Engineering Applications of Artificial Intelligence 23 (2010) 1158–1172

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Collaborative multi-agent rock facies classification from wireline well log data

Christopher M. Gifford [a,*], Arvin Agah [b]

[a] National Security Technology Department, The Johns Hopkins University Applied Physics Laboratory, Laurel, MD 20723, USA
[b] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA

ARTICLE INFO

Article history:
Received 8 June 2009
Received in revised form
2 February 2010
Accepted 10 February 2010
Available online 7 March 2010

Keywords:
Rock classification
Well logs
Collaborative learning
Multi-agent systems
Applied artificial intelligence

ABSTRACT

Gas and oil reservoirs have been the focus of modeling efforts for decades as an attempt to locate zones with high volumes. Certain subsurface layers and layer sequences, such as those containing shale, are known to be impermeable to gas and/or liquid. Oil and natural gas then become trapped by these layers, making it possible to drill wells to reach the supply, and extract for use. The drilling of these wells, however, is costly. In this paper, we utilize multi-agent machine learning and classifier combination to learn rock facies sequences from wireline well log data. The paper focuses on how to construct a successful set of classifiers, which periodically collaborate, to increase the classification accuracy. Utilizing multiple, heterogeneous collaborative learning agents is shown to be successful for this classification problem. Utilizing the Multi-Agent Collaborative Learning Architecture, 84.5% absolute accuracy was obtained, an improvement of about 6.5% over the best results achieved by the Kansas Geological Survey with the same data set. A number of heuristics are presented for constructing teams of multiple collaborative classifiers for predicting rock facies.

1. Introduction

Gas and oil reservoirs have been the focus of modeling efforts for decades as an attempt to locate zones with high volumes. As the Earth changes over time, different rock type layers (or lithofacies) are created. These subsurface layers form time-based cyclic sequences, where layers that are superimposed on top of others are geologically younger. Certain layers and layer sequences, such as shale, are known to be impermeable to gas and/or liquid. Oil and natural gas then become trapped by these layers, making it possible to drill wells (also called cores) to reach the supply and extract for use. The ability to predict where these layers and known sequences are, based on properties of core samples, provides the potential for more efficient "payzone" drilling. This process of well log analysis is an important step in geophysical exploration, as it can aid in drilling path optimization. Well log analysis requires geologists and geophysicists, which have the necessary knowledge to relate rock types and layering to specific events and resources. The focus of this paper is on how machine learning can be used to automate such tasks.

To date, rock facie (subsurface layer) classification has largely used wireline log measurements together with multivariate statistical methods, fuzzy mathematical approaches, and artificial neural networks. Wireline well logs are physical and chemical rock measurements recorded by lowering logging tools with specialized sensors into wells after they have been drilled. Properties such as permeability, porosity, and liquid content can be extracted using specific measurement mediums. Electrical measurements offer information on saturation (e.g., oil or water); while acoustic measurements can provide information about grain size. Thus, certain rocks and layers will exhibit characteristic measurement signatures based on what they contain. Experts analyze these data to determine the rock type and sequence. This not only represents a tedious and time-consuming task, but the process of drilling wells and analyzing them is costly. These costs greatly increase as the number of wells and log data attributes increase.

Prominent applications of machine learning to geology include material classification while drilling for coal mine roof stability (LaBelle et al., 2000), mapping rock mechanical properties using seismic data (Liu and Sacchi, 2003), identification of water-flooded oil layers (Shang et al., 2008), tephra layer correlation for predicting volcanic eruptions (Rogova et al., 2007), and soil classification (Bhattacharya and Solomatine, 2006). These works demonstrate the effectiveness of the use of artificial intelligence in the geoscience fields.

In this study, we utilize wireline well log data collected from the Council Grove Group (Panama Field) in the Midwest as part of a multi-agent model-creation effort for a rock facies classification

* Corresponding author. Tel.: +1 443 778 4958; fax: +1 443 778 9188.
E-mail address: Chris.Gifford@jhuapl.edu (C.M. Gifford).

0952-1976/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2010.02.004

problem. Specifically, a detailed study of team learning, collaboration, and decision combination has been conducted. Experimental results are presented, including successful team compositions, individual learning algorithm contribution, team size, collaboration frequency, and team diversity. Finally, the primary team learning and collaboration aspects discovered as part of this application are summarized.

