictive approaches based on Machine Learning techniques may allow anticipative detection at the cost of some false positives. Classifiers learn patterns in the known malicious codes extrapolating to yet unseen codes. A taxonomy of such approaches is given in [183]. describing the basic code representation by byte and opcode *n*-grams, strings, and others like portable executable features. Feature selection processes, such as the Fisher score, are applied to find the most informative features. Finally, classifiers tested in this problem include a wide variety of MCS combining diverse base classifiers with all standard fuser designs. Results have been reported that MCS overcome other approaches, are better suitable for active learning needed to keep the classifiers updated and tuned to the changing malicious code versions.

6.2.3. Intrusion detection

Intrusion Detection and Intrusion Prevention deal with the identification of intruder code in a networked environment via the monitoring of communication patterns. Intruder detection performed as an anomaly detection process allows to detect previously unseen patterns, at the cost of false alarms, contrary to signature based approaches. The problem is attacked by modular MCS whose compounding base classifiers are one-class classifiers built by the Parzen window probability density estimation approach [128]. Each module is specialized in a specific protocol or network service, so that different thresholds can be tuned for each module allowing some optimization of the false alarm rate. On the other hand, Intrusion Prevention tries to impede the execution of the intruder code by fail-safe semantics, automatic response and adaptive enforcement. An approach relies on the fact that Instruction Set Randomization prevents code injection attacks, so that detected injected code can be used for adaptation of the anomaly classifier and the signature-based filtering [184]. Clustering of *n*-grams is performed to obtain a model of the normal communication behavior which is accurate allowing zero-day detection of worm infection even in the case of low payload or slow penetration [185]. The interesting proposed hybrid intrusion detection was presented in [186], where decision trees and support vector machines are combined as a hierarchical hybrid intelligent system model.

6.2.4. Wireless sensor networks

Wireless sensor networks (WSNs) are collections of inexpensive, low power devices deployed over a geographical space for monitoring, measuring and event detection. Anomalies in the WSN can be due to failures in software or hardware, or to malicious attacks compelling the sensors to bias or drop their information and measurements. Anomaly detection in WSN is performed using an ensemble of binary classifiers, each tuned on diverse parameters and built following a different approach (Average, autorregresive, neural network, ANFIS). The decision is made by a weighted combination of the classifiers outputs [187].

6.3. Banking, credit risk, fraud detection

In the current economical situation, the intelligent processing of financial information, the assessing of financial or credit risks, and related issues have become a prime concern for society and for the computational intelligence community. Developing new tools may allow to avoid in the future the dire problems faced today by society. In this section we review some of the most important issues, gathering current attempts to the deal with them.

6.3.1. Fraud detection

Fraud detection involves identifying fraud as soon as possible after it has been perpetrated. Fraud detection [188] is big area of research and applications of machine learning, which has provided techniques to counteract fraudsters in credit card fraud, money laundering, telecommunications fraud, and computer intrusion. MCS have been also applied successfully in this domain. A key task is modeling the normal behavior in order to be able to establish suspicion scores for outliers. Probabilistic networks are specific one-class classifiers that are well suited to this task, and bagging of probabilistic networks has been proposed as a general tool for fraud detection because the MCS approach improves the robustness of the normal behavior modeling [189].

6.3.2. Credit card fraud

Specific works on credit card fraud detection use real-life data of transactions from an international creditcard operation [190]. The exploration of the sensitivity to the ratio of fraud to non-fraud of the random undersampling approach to deal with unbalanced class sizes is required to validate the approaches. Comparing RF against SVM and logisti regression [190], RF was the best performer in all experimental conditions as measured by almost all performance measurements. Other approaches to this problem include a bagged ensemble of SVM tested on a british card application approval dataset [191].

6.3.3. Stock market

Trade based stock market manipulation try to influence the stock values simply by buying and then selling. It is difficult to detect because rules for detection quickly become outdated. An innovative research track is the use of peer-group analysis for trade stock manipulation detection, based on the detection of outliers whose dynamic behavior separates from that of the previously similar stock values, its peers [192]. Dynamic clustering allows to track in time the evolution of the community of peers related to the stocks under observation, and outlier detection techniques are required to detect the manipulation events.

6.3.4. Credit risk

Credit risk prediction models seek to predict whether an individual will default on a loan or not. It is greatly affected by the unavailability, scarcity and incompleteness of data. The application of machine learning to this problem includes the evaluation of bagging, boosting, stacking as well as other conventional classifiers over three benchmarking datasets, including sensitivity to noise added to the attributes [193]. Another approach for this problem is the Error Trimmed Boosting (ETB) [194] which has been tested over a privative dataset provided by a company. ETB consists in the iterative selection of subsets of samples based on their error under the current classifier. An special case of credit risk is enterprise risk assessment which has a strong economic effect due to the financial magnitude of the entities involved. To deal with this problem a combination of bagging and random subspace feature selection using SVM as the base classifier has been developed and tested. The resulting method has increased diversity improving results over a dataset provided by the Bank of China [195]. Bankruptcy prediction is a dramatic special case of credit risk. Ensemble systems with diversity ensured by genetic algorithm based selection of component classifiers is proposed in [196] for bankruptcy prediction in South Korean firms. The prediction of failure of dotcom companies has been a matter of research since the bubble explosion after the year 2000. Tuning a hybrid of PNN, MLP and genetic programming classifiers over a set of features selected applying a *t*-test and *F*-test for relevance to the categorical variable has given some solutions [197]. The same approach is reported in [198] to detect fraud in the financial statement of big companies.

6.3.5. Financial risks

Uncertainty in the financial operations is identified with the financial risks such as credit, business, investment, and operational risks. Financial distress can be detected by clustering and MCS in four different combination models. Clustering is performed by classical SOM and *k*-means algorithms and used to partition the data space prior to MCS training [199]. Experimental framework for the evaluation of financial risk assessment models, giving a specific performance measures allow the exploration of computational solutions to these problems [200]. Several conventional classifiers and MCS have been tested in this framework using a large pool of datasets. Bank performance and bankruptcy prediction is addressed using a widely heterogenous MCS including PNN,RBF, MLP, SVM, CART trees, and a fuzzy rule system. The effect of PCA initial dimensionality reduction is also tested [201]. The effect of feature construction from previous experience and *a priori* information in the efficiency of classifiers for early warning of bank failures is reported in [202].

6.3.6. New fraud trends

Prescription fraud has been identified as a cause of substantial monetary loss in health care systems, it consists in the prescription of unnecessary medicaments. The research works need to real life data from a large multi-center medical prescription database [203]. The authors use a novel distance based on data-mining approach in a system which is capable of self-learning by regular updates. The system is designed to perform on-line risky prescription detection followed by off-line expert evaluation.

A new brand of frauds appear in the online gaming and lotteries, i.e. intended for money laundering, whose detection is dealt with a mixture of supervised and unsupervised classifiers [204]. To be adaptive to fraudster evolving strategies, it is required to emphasize online learning, and online cluster detection. Fraud in telecommunication systems involving usage beyond contract specifications is dealt with in [205] by a preprocessing, clustering and classification pipeline. Clustering has been found to improve classification performance, and boosted trees are the best performing approach. The analysis of social networks by means of MCS may allow the detection of fraud in automobile insurance, consisting in staging traffic accidents and issuing fake insurance claims to their general or vehicle insurance company [206].

6.4. Medicine

Medicine is a big area of application of any innovative computational approach, dealing with massive amounts of data in some instances, and with very imprecise or ambiguous data in other situations. The range of applications is quite big, so here we only give a scrap of all the current problems and approaches related with the MCS paradigm. In Medicine, a specific research area since the inception of Artificial Intelligence is the construction of Computer Aided Diagnosis (CAD) systems or Clinical Decision Support Systems (CDSS) [207], which involve as the final step some kind of classifier predicting the subject's disease or normal status. In CDSS development, there are several steps such as the definition of the sensor providing the data, the preprocessing of the data to normalize it and remove noise, the selection of features, and the final selection of the classifier.

6.4.1. Coronary diseases

A recent instance of CDSS is the application to cardiovascular disease diagnosis of an heterogenous collection of classifiers, composed of SVM, bayesian networks and ANN [208] finding ten new biomarkers. In this AptaCDSS-E process starts with the use of an aptamer biochip scanning protein expression levels which is the input to physician taking the decisions afterwards. Feature selection is performed by an ANOVA analysis. Doctor decisions are stored for system retraining. Classifier combination is done by majority voting or hierarchical fusion. Many CAD systems related with coronary diseases are based on the information provided by the electrocardiogram (ECG), so that many of them rely on the features extracted from them. Coronary artery disease is a broad term that encompasses any condition that affects the heart. It is a chronic disease in which the coronary arteries gradually harden and narrow, there have approaches to provide CAD for this condition, such as the use of a mixture of three ANNs for the prediction of coronary artery disease [209]. The dysfunction or abnormality of one or more of the heart four valves is called valvular heart disease. Its diagnosis is performed by neural network ensembles in [209,210] over features selected by a correlation analysis with the categorical variable. Two separate ANNs are trained to identify myocardial infarction on training sets with different statistics regarding the percentage of patients in [211]. The network specialized in healthy controls is applied to the new data, if the output is below a threshold the subject is deemed healthy, otherwise the disease-specific network is applied to decide.

6.4.2. Proteomics

Proteins are said to have a common fold if they have the same major secondary structure in the same arrangement and with the same topology. Machine learning techniques have been proposed for three-dimensional protein structure prediction. Early approaches consisted in hybrid systems, such as the ANN, statistical classifier and case base reasoning classifier combined by majority voting of [212]. For instance, an ensemble of K-local hyperplanes based on random subspace and feature selection has been tested [213], where feature selection is done according to distance to the class centroids. A recent approach is the MarFold [214] combining by majority voting three margin-based classifiers for protein fold recognition: the adaptive local hyperplane (ALH), the k-neighborhood ALH and the SVM.

6.4.3. Neuroscience

In the field of Neurosciences, the machine learning approach is gaining widespread acceptation. It is used for the classification of image data searching for predictive non-invasive biomarkers that may allow early or prodromal diagnosis of a number of degenerative diseases which have increasing impact in the society due to the aging of populations around the world. Diverse MCS approaches have been applied to structural MRI data, specifically for the classification of Alzheimer disease patients, such as an RVM based two stage pipeline [45], variations of Adaboost [215], hybridizations of kernel and Dendritic Computing approaches [216]. Classifier Ensembles have been applied to the classification of fMRI data [217,218] and its visual decoding [219], which is the reconstruction of the visual stimuli from the fMRI data.

6.5. Recommender systems

Nowadays, recommender systems are the focus of intense research [220]. They try to help consumers to select the product that may be interesting for them based on their previous searches and transactions, but such systems are expanding beyond typical sales. They are used to predict which mobile telephone subscribers are in risk of switching to another provider, or to advice conference organizers about assigning papers to peer reviewers [221]. Burke [222] proposed hybrid recommender systems combining two or more recommendation techniques to improve performance avoiding the drawbacks of an individual recommender. Similar observations were confirmed by Balabanovic et al. [223] and Pazzani [224] who demonstrated that hybrid method recommentations improve collaborative and content-based approaches.

There are several interesting works which apply the hybrid and combined approach to recommender systems. Jahrer and Töscher [225] demonstrated the advantage of ensemble learning applied to the combination of different collaborative filtering algorithms on the Netix Prize dataset. Porcel et al. [226] developed an hybrid fuzzy recommender system to help disseminate information about research resources in the field of interest of a user. Claypool et al. [227] performed a linear combination of the ratings obtained from individual recommender systems into one final recommendation, while Pazzani proposed to use a voting scheme [224]. Billsus and Pazzani [228] selected the best recommendation on the basis of a recommendation quality metric as the level of confidence while Tran and Cohen [229] preferred an individual which is the most consistent with the previous ratings of the user. Kunaver et al. [230] proposed Combined Collaborative Recommender based on three different collaborative recommender techniques. Goksedef and Gundoz-Oguducu [231] combined the results of several recommender techniques based on Web usage mining.

7. Final remarks

We have summarized the main research streams on multiple classifier systems, also known in the literature as combined classifier or classifier ensemble. Such hybrid systems are the focus of intense research recently, so fruitful that our review could not be exhaustive. Key issues related to the problem under consideration are classifier diversity and methods of classifier combination.

The diversity is believed to provide improved accuracy and classifier performance. Most works try to obtain maximum diversity by different means: introducing classifier heterogeneity, bootstrapping the training data, randomizing feature selection, randomizing subspace projections, boosting the data weights, and many combinations of these ideas. Nowadays, the diversity hypothesis has not been fully proven, either theoretically or empirically. However, the fact is that MCSs show in most instances improved performance, resilience and robustness to high data dimensionality and diverse forms of noise, such as labeling noise.

The there are several propositions how to combine the classifier outputs, what was presented in this work, nonetheless we point out that classifier combination is not the only way to produce hybrid classifier systems. We envisage further possibilities of hybridization such as:

- Merging the raw data from different sources into one repository and then train the classifier.
- Merging the raw data and a *prior* expert knowledge (e.g., learning sets and human expert rules to improve rules on the basis of incoming data).
- Merging a prior expert knowledge and classification models returned by machine learning procedures.