2. Background and related work

There exists a set of traditional log curves that are used in practice to differentiate between rock facies, each measurement of which offers an important piece to be able to classify rock content. The following are a few of these curves (Shi et al., 2005):

- *Resistivity*: Electrical measurement that detects vertical changes in conductivity; Resistivity is the reciprocal of conductivity. Different liquids respond differently to this measurement (e.g., salt water conducts electricity, whereas oil does not).
- *Gamma Ray*: Device measuring natural radiation. Formations that contain higher amounts of organic material (such as shale) emit more radiation, resulting in higher gamma ray counts.
- *Neutron*: Device measuring hydrogen atom presence. Given the assumption that pores are filled with either water or hydrocarbons, this provides information about pore space.
- *Acoustic*: Sonic tool measuring porosity and potential gas content of formations.

Several research efforts have applied machine learning methods to the problems of reservoir and rock formation identification. Machine learning fits well into this problem as it has the potential to make the process more efficient. Specifically, it is possible to formalize the expert knowledge through knowledge engineering, but can be very challenging to capture the expert knowledge on a deep and sufficient level. Although expert systems and single-classifier systems (such as neural networks) have been used with success, more sophisticated methods are needed to increase the accuracy of field systems.

Manual study of digital sensor measurements from many core sites and log measurements is impractical and expensive. Using existing wireline well log data, algorithms can be used to automatically create (or learn) a model that relates the log measurements to rock facies. Once these models are created, they can be validated on other known core sequences to evaluate their accuracy. Furthermore, information from these learned models can be used to intelligently interpolate subsurface behavior between well locations, allowing for general 3D subsurface models of a region. However, due to spatial distribution and heterogeneity of subsurface properties, this yields additional complexities.

Researchers have used a combination of data mining and an expert system to perform intelligent well log analysis with data from the Xinjiang Province in China (Shi et al., 2005). That study included boreholes containing subsurface layers of shale, water, oil, and a mixture of water and oil. The RIPPER inductive learning algorithm was used for its efficiency and human-readable production rules. Decision trees have also been used for facies identification. In Al-Faraj (1998), the C4.5 decision tree learning algorithm was used to classify a total of five facies with a variety of natural and synthetic variables. The primary attributes used in the study were gamma ray, bulk density, neutron porosity, and depth information. Several other variables were computed from these attributes, and incrementally provided to the decision tree to increase overall testing accuracy.

In naturally occurring strata, multiple geologic facies exist and are non-linearly coupled. Over time, processes deform these layers in complex ways. Boundaries between them are difficult to properly model, and represent areas of classification difficulty. This translates to a multi-class classification problem with non-linear decision boundaries. Support vector machines (SVM), due to their recent surge in popularity and rigorous mathematical basis, have also been applied to this problem of facies delineation. Synthetic hydraulic conductivity data were utilized in Tartakovsky and Wohlberg (2004) to study SVM and statistical method performance for the idealized problem of identifying the stratified boundary between two layers (a two-class, linearly separable problem). The SVM was found to outperform the geostatistical approach for boundary estimation. This work was extended (Wohlberg et al., 2006) for the non-linear boundary case using similar synthetic data. It was found that error decreased logarithmically with increasing sampling density.

Neural networks have seen more widespread use in geophysical applications (e.g., van der Baan and Jutten, 2000 includes a review of neural network use in these applications, and (Wong et al., 1995) presents a porosity prediction example). The work presented in Saggaf and Nebrija (2000) used unsupervised analysis to segregate wells into classes based on well logs, and then applied a supervised learner to create a model for estimating lithology in horizontal wells in Saudi Arabia. Their method also allowed for confidence measures given facies data uncertainty. Unsupervised analysis took place using a single-layer neural network, where the neuron most resembling the input is rewarded by progressively moving closer to the center of the corresponding input data cluster (termed "competitive learning"). Intuitively, confidence was found to significantly decrease at facie transition zones.

Some researchers have concluded that more advanced machine learning methods are required to increase accuracy beyond that of single classifiers. Multi-classifier systems, such as neural network ensembles and committee machines, have been evaluated for lithology recognition and reservoir characterization. Further, the use of multiple classifiers brings about methods for combining their decisions. Experimental results presented in this paper support this research thrust.