For such a problem we have to take into consideration issues related to data privacy, computational and memory efficiency.

Acknowledgments

We would like to thank the anonymous reviewers for their diligent work and efficient efforts. We are also grateful to the Editor-in-Chief, Prof. Belur V. Dasarathy, who encouraged us to write this survey for this prestigious journal.

MichałWoźniak was supported by The Polish National Science Centre under the Grant No. N519 576638 which is being realized in years 2010–2013.

References

[1] J. Neumann, The Computer and the Brain, Yale University Press, New Haven, CT, USA, 1958.
[2] A. Newell, Intellectual issues in the history of artificial intelligence, in: F. Machlup, U. Mansfield (Eds.), The Study of Information: Interdisciplinary Messages, John Wiley & Sons Inc., New York, NY, USA, 1983, pp. 187–294.
[3] 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.
[4] C.K. Chow, Statistical independence and threshold functions, IEEE Transactions on Electronic Computers EC-14 (1) (1965) 66–68.
[5] L. Shapley, B. Grofman, Optimizing group judgmental accuracy in the presence of interdependencies, Public Choice 43 (3) (1984) 329–333.
[6] B.V. Dasarathy, B.V. Sheela, A composite classifier system design: concepts and methodology, Proceedings of the IEEE 67 (5) (1979) 708–713.
[7] L. Rastrigin, R.H. Erenstein, Method of Collective Recognition, Energoizdat, Moscow, 1981.
[8] L. Hansen, P. Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (10) (1990) 993–1001, http://dx.doi.org/10.1109/34.58871.
[9] L. Xu, A. Krzyzak, C. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Transactions on Systems, Man and Cybernetics 22 (3) (1992) 418–435.
[10] K. Tumer, J. Ghosh, Analysis of decision boundaries in linearly combined neural classifiers, Pattern Recognition 29 (2) (1996) 341–348.
[11] T. Ho, J.J. Hull, S. Srihari, Decision combination in multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1) (1994) 66–75.
[12] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.
[13] R. Schapire, The strength of weak learnability, Machine Learning 5 (2) (1990) 197–227.
[14] Y. Freund, Boosting a weak learning algorithm by majority, Information Computing 121 (2) (1995) 256–285.
[15] M. Kearns, U. Vazirani, An Introduction to Computational Learning Theory, MIT Press, Cambridge, MA, USA, 1994.
[16] D. Angluin, Queries and concept learning, Machine Learning 2 (4) (1988) 319–342.
[17] A. Jain, R. Duin, M. Jianchang, Statistical pattern recognition: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.
[18] N. Oza, K. Tumer, Classifier ensembles: select real-world applications, Information Fusion 9 (1) (2008) 4–20.
[19] R. Polikar, Ensemble based systems in decision making, IEEE Circuits and Systems Magazine 6 (3) (2006) 21–45.
[20] R. Polikar, Ensemble learning, Scholarpedia 3 (12) (2008) 2776.
[21] L. Rokach, Taxonomy for characterizing ensemble methods in classification tasks: a review and annotated bibliography, Computational Statistics and Data Analysis 53 (12) (2009) 4046–4072.
[22] L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[23] L. Rokach, Pattern Classification Using Ensemble Methods, Series in Machine Perception and Artificial Intelligence, World Scientific, 2010.

[24] G. Seni, J. Elder, Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions, Morgan and Claypool Publishers, 2010.

[25] B. Baruque, E. Corchado, Fusion Methods for Unsupervised Learning Ensembles, Springer Verlag New York, Inc., 2011.

[26] R. Duda, P. Hart, D. Stork, Pattern Classification, second ed., Wiley, New York, 2001.

[27] E. Alpaydin, Introduction to Machine Learning, second ed., The MIT Press, 2010.

[28] C. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[29] T. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, Heidelberg, 2000, pp. 1–15.

[30] G. Marcialis, F. Roli, Fusion of face recognition algorithms for video-based surveillance systems, in: G.L. Foresti, C. Regazzoni, P. Varshney (Eds.), 2003, pp. 235–250.

[31] S. Hashem, Optimal linear combinations of neural networks, Neural Networks 10 (4) (1997) 599–614.

[32] R. Clemen, Combining forecasts: a review and annotated bibliography, International Journal of Forecasting 5 (4) (1989) 559–583.

[33] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Series in Machine Learning, Morgan Kaufman Publishers, 1993.

[34] T. Wilk, M. Wozniak, Complexity and multithreaded implementation analysis of one class-classifiers fuzzy combiner, in: E. Corchado, M. Kurzynski, M. Wozniak (Eds.), Hybrid Artificial Intelligent Systems, Lecture Notes in Computer Science, vol. 6679, Springer, Berlin/Heidelberg, 2011, pp. 237–244.

[35] T. Kacprzak, K. Walkowiak, M. Wozniak, Optimization of overlay distributed computing systems for multiple classifier system – heuristic approach, Logic Journal of IGPL, doi:10.1093/jigpal/jzr020.

[36] K. Walkowiak, Anycasting in connection-oriented computer networks: models, algorithms and results, International Journal of Applied Mathematics and Computer Sciences 20 (1) (2010) 207–220.

[37] R. Agrawal, R. Srikant, Privacy-preserving data mining, SIGMOD Records 29 (2) (2000) 439–450.

[38] G. Giacinto, F. Roli, G. Fumera, Design of effective multiple classifier systems by clustering of classifiers, in: Proceedings of the 15th International Conference on Pattern Recognition, 2000, vol. 2, 2000, pp. 160–163.

[39] T. Ho, Complexity of classification problems and comparative advantages of combined classifiers, in: Proceedings of the First International Workshop on Multiple Classifier Systems, MCS '00, Springer-Verlag, London, UK, 2000, pp. 97–106.

[40] F. Roli, G. Giacinto, Design of Multiple Classifier Systems, World Scientific Publishing, 2002.

[41] L. Lam, Classifier combinations: implementations and theoretical issues, in: Proceedings of the First International Workshop on Multiple Classifier Systems, MCS '00, Springer-Verlag, London, UK, 2000, pp. 77–86.

[42] A.F.R. Rahman, M.C. Fairhurst, Serial combination of multiple experts: a unified evaluation, Pattern Analysis and Applications 2 (1999) 292–311.

[43] G. Fumera, I. Pillai, F. Roli, A two-stage classifier with reject option for text categorisation, 5th International Workshop on Statistical Techniques in Pattern Recognition (SPR 2004), vol. 3138, Springer, Lisbon, Portugal, 2004, pp. 771–779.

[44] P. Bartlett, M. Wegkamp, Classification with a reject option using a hinge loss, Journal of Machine Learning Research 9 (2008) 1823–1840.