Ensembles of neural networks, sometimes called committee machines, were used in Santos et al. (2003) with data from a Brazilian offshore well. Different committees of neural networks with identical structures (single hidden layer) were developed via approaches of pattern replication (one neural network per class), bootstrapping (series of patterns created using random sampling with replacement), and iterative pattern selection (sample probabilities for misclassified patterns iteratively increased). Classifier combination methods of averaging, majority vote, rank counting, and fuzzy integrals were assessed and compared against the performance of a single neural network. Gamma ray, acoustic, bulk density, resistivity, caliper, and depth information were used as attributes for three rock classes. For each committee formation method, several experiments were performed with up to 100 networks per committee. The fuzzy combination method was found to provide the highest accuracy. It was also found that having more neural networks generally offered increased classification accuracy. The best committee machine outperformed the reference neural network by over 7%. This study was an extension of dos Santos et al. (2002), which removed the caliper attribute and increased the classes from three to eight. The authors also separated the wells into two categories: entire well and reservoir (depth greater than 2500 meters). Results were similar, showing that pattern replication worked well for the case of non-stratified data sets.

The work in Bhatt et al. (2001) and Bhatt (2002) also utilizes several neural networks for wireline and measurement-while-drilling analysis to predict porosity, permeability, and fluid saturation. These properties are key for determining flow patterns. The measurement-while-drilling approach offers a more real-time look at reservoir properties, allowing the drillers to steer their efforts based on subsurface properties (termed "geosteering"). Each property-based neural network in this study is identical with individual randomized starting weights; each being trained on the same pattern or a subset, but focusing on a single property (e.g., permeability, porosity). The outputs of each network are combined using optimal linear combination, which aims at reducing variance. The authors noted that such an approach is useful as each neural network typically found differing local minima. For lithofacies prediction, each neural network was trained to be an expert on a single facie, rejecting others. A gating network was also utilized to introduce prior knowledge of geology (e.g., local stratigraphy) into the network models. Facies prediction for the Ness Formation achieved an average hit-rate of above 90%.

From these works, it has been shown that using machine learning to create geophysical models for well log analysis is not only feasible, but can provide high levels of accuracy while offering a significant increase in efficiency. In the following sections, we discuss our approach, using multiple heterogeneous collaborative learning agents to further increase accuracy for the problem of well log analysis and facie classification. Experiments focused on agents' team size, composition, collaboration frequency, and data division results are presented. This approach could also be applied to measurement-while-drilling and finding stores of other gasses of interest such as CO_2, but is not explicitly discussed here.

3. Combining decisions from multiple classifiers

Classifier combination is an area in machine learning that has offered advances in classification accuracy for complex data sets. It has been termed differently in the literature, namely, classifier fusion, mixture of experts, committees, ensembles, teams, pools, collective recognition, composite systems, etc. When predictions from multiple classifiers are combined, they are said to form an ensemble that is then used to classify new instances. Several methods have been developed to combine classifiers, the most popular of which are voting, boosting, bagging, and stacking.

The majority of efforts in combining classifiers have incorporated homogeneous learning algorithms. Fewer works have focused on combining heterogeneous learners, which is the focus of our study. The term "heterogeneous" is used in different contexts in the literature. Popular classifier combination approaches such as boosting and bagging involve homogeneous learning algorithms and manipulation of the training data set. The produced models are therefore homogeneous in representation; although the learners may output slightly different predictions. Use of different learning algorithms, and therefore heterogeneous model representations, is considered purely heterogeneous in the contexts of both algorithm and model. Integrating multiple learning paradigms within the same learner is viewed as a hybrid scheme (Kim et al., 2000).

When developing a multi-classifier system, its members can be a mixture of weak (high error) and strong (low error) classifiers. Weak classifiers are typically simple to create, at the expense of their accuracy on complex data sets. Strong classifiers are typically time-consuming and expensive to create, as their parameters are fine-tuned for maximum performance. Furthermore, some classifiers perform better than others on the same data set due to their algorithmic nature. Combining weak/strong or homogeneous/heterogeneous classifiers offers the benefit of encompassing different levels of expertise and knowledge bases. Exploiting and studying these properties as advantages for classification is a driving force for multi-agent machine learning.

An important aspect related to difference in learners is their level of error correlation relative to one another, and as a team. The more correlated (less disjoint) individual learners are, the less complementary they may be. If uncorrelated, they likely will misclassify different instances. Combining them better enables the system to correctly classify more instances. A significant improvement over a single classifier can only occur if the individual classifier theories are substantially different. It is desired to obtain a balance between high performance and complementarity in a team which combines decisions. If one learner does not predict correctly, the other learners should be able to do so. Diverse models are therefore more likely to produce error in different ways. Methods to accomplish this typically involve introducing diversity in terms of learning paradigms, feature subsets, or training sets (Tumer and Ghosh, 1996).

One of the primary questions in this area of study is whether combining classifiers is better than selecting the best classifier. Several works support that classifier combination provides an improvement in most cases, assuming that the classifiers exhibit reasonable individual accuracy. One such study utilized stacking with model trees to combine multiple heterogeneous learners (Dzeroski and Zenko, 2004). Each learner utilized the full data set to produce a base-level model, the output of which is then combined with other base-level models using a meta-level classifier. Their results also indicated that the number of base-level classifiers did not significantly affect the results. Similarly, Klein et al. (2002) found that when using voting and entropy methods as the heterogeneous classifier combination mechanism for word-sense disambiguation, increasing the number of less accurate classifiers adversely affected those with higher accuracy. Use of more classifiers, even if heterogeneous, does not always translate to better results. Depending on the combination method, the relative impact of adding classifiers can diminish as team size increases. Our work attempts to address this by implementing a framework where team configurations can be analyzed to determine what factors make a team of classifiers successful.

Although selection of the best individual classifier is easier and occasionally effective, combination techniques scale better to larger and more complex learning problems. Even combining all classifiers in an ensemble can be improved upon by selecting for combination only those that perform significantly better than others, termed Selective Fusion (Tsoumakas et al., 2005). Using this technique, together with simple voting methods, enables the fine-tuning of diverse ensembles for specific data sets. It also offers performance comparable to other heterogeneous classifier combination methods such as stacking, without the additional computation and meta-learning costs. Researchers have studied the effectiveness of switching between selection (occurring in regions of the feature space where single learners are dominant) and fusion (occurring in every region not dominated by a single learner) (Kuncheva, 2002). The results offer motivation to investigate other methods that may offer comparable performance, such as collaborative learning discussed in this paper.

Techniques such as boosting and stacking can also be chained together to achieve high accuracy. In Ebrahimpour (2007), a base-level classifier is trained, and examples that it incorrectly classifies are serially added to the next classifier's training set. Each base-level classifier likely then has a different error rate, but when combined still represent homogeneous models even though they are largely viewed as diverse. This process continues for all base-level classifiers, their outputs are averaged, and finally

combined using a cross-validated tree classifier. Results showed that this ensemble method outperformed conventional boosting and stacking. A homogeneous ensemble of decision stump learners combining the popular methods of boosting, bagging, and dagging via the sum voting methodology has also proven successful with some data sets (Kotsianti and Kanellopoulos, 2007). This is an example of how heterogeneity can lead to more robust and high-performance classifiers.

Other efforts have studied the use of training multiple classifiers on different feature subsets prior to their combination (Chen et al., 1997). Focusing on differing and potentially overlapping feature subsets creates additional diversity, which could lead to an improved combined model. Some researchers also found that performance degradation occurs as the percentage of training batches (sets of training examples) overlap (Ting and Low, 1996). Multi-agent learning systems offer a medium for additional study on how knowledge from learners with different, potentially overlapping feature and data subsets can be combined. Interaction between the learners during the learning process may prove to increase learning efficiency, robustness, and accuracy.

4. Kansas Geological Survey well log data set

This section describes the data set statistics, its collection details, and recent machine learning and model-creation efforts by the Kansas Geological Survey. We have utilized this data set for the multi-agent collaborative learning study.

4.1. Data set properties

The Kansas Geological Survey (KGS) (KGS, 2009) has been studying the Hugoton and Panoma Fields in Southwest Kansas and Oklahoma to create subsurface models (Dubois et al., 2007; Bohling and Dubois, 2003; Dubois et al., 2003, 2004). These fields comprise the largest gas-producing area in North America as of 2007, with 963 billion mm^3 of gas from over 12,000 wells (Dubois et al., 2007). With such a large field and possibility for many core sites, manual study of wireline logs becomes very time consuming and impractical. Automatic methods, such as the use of machine learning algorithms or statistical approaches, are therefore desirable.