[45] M. Termenon, M. Graña, A two stage sequential ensemble applied to the classification of alzheimer's disease based on MRI features, Neural Processing Letters 35 (1) (2012) 1–12.

[46] P. Clark, T. Niblett, The CN2 induction algorithm, Machine Learning 3 (4) (1989) 261–283.

[47] R. Rivest, Learning decision lists, Machine Learning 2 (3) (1987) 229–246.

[48] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139, http://dx.doi.org/10.1006/jcss.1997.1504.

[49] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowledge and Information Systems 14 (1) (2008) 1–37, http://dx.doi.org/10.1007/s10115-007-0114-2.

[50] R. Schapire, The strength of weak learnability, Machine Learning 5 (2) (1990) 197–227, http://dx.doi.org/10.1023/A:1022648800760.

[51] J. Kivinen, M.K. Warmuth, Boosting as entropy projection, in: Proceedings of the Twelfth Annual Conference on Computational Learning Theory, 1999. <http://dl.acm.org/citation.cfm?id=307424>.

[52] D. Partridge, W. Krzanowski, Software diversity: practical statistics for its measurement and exploitation, Information and Software Technology 39 (10) (1997) 707–717.

[53] G. Brown, L. Kuncheva, "good" And "bad" diversity in majority vote ensembles, in: Proceedings MCS 2010, pp. 124–133.

[54] M. Smetek, B. Trawinski, Selection of heterogeneous fuzzy model ensembles using self-adaptive genetic algorithms, New Generation Computing 29 (2011) 309–327.

[55] A.J.C. Sharkey, N. Sharkey, Combining diverse neural nets, Knowledge Engineering Review 12 (3) (1997) 231–247.

[56] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, Advances in Neural Information Processing Systems 7 (1995) 231–238.

[57] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, Machine Learning: ECML 2001 (2001) 576–587.

[58] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, Information Fusion 6 (1) (2005) 5–20.

[59] N. Ueda, R. Nakano, Generalization error of ensemble estimators, in: Proceedings of IEEE International Conference on Neural Networks, Washington, USA, 1996, pp. 90–95.

[60] G. Brown, J. Wyatt, P. Tiňo, Managing diversity in regression ensembles, Journal of Machine Learning Research 6 (2005) 1621–1650.

[61] L. Kuncheva, C. Whitaker, C. Shipp, R. Duin, Limits on the majority vote accuracy in classifier fusion, Pattern Analysis and Applications 6 (2003) 22–31.

[62] Y. Bi, The impact of diversity on the accuracy of evidential classifier ensembles, International Journal of Approximate Reasoning 53 (4) (2012) 584–607.

[63] D. Margineantu, T. Dietterich, Pruning adaptive boosting, in: Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, pp. 211–218.

[64] D. Skalak, The sources of increased accuracy for two proposed boosting algorithms, in: Proceedings of the American Association for Arti Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop, 1996, pp. 120–125.

[65] G. Giacinto, F. Roli, Design of effective neural network ensembles for image classification purposes, Image Vision Computing 19 (9-10) (2001) 699–707.

[66] R. Kohavi, D. Wolpert, Bias plus variance decomposition for zero-one loss functions, in: ICML-96, 1996.

[67] J. Fleiss, J. Cuzick, The reliability of dichotomous judgments: unequal numbers of judgments per subject, Applied Psychological Measurement 4 (3) (1979) 537–542.

[68] P. Cunningham, J. Carney, Diversity versus quality in classification ensembles based on feature selection, in: Proceedings of the 11th European Conference on Machine Learning, ECML '00, Springer-Verlag, London, UK, 2000, pp. 109–116.

[69] C. Shipp, L. Kuncheva, Relationships between combination methods and measures of diversity in combining classifiers, Information Fusion 3 (2) (2002) 135–148.

[70] E.K. Tang, P.N. Suganthan, X. Yao, An analysis of diversity measures, Machine Learning 65 (1) (2006) 247–271.

[71] G. Martinez-Mu/ noz, D. Hern/'andez-Lobato, A. Suarez, An analysis of ensemble pruning techniques based on ordered aggregation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 245–259.

[72] D. Ruta, B. Gabrys, Classifier selection for majority voting, Information Fusion 6 (1) (2005) 63–81.

[73] R. Banfield, L. Hall, K. Bowyer, W. Kegelmeyer, Ensemble diversity measures and their application to thinning, Information Fusion 6 (1) (2005) 49–62.

[74] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artificial Intelligence 137 (1-2) (2002) 239–263.

[75] B. Gabrys, D. Ruta, Genetic algorithms in classifier fusion, Applied Soft Computing 6 (4) (2006) 337–347.

[76] I. Partalas, G. Tsoumakas, I. Vlahavas, Pruning an ensemble of classifiers via reinforcement learning, Neurocomputing 72 (7–9) (2009) 1900–1909.

[77] Q. Dai, A competitive ensemble pruning approach based on cross-validation technique, Knowledge-Based Systems (0) (2012), http://dx.doi.org/10.1016/j.knosys.2012.08.024.

[78] Y. Peng, Q. Huang, P. Jiang, J. Jiang, Cost-sensitive ensemble of support vector machines for effective detection of microcalcification in breast cancer diagnosis, in: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Lecture Notes in Computer Science, vol. 3614, Springer, Berlin/Heidelberg, 2005, pp. 483–493.

[79] K. Jackowski, B. Krawczyk, M. Woniak, Cost-sensitive splitting and selection method for medical decision support system, in: H. Yin, J.A. Costa, G. Barreto (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2012, Lecture Notes in Computer Science, vol. 7435, Springer, Berlin Heidelberg, 2012, pp. 850–857.

[80] W. Du, Z. Zhan, Building decision tree classifier on private data, in: Proceedings of the IEEE International Conference on Privacy, Security and Data Mining – Volume 14, CRPIT '14, Australian Computer Society, Inc., Darlinghurst, Australia, 2002, pp. 1–8.

[81] B. Krawczyk, M. Wozniak, Privacy preserving models of k-NN algorithm, in: R. Burduk, M. Kurzynski, M. Wozniak, A. Zolnierek (Eds.), Computer Recognition Systems 4, Advances in Intelligent and Soft Computing, vol. 95, Springer, Berlin/Heidelberg, 2011, pp. 207–217.

[82] Y. Lindell, B. Pinkas, Secure multiparty computation for privacy-preserving data mining, IACR Cryptology ePrint Archive 2008 (2008) 197.