KGS has focused primarily on comparing four individual model-creation methods: Bayesian, K-Nearest Neighbors (KNN), Fuzzy Logic, and Back-propagation Neural Networks. KGS created a data set for the Council Grove Group in the Panama Field, which was divided into eight rock facies, each representing a class (type) of rock:

1. Continental origin, coarse siltstone.
2. Continental origin, shaley fine siltstone.
3. Marine origin, marine siltstone.
4. Marine origin, carbonate mudstone.
5. Marine origin, wackestone.
6. Marine origin, fine-crystalline dolomite.
7. Marine origin, packstone.
8. Marine origin, grainstone.

The order of these facies generally translates to increased permeability for a given porosity, and adjacent classes mostly exhibit similar properties. Facies 6–8 represent the primary gas payzone facies.

Seven numeric measurement features/attributes are used in the data set. These attributes are utilized to model the data, using rock type as the target for training.

1. RPOS: Relative position of a sample from marine/continental half-cycle boundary.
2. NM-M: Non-marine/marine indicator (sample origin).
3. GR: Natural gamma radiation.
4. RTA: Apparent true resistivity.
5. N-D: Average neutron and density porosity.
6. PHIND: Neutron and density porosity difference.
7. PE: Photoelectric effect.

The addition of the non-marine/marine indicator and relative position of a sample from the boundary of its indicator were included to incorporate geologic knowledge. These features were recorded at half-foot intervals within the wells. Some wells do not contain the PE feature, so the data set was separated into one set with and one set without this feature for independent study. Our research using this data set only focuses on the set containing PE measurements (a total of five geographically distributed wells).

KGS performed a detailed study to compare the four classification methods on both data sets (Dubois et al., 2007). As facies close to one another are generally similar, the authors computed absolute accuracy, within-one accuracy, and accuracy on a specific set of payzone facies known for reservoir storage and flow. Within-one accuracy was important in their study, as they wanted to maximize accuracy on a specific set of facies (classes), where being within one facie of being absolutely correct was acceptable. As our work focuses on the PE data set and absolute accuracy for all facies, Table 1 summarizes absolute accuracy results they achieved on the testing portion of this data set (Dubois et al., 2007). The data was randomly split into two-thirds training and one-third testing portions to mimic the KGS experimental setup.

The authors noted that the fuzzy logic classifier was not effective in classifying some facies, two of which were payzone facies. The KNN classifier showed a lower absolute correctness for the payzone facies, but improved the overall absolute accuracy. Finally, the neural network performed best at predicting facies, yet still had some trouble with the payzone facies. The final neural network that was used in their work was the culmination of many cross-validated tests to determine the best network architecture for this data set. Therefore, the 78% achieved by the neural network represents the current best performance that has been achieved using any learning approach with this data set.

4.2. Data set statistics

KGS results implied that different learning algorithms had difficulty with different areas and classes of the data set. This

Table 1
Testing results obtained by KGS using four independent learning methods (Dubois et al., 2007).

Learning method	Approach	Accuracy (%)
Bayesian	Traditional quadratic Bayesian method	62
Fuzzy logic	Used a fuzzifier to turn the 7-element input into a 56-element feature vector by computing a degree of membership for each element to all eight classes.	62
KNN CDF	K=20 nearest neighbors	67
Neural Network	Network comprised of 7 input nodes (attributes as inputs), a single layer with 50 hidden nodes, and 8 output nodes (probability for each class/facie). Damping parameter of 0.1 and iteration 100 times.	78

prompted us to look to classifier combination and collaborative learning to improve overall absolute facies classification accuracy. This is a difficult classification problem, as rock properties controlling gas production overlap and wireline logging tools tend to average and blur the facies boundaries (Dubois et al., 2004). Although the KGS neural network was able to achieve a high payzone accuracy, our work is more concerned with improving the overall absolute accuracy for all facies.

The same KGS data set was obtained to perform our collaborative multi-agent learning study. Statistics of the entire (PE and NoPE combined) data set are presented in Dubois et al. (2007). Our study is focused on use of all seven attributes (the portion of the data set containing the PE measure). All instances with missing PE values were removed, resulting in a total of 2274 training and testing instances. These instances were randomly split and stratified into two-thirds training and one-third testing portions, the same training and testing setup utilized by KGS to create their models and compare their classifiers. Each attribute is normalized prior to learning, with the testing data being normalized using the training data's mean and standard deviation.