[83] K. Walkowiak, S. Sztajer, M. Wozniak, Decentralized distributed computing system for privacy-preserving combined classifiers – modeling and optimization, in: B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, B. Apduhan (Eds.), Computational Science and Its Applications – ICCSA 2011, Lecture Notes in Computer Science, Vol. 6782, Springer, Berlin/Heidelberg, 2011, pp. 512–525.

[84] A. Pavlo, E. Paulson, A. Rasin, D. Abadi, D. DeWitt, S. Madden, M. Stonebraker, A comparison of approaches to large-scale data analysis, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, SIGMOD '09, ACM, New York, NY, USA, 2009, pp. 165–178.

[85] R.E. Schapire, The boosting approach to machine learning: an overview, in: MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA, USA, 2001.

[86] T. Ho, Random decision forests, in: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)–Volume 1, ICDAR '95, IEEE Computer Society, Washington, DC, USA, 1995, pp. 278–.

[87] T. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 832–844.

[88] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[89] M. Skurichina, R. Duin, Bagging, boosting and the random subspace method for linear classifiers, Pattern Analysis and Applications 5 (2) (2002) 121–135.

[90] G. Tremblay, R. Sabourin, P. Maupin, Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm, in: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1–Volume 01, ICPR '04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 208–.

[91] S. Bay, Nearest neighbor classification from multiple feature subsets, Intelligent Data Analysis 3 (3) (1999) 191–209.

[92] L. Nanni, Letters: Experimental comparison of one-class classifiers for online signature verification, Neurocomputing 69 (7–9) (2006) 869–873.

[93] D. Tao, X. Tang, X. Li, X. Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Transactions on Pattern Analysis Machine Intelligence 28 (7) (2006) 1088–1099.

[94] K. Ting, J. Wells, S. Tan, S. Teng, G. Webb, Feature-subspace aggregating: ensembles for stable and unstable learners, Machine Learning 82 (2011) 375–397.

[95] R. Bryll, R. Gutierrez-Osuna, F. Quek, Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, Pattern Recognition 36 (6) (2003) 1291–1302.

[96] Y. Baram, Partial classification: the benefit of deferred decision, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8) (1998) 769–776.

[97] L. Cordella, P. Foggia, C. Sansone, F. Tortorella, M. Vento, A cascaded multiple expert system for verification, in: Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 1857, Springer, Berlin/Heidelberg, 2000, pp. 330–339.

[98] K. Goebel, W. Yan, Choosing classifiers for decision fusion, in: Proceedings of the Seventh International Conference on Information Fusion, 2004, pp. 563–568.

[99] B. Baruque, S. Porras, E. Corchado, Hybrid classification ensemble using topology-preserving clustering, New Generation Computing 29 (2011) 329–344.

[100] L. Kuncheva, Clustering-and-selection model for classifier combination, in: Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000, vol. 1, 2000, pp. 185–188.

[101] K. Jackowski, M. Wozniak, Algorithm of designing compound recognition system on the basis of combining classifiers with simultaneous splitting feature space into competence areas, Pattern Analysis and Applications 12 (4) (2009) 415–425.

[102] M. Wozniak, B. Krawczyk, Combined classifier based on feature space partitioning, International Journal of Applied Mathematics and Computer Sciences 22 (4) (2012) 855–866.

[103] H. Lee, C. Chen, J. Chen, Y. Jou, An efficient fuzzy classifier with feature selection based on fuzzy entropy, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 31 (3) (2001) 426–432.

[104] J. Hong, J. Min, U. Cho, S. Cho, Fingerprint classification using one-vs-all support vector machines dynamically ordered with naïve bayes classifiers, Pattern Recognition 41 (2008) 662–671.

[105] A.R. Ko, R. Sabourin, A. Britto, From dynamic classifier selection to dynamic ensemble selection, Pattern Recognition 41 (5) (2008) 1735–1748.

[106] L. Didaci, G. Giacinto, F. Roli, G. Marcialis, A study on the performances of dynamic classifier selection based on local accuracy estimation, Pattern Recognition 38 (11) (2005) 2188–2191.

[107] G. Giacinto, F. Roli, Dynamic classifier selection based on multiple classifier behavior, Pattern Recognition 34 (9) (2001) 1879–1881.

[108] M. de Souto, R. Soares, A. Santana, A. Canuto, Empirical comparison of dynamic classifier selection methods based on diversity and accuracy for building ensembles, in: IJCNN 2008, IEEE International Joint Conference on Neural Networks, 2008, IEEE World Congress on Computational Intelligence, 2008, pp. 1480–1487.

[109] T. Woloszynski, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, Pattern Recognition 44 (1011) (2011) 2656–2668.

[110] T. Woloszynski, M. Kurzynski, P. Podsiadlo, G. Stachowiak, A measure of competence based on random classification for dynamic ensemble selection, Information Fusion 13 (3) (2012) 207–213.

[111] W. Street, Y. Kim, A streaming ensemble algorithm (sea) for large-scale classification, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '01, ACM, New York, NY, USA, 2001, pp. 377–382.

[112] H. Wang, W. Fan, P. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, ACM, New York, NY, USA, 2003, pp. 226–235.

[113] Y. Zhang, X. Jin, An automatic construction and organization strategy for ensemble learning on data streams, SIGMOD Record 35 (3) (2006) 28–33.

[114] J. Kolter, M. Maloof, Dynamic weighted majority: a new ensemble method for tracking concept drift, in: ICDM 2003, Third IEEE International Conference on Data Mining, 2003, 2003, pp. 123–130.

[115] A. Tsymbal, M. Pechenizkiy, P. Cunningham, S. Puuronen, Dynamic integration of classifiers for handling concept drift, Information Fusion 9 (1) (2008) 56–68.

[116] X. Zhu, X. Wu, Y. Yang, Effective classification of noisy data streams with attribute-oriented dynamic classifier selection, Knowledge Information Systems 9 (3) (2006) 339–363.

[117] D. Tax, R. Duin, Using two-class classifiers for multiclass classification, in: Proceedings of the 16th International Conference on Pattern Recognition, 2002, vol. 2, 2002, pp. 124 –127.

[118] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2 (1995) 263–286.

[119] K. Duan, S. Keerthi, W. Chu, S. Shevade, A. Poo, Multi-category classification by soft-max combination of binary classifiers, in: Proceedings of the 4th International Conference on Multiple Classifier Systems, MCS'03, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 125–134.

[120] A. Passerini, M. Pontil, P. Frasconi, New results on error correcting output codes of kernel machines, IEEE Transactions on Neural Networks 15 (1) (2004) 45–54.

[121] T. Wu, C. Lin, R. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 5 (2004) 975–1005.

[122] J. Friedman, Another Approach to Polychotomous Classification, Tech. rep., Department of Statistics, Stanford University, 1996.

[123] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, Pattern Recognition 43 (1) (2010) 128–142.

[124] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, Pattern Recognition 44 (8) (2011) 1761–1776.

[125] D. Tax, R.P.W. Duin, Characterizing one-class datasets, in: Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa, 2005, pp. 21–26.

[126] D. Tax, R. Duin, Combining one-class classifiers, in: Proceedings of the Second International Workshop on Multiple Classifier Systems, MCS '01, Springer-Verlag, London, UK, 2001, pp. 299–308.

[127] T. Wilk, M. Wozniak, Soft computing methods applied to combination of one-class classifiers, Neurocomputing 75 (2012) 185–193.

[128] G. Giacinto, R. Perdisci, M. Del Rio, F. Roli, Intrusion detection in computer networks by a modular ensemble of one-class classifiers, Information Fusion 9 (2008) 69–82.

[129] Y. Hu, Handbook of Neural Network Signal Processing, 1st ed., CRC Press, Inc., Boca Raton, FL, USA, 2000.

[130] K. Woods, W.P. Kegelmeyer Jr., K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (4) (1997) 405–410.

[131] M. Wozniak, M. Zmyslony, Combining classifiers using trained fuser – analytical and experimental results, Neural Network World 13 (7) (2010) 925–934.

[132] S. Raudys, Trainable fusion rules. I. Large sample size case, Neural Networks 19 (10) (2006) 1506–1516.

[133] M. van Erp, L. Vuurpijl, L. Schomaker, An overview and comparison of voting methods for pattern recognition, in: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition, 2002, 2002, pp. 195–200.

[134] M. Wozniak, K. Jackowski, Some remarks on chosen methods of classifier fusion based on weighted voting, in: E. Corchado, X. Wu, E. Oja, A. Herrero, B. Baruque (Eds.), Hybrid Artificial Intelligence Systems, Lecture Notes in Computer Science, vol. 5572, Springer, Berlin/Heidelberg, 2009, pp. 541–548.

[135] S. Raudys, Trainable fusion rules. II. Small sample-size effects, Neural Networks 19 (10) (2006) 1517–1527.

[136] H. Inoue, H. Narihisa, Optimizing a multiple classifier system, in: M. Ishizuka, A. Sattar (Eds.), PRICAI 2002: Trends in Artificial Intelligence, Lecture Notes in Computer Science, vol. 2417, Springer, Berlin/Heidelberg, 2002, pp. 1–16.

[137] L. Alexandre, A. Campilho, M. Kamel, Combining independent and unbiased classifiers using weighted average., in: Proceedings ICPR 2000, 2000, pp. 2495–2498.

[138] B. Biggio, G. Fumera, F. Roli, Bayesian analysis of linear combiners, in: Proceedings of the 7th International Conference on Multiple Classifier Systems, MCS '07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 292–301.

[139] J. Kittler, F. Alkoot, Sum versus vote fusion in multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (1) (2003) 110–115.

[140] N. Rao, A generic sensor fusion problem: classification and function estimation, in: F. Roli, J. Kittler, T. Windeatt (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 3077, Springer, 2004, pp. 16–30.

[141] D. Opitz, J. Shavlik, Generating accurate and diverse members of a neural-network ensemble, in: NIPS, 1995, pp. 535–541.

[142] L. Rokach, O. Maimon, Feature set decomposition for decision trees, Intelligent Data Analysis 9 (2) (2005) 131–158.

[143] G. Fumera, F. Roli, A theoretical and experimental analysis of linear combiners for multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 942–956, http://dx.doi.org/10.1109/TPAMI.2005.109.

[144] M. Wozniak, Experiments on linear combiners, in: E. Pietka, J. Kawa (Eds.), Information Technologies in Biomedicine, Advances in Soft Computing, vol. 47, Springer, Berlin/Heidelberg, 2008, pp. 445–452.

[145] R. Duin, The combining classifier: to train or not to train? in: Proceedings of the 16th International Conference on Pattern Recognition, 2002, vol. 2, 2002, pp. 765–770.

[146] R. Jacobs, M. Jordan, S. Nowlan, G. Hinton, Adaptive mixtures of local experts, Neural Computation 3 (1991) 79–87.

[147] R. Jacobs, Methods for combining experts' probability assessments, Neural Computation 7 (5) (1995) 867–888.

[148] V. Tresp, M. Taniguchi, Combining estimators using non-constant weighting functions, Advances in Neural Information Processing Systems, vol. 7, MIT Press, 1995, pp. 419–426.

[149] P. Cheeseman, M. Self, J. Kelly, J. Stutz, W. Taylor, D. Freeman, AutoClass: a Bayesian classification system, in: Machine Learning: Proceedings of the Fifth International Workshop, Morgan Kaufman, 1988.

[150] S. Shlien, Multiple binary decision tree classifiers, Pattern Recognition 23 (7) (1990) 757–763.

[151] M. Wozniak, Experiments with trained and untrained fusers, in: E. Corchado, J. Corchado, A. Abraham (Eds.), Innovations in Hybrid Intelligent Systems, Advances in Soft Computing, vol. 44, Springer, Berlin/Heidelberg, 2007, pp. 144–150.

[152] M. Wozniak, Evolutionary approach to produce classifier ensemble based on weighted voting, in: NaBIC 2009, World Congress on Nature & Biologically Inspired Computing, 2009, IEEE, 2009, pp. 648–653.

[153] L. Lin, X. Wang, B. Liu, Combining multiple classifiers based on statistical method for handwritten chinese character recognition, in: Proceedings of the 2002 International Conference on Machine Learning and Cybernetics, 2002, vol. 1, 2002, pp. 252–255.

[154] Z. Zheng, B. Padmanabhan, Constructing ensembles from data envelopment analysis, INFORMS Journal on Computing 19 (4) (2007) 486–496.

[155] D. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.

[156] Y. Huang, C. Suen, A method of combining multiple experts for the recognition of unconstrained handwritten numerals, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (1) (1995) 90–94.

[157] M.M. Gaber, A. Zaslavsky, S. Krishnaswamy, Mining data streams: a review, SIGMOD Record 34 (2) (2005) 18–26.

[158] A. Patcha, J.-M. Park, An overview of anomaly detection techniques: existing solutions and latest technological trends, Computer Network 51 (12) (2007) 3448–3470.

[159] M.M. Black, R.J. Hickey, Classification of customer call data in the presence of concept drift and noise, in: Proceedings of the First International Conference on Computing in an Imperfect World, Soft-Ware 2002, Springer-Verlag, London, UK, 2002, pp. 74–87.

[160] H. Wang, W. Fan, P.S. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, ACM, New York, NY, USA, 2003, pp. 226–235.