Tables 2 and 3 present the statistics of this data set, including the seven attributes and distributions for the eight classes (facies) which form the classification problem. These facies distinguish between rock type and texture, and were selected to maximize distinguishable wireline log curves, minimize the number required to represent the reservoir accurately, and distinguish between porosity/permeability/pressure relationships (Dubois et al., 2007). The boundaries between facies represent the most difficult classification area, as the boundaries can be blurred by the logging sensors.

The structure of the features reveals the difficulty of classifying facies based upon it. Considering two features with the lowest variance (GR and RTA), one would expect a significant amount of overlap. As shown in Fig. 1, the data exhibits substantial overlap. Each individual attribute in the data set has been normalized from 0 to 1. Most facies appear to be somewhat evenly distributed from 0 to 0.4 for gamma radiation (GR), and from 0 to 0.4 for apparent true resistivity (RTA). The most pronounced differences between two facies are between facies 3 and 8, which also substantially overlap.

As for the two features with the highest variance (NM-M and RPOS), Fig. 2 shows even these exhibit substantial overlap. In theory, this separates the facies into two clearly differentiable groups. However, even this distinction is not absolute. For example, while most of facies 8 is denoted marine, some of its samples are also non-marine. Additionally, the next highest variance feature, relative position (RPOS), does not allow for much differentiation. It is difficult to distinguish on this dimension where one facie begins and another ends. These aspects illustrate the difficulty of this classification problem.

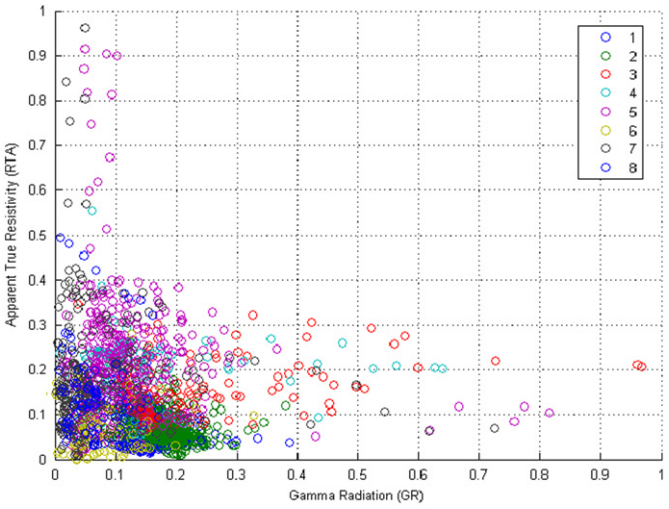

Fig. 1. Overlap for the two lowest variance features by facie/class: gamma radiation and apparent true resistivity.

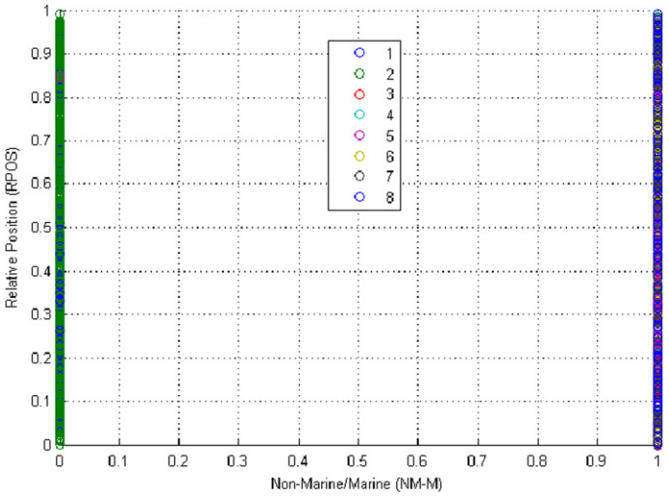

Fig. 2. Overlap for the two highest variance features by facie/class: non-marine/marine and relative position. The non-marine/marine indicator is binary (0 or 1).

Table 2
Attribute statistics for the KGS data set.

Attribute	RPOS	NM-M	GR	RTA	PHIND	N-D	PE
Maximum	1	2	361.15	32.107	30	28.2	6.321
Minimum	0.0098	1	12.036	1.24	0.55	−20.74	0.096
Average	0.5233	1.5211	63.021	5.3459	12.2258	4.4465	3.7544

Table 3
Class distribution for the KGS data set, including training and testing portions.