[161] M.M. Gaber, P.S. Yu, Classification of changes in evolving data streams using online clustering result deviation, in: Proc. Of International Workshop on Knowledge Discovery in Data Streams, 2006.

[162] M. Markou, S. Singh, Novelty detection: a review – Part 1: Statistical approaches, Signal Process 83 (12) (2003) 2481–2497.

[163] M. Salganicoff, Density-adaptive learning and forgetting, in: Machine Learning: Proceedings of the Tenth Annual Conference, Morgan Kaufmann, San Francisco, CA, 1993.

[164] R. Klinkenberg, T. Joachims, Detecting concept drift with support vector machines, in: Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 487–494.

[165] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldá, R. Morales-Bueno, Early drift detection method, in: Fourth International Workshop on Knowledge Discovery from Data Streams, 2006.

[166] I. Zliobaite, Change with delayed labeling: when is it detectable?, in: Proceedings of the 2010 IEEE International Conference on Data Mining Workshops, ICD-MW '10, IEEE Computer Society, Washington, DC, USA, 2010, pp 843–850.

[167] G. Giacinto, F. Roli, L. Bruzzone, Combination of neural and statistical algorithms for supervised classification of remote-sensing images, Pattern Recognition Letters 21 (5) (2000) 385–397.

[168] V. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, J. Rigol-Sanchez, An assessment of the effectiveness of a random forest classifier for land-cover classification, ISPRS Journal of Photogrammetry and Remote Sensing 67 (0) (2012) 93–104.

[169] P. Gislason, J. Benediktsson, J. Sveinsson, Random forests for land cover classification, Pattern Recognition Letters 27 (4) (2006) 294–300.

[170] J.-W. Chan, D. Paelinckx, Evaluation of random forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery, Remote Sensing of Environment 112 (6) (2008) 2999–3011.

[171] J. Peters, N. Verhoest, R. Samson, M. Meirvenne, L. Cockx, B. Baets, Uncertainty propagation in vegetation distribution models based on ensemble classifiers, Ecological Modelling 220 (6) (2009) 791–804.

[172] M. Han, X. Zhu, W. Yao, Remote sensing image classification based on neural network ensemble algorithm, Neurocomputing 78 (1) (2012) 133–138.

[173] B. Waske, M. Braun, Classifier ensembles for land cover mapping using multitemporal SAR imagery, ISPRS Journal of Photogrammetry and Remote Sensing 64 (5) (2009) 450–457 (theme Issue: Mapping with SAR: Techniques and Applications).

[174] B. Waske, S. van der Linden, C. Oldenburg, B. Jakimow, A. Rabe, P. Hostert, imageRF – a user-oriented implementation for remote sensing image analysis with random forests, Environmental Modelling & Software 35 (0) (2012) 192–193.

[175] U. Maulik, D. Chakraborty, A self-trained ensemble with semisupervised SVM: an application to pixel classification of remote sensing imagery, Pattern Recognition 44 (3) (2011) 615–623.

[176] A. Henriques, A. Doria-Neto, R. Amaral, Classification of multispectral images in coral environments using a hybrid of classifier ensembles, Neurocomputing 73 (7–9) (2010) 1256–1264.

[177] Y. Maghsoudi, M. Collins, D. Leckie, Polarimetric classification of boreal forest using nonparametric feature selection and multiple classifiers, International Journal of Applied Earth Observation and Geoinformation 19 (0) (2012) 139–150.

[178] L. Bruzzone, R. Cossu, G. Vernazza, Combining parametric and non-parametric algorithms for a partially unsupervised classification of multitemporal remote-sensing images, Information Fusion 3 (4) (2002) 289–297.

[179] L. Bruzzone, R. Cossu, G. Vernazza, Detection of land-cover transitions by combining multidate classifiers, Pattern Recognition Letters 25 (13) (2004) 1491–1500.

[180] P. Du, S. Liu, J. Xia, Y. Zhao, Information fusion techniques for change detection from multi-temporal remote sensing images, Information Fusion 14 (1) (2013) 19–27.

[181] P. Arun-Raj-Kumar, S. Selvakumar, Distributed denial of service attack detection using an ensemble of neural classifier, Computer Communications 34 (11) (2011) 1328–1341.

[182] P. Kumar, S. Selvakumar, Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems, Computer Communications (0) (2012).

[183] A. Shabtai, R. Moskovitch, Y. Elovici, C. Glezer, Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey, Information Security Technical Report 14 (1) (2009) 16–29.

[184] M. Locasto, K. Wang, A. Keromytis, S. Stolfo, Flips: hybrid adaptive intrusion prevention, in: Proceedings of the 8th International Conference on Recent Advances in Intrusion Detection, RAID'05, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 82–101.

[185] K. Wang, G. Cretu, S. Stolfo, Anomalous payload-based worm detection and signature generation, in: Proceedings of the 8th International Conference on Recent Advances in Intrusion Detection, RAID'05, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 227–246.

[186] S. Peddabachigari, A. Abraham, C. Grosan, J. Thomas, Modeling intrusion detection system using hybrid intelligent systems, Journal of Network and Computer Applications 30 (1) (2007) 114–132.

[187] D.-I. Curiac, C. Volosencu, Ensemble based sensing anomaly detection in wireless sensor networks, Expert Systems with Applications 39 (10) (2012) 9087–9096.

[188] R.J. Bolton, D.J. Hand, Statistical fraud detection: a review, Statistical Science 17 (3) (2002) 235–255.

[189] F. Louzada, A. Ara, Bagging k-dependence probabilistic networks: an alternative powerful fraud detection tool, Expert Systems with Applications 39 (14) (2012) 11583–11592.

[190] S. Bhattacharyya, S. Jha, K. Tharakunnel, J. Westland, Data mining for credit card fraud: a comparative study, Decision Support Systems 50 (3) (2011) 602–613.

[191] L. Yu, W. Yue, S. Wang, K. Lai, Support vector machine based multiagent ensemble learning for credit risk evaluation, Expert Systems with Applications 37 (2) (2010) 1351–1360.

[192] Y. Kim, S. Sohn, Stock fraud detection using peer group analysis, Expert Systems with Applications 39 (10) (2012) 8986–8992.

[193] B. Twala, Multiple classifier application to credit risk assessment, Expert Systems with Applications 37 (4) (2010) 3326–3336.

[194] S. Finlay, Multiple classifier architectures and their application to credit risk assessment, European Journal of Operational Research 210 (2) (2011) 368–378.

[195] G. Wang, J. Ma, A hybrid ensemble approach for enterprise credit risk assessment based on support vector machine, Expert Systems with Applications 39 (5) (2012) 5325–5331.

[196] M. Kim, D. Kang, Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction, Expert Systems with Applications 39 (10) (2012) 9308–9314.

[197] P. Ravisankar, V. Ravi, I. Bose, Failure prediction of dotcom companies using neural network–genetic programming hybrids, Information Sciences 180 (8) (2010) 1257–1267.

[198] P. Ravisankar, V. Ravi, G. Rao, I. Bose, Detection of financial statement fraud and feature selection using data mining techniques, Decision Support Systems 50 (2) (2011) 491–500.

[199] C. Tsai, Combining cluster analysis with classifier ensembles to predict financial distress, Information Fusion (0) (2011).

[200] Y. Peng, G. Wang, G. Kou, Y. Shi, An empirical study of classification algorithm evaluation for financial risk prediction, Applied Soft Computing 11 (2) (2011) 2906–2915.

[201] V. Ravi, H. Kurniawan, P. Nwee-Kok-Thai, P. Ravi-Kumar, Soft computing system for bank performance prediction, Applied Soft Computing 8 (1) (2008) 305–315.

[202] H. Zhao, A. Sinha, W. Ge, Effects of feature construction on classification performance: an empirical study in bank failure prediction, Expert Systems with Applications 36 (2, Part 2) (2009) 2633–2644.

[203] K. Aral, H. Guvenir, I. Sabuncuoglu, A. Akar, A prescription fraud detection model, Computer Methods and Programs in Biomedicine 106 (1) (2012) 37–46.

[204] I. Christou, M. Bakopoulos, T. Dimitriou, E. Amolochitis, S. Tsekeridou, C. Dimitriadis, Detecting fraud in online games of chance and lotteries, Expert Systems with Applications 38 (10) (2011) 13158–13169.

[205] H. Farvaresh, M. Sepehri, A data mining framework for detecting subscription fraud in telecommunication, Engineering Applications of Artificial Intelligence 24 (1) (2011) 182–194.

[206] L. Subelj, S. Furlan, M. Bajec, An expert system for detecting automobile insurance fraud using social network analysis, Expert Systems with Applications 38 (1) (2011) 1039–1052.

[207] A.X. Garg, N.K.J. Adhikari, H. McDonald, M.P. Rosas-Arellano, P.J. Devereaux, J. Beyene, J. Sam, R.B. Haynes, Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review, Journal of the American Medical Association 293 (10) (2005) 1223–1238.

[208] J. Eom, S. Kim, B. Zhang, AptaCDSS-E: a classifier ensemble-based clinical decision support system for cardiovascular disease level prediction, Expert Systems with Applications 34 (4) (2008) 2465–2479.

[209] R. Das, I. Turkoglu, A. Sengur, Effective diagnosis of heart disease through neural networks ensembles, Expert Systems with Applications 36 (4) (2009) 7675–7680.

[210] R. Das, I. Turkoglu, A. Sengur, Diagnosis of valvular heart disease through neural networks ensembles, Computer Methods and Programs in Biomedicine 93 (2) (2009) 185–191.

[211] W. Baxt, Improving the accuracy of an artificial neural network using multiple differently trained networks, Neural Computation 4 (5) (1992) 772–780.

[212] X. Zhang, J. Mesirov, D. Waltz, Hybrid system for protein secondary structure prediction, Journal of Molecular Biology 225 (4) (1992) 1049–1063.

[213] L. Nanni, Ensemble of classifiers for protein fold recognition, Neurocomputing 69 (7) (2006) 850–853.

[214] T. Yang, V. Kecman, L. Cao, C. Zhang, J.Z. Huang, Margin-based ensemble classifier for protein fold recognition, Expert Systems with Applications 38 (10) (2011) 12348–12355.

[215] A. Savio, M. Garcia-Sebastian, D. Chyzyk, C. Hernandez, M. Graña, A. Sistiaga, A.L. de Munain, J. Villanua, Neurocognitive disorder detection based on feature vectors extracted from VBM analysis of structural MRI, Computers in Biology and Medicine 41 (8) (2011) 600–610.

[216] D. Chyzhyk, M. Graña, A. Savio, J. Maiora, Hybrid dendritic computing with kernel-LICA applied to alzheimer's disease detection in MRI, Neurocomputing 75 (1) (2012) 72–77.

[217] L. Kuncheva, J. Rodriguez, Classifier ensembles for fMRI data analysis: an experiment, Magnetic Resonance Imaging 28 (4) (2010) 583–593.

[218] C. Plumpton, L. Kuncheva, N. Oosterhof, S. Johnston, Naive random subspace ensemble with linear classifiers for real-time classification of fMRI data, Pattern Recognition 45 (6) (2012) 2101–2108.

[219] C. Cabral, M. Silveira, P. Figueiredo, Decoding visual brain states from fMRI using an ensemble of classifiers, Pattern Recognition 45 (6) (2012) 2064–2074.

[220] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, ACM Transactions Information Systems 23 (1) (2005) 103–145.

[221] J. Konstan, J. Riedl, How online merchants predict your preferences and prod you to purchase, IEEE Spectrum 49 (10) (2012) 48–56.

[222] R. Burke, Hybrid recommender systems: survey and experiments, User Modeling and User-Adapted Interaction 12 (4) (2002) 331–370.

[223] M. Balabanović, Y. Shoham, Fab: content-based, collaborative recommendation, Communications of the ACM 40 (3) (1997) 66–72.

[224] M.J. Pazzani, A framework for collaborative, content-based and demographic filtering, Artificial Intelligence Review 13 (5–6) (1999) 393–408.

[225] M. Jahrer, A. Töscher, R. Legenstein, Combining predictions for accurate recommender systems, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, ACM, New York, NY, USA, 2010, pp. 693–702.

[226] C. Porcel, A. Tejeda-Lorente, M. Martínez, E. Herrera-Viedma, A hybrid recommender system for the selective dissemination of research resources in a technology transfer office, Information Sciences 184 (1) (2012) 1–19.

[227] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, M. Sartin, Combining content-based and collaborative filters in an online newspaper, in: Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation, ACM, 1999.

[228] D. Billsus, M. Pazzani, User modeling for adaptive news access, User Modeling and User-Adapted Interaction 10 (2–3) (2000) 147–180.

[229] T. Tran, R. Cohen, Hybrid recommender systems for electronic commerce, in: Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04, AAAI Press, Menlo Park, CA, 2000, pp. 78–83.

[230] M. Kunaver, T. Pozrl, M. Pogacnik, J. Tasic, Optimisation of combined collaborative recommender systems, AEU – International Journal of Electronics and Communications 61 (7) (2007) 433–443.

[231] M. Goksedef, S. Gundoz-Oguducu, Combination of web page recommender systems, Expert Systems with Applications 37 (4) (2010) 2911–2922.