Class	1	2	3	4	5	6	7	8	Total
Entire data set	549	539	170	152	422	87	261	94	2274
Training set (two-thirds)	366	359	113	102	281	58	174	63	1516
Testing set (one-third)	183	180	57	50	141	29	87	31	758

5. Research methodology

5.1. WEKA machine learning suite

WEKA, developed and managed by the University of Waikato, is a comprehensive open-source set of object-oriented machine learning and data mining algorithms written in Java (Witten and Frank, 2005). In addition to classification, regression, clustering, and association rule algorithms, it provides methods for data filtering, pre-processing, and visualization. The set of packages provides interfaces to preprocessing routines including feature selection, categorical and numeric learning tasks, performance enhancement of classifiers, evaluation to different criteria such as root mean squared error, and experimental support for verifying the robustness of models. Functionalities such as training, testing, and employing constructed models are provided for all algo-

rithms. Additionally, all algorithms are implemented using the same software architecture, making comparing, combining, and cross-validating algorithms straight-forward.

WEKA has been extensively used in machine learning research to develop new techniques and compare to the state-of-the-art. Other open-source toolsets, such as RapidMiner, provide similar data mining and classification algorithms. WEKA was selected for its ease of use and flexible command-line functionality.

5.2. Multi-agent collaborative learning architecture

The WEKA machine learning suite is utilized as a base for the implementation of the proposed Multi-Agent Collaborative Learning Architecture (Gifford, 2009). The primary uses of WEKA in this architecture are to prepare the data for experimentation and provide implementations of various classification algorithms. Additional Java and script/batch file implementations act as a wrapper for machine learning experiments involving single or multiple learners (i.e., for teams of any size), homogeneous or heterogeneous team composition, independent or collaborative learning (with the ability to vary the number of collaboration events during learning), and combining the decisions of the learners using a variety of accuracy- and vote-based combination techniques. This offers a robust and flexible architecture for machine learning studies involving multiple, collaborative learning agents.

The Multi-Agent Collaborative Learning Architecture was utilized to study aspects of team learning for the KGS rock classification data set. By default, each learner is provided a randomized version (different order of instances for each learner) of the full training data set. Learning takes place by training each classifier using the training data set. At this time, each learner tests its model on the data on which it was trained. Once training is complete, the testing data set is passed through each learner's model and the corresponding class probabilities (predictions) are recorded. The final individual testing predictions are then used for combining decisions from multiple learners. This acts as the final collaboration step, which fuses the knowledge from multiple learners to a single team classifier via accuracy- and vote-based mechanisms.

Decision combination utilizes each learner's classification for each testing instance to arrive at a single team classification per combination method. A learner's prediction consists of a probability that the testing instance belongs to each of the possible classes. The highest probability represents the predicted class for each classifier. The Multi-Agent Collaborative Learning Architecture calculates team classification accuracy for each of the 11 implemented vote-based combination methods. The combination method resulting in the best classification accuracy is selected, reflecting the overall performance of the team.

The following vote-based combination methods are used for evaluation. Selective variations of all methods listed below, with the exception of Select Best, were also examined. Selective variations for each method only take into account learners that have a testing accuracy $\geq$ the mean team testing accuracy. Ties are broken by using the class selected by the learner with the highest accuracy on the testing data, unless otherwise stated.

Max: Selects the class with the absolute maximum probability value out of all learners.

Average: Averages all predictions for each class from all learners and selects the class corresponding to the highest average value.

Multiply: Multiplies all prediction values for each class from all learners and selects the class corresponding to the highest resulting value.

Majority vote: Counts the number of votes (highest class probability) for each class from all learners. The class with the most votes wins. Ties are broken by using the first occurrence of the highest sum of testing accuracies from voters of the class receiving the most votes.

Weighted vote: The class vote from each learner is multiplied by its accuracy on the training data (as a percentage). All votes are then summed, and the class with the highest vote tally wins. This essentially represents the learner with the highest accuracy on the training data having the most influence in which class is selected. Ties are broken by using the first occurrence of the maximum vote value.

Select best: The learner with the highest accuracy on the training data is used for all final predictions.

5.3. Experimental setup

A tournament-style process with rounds based on team size is employed to perform a comprehensive study on all aspects of multi-agent learning for this application. Team configurations exhibiting the highest testing accuracy move on to the next round, where larger teams are constructed from the best teams from the previous round.

The first step for a specific data set is to coarsely test a variety of individual machine learning algorithms, each with a few different settings and initialization seeds, to determine which algorithms perform well in general for the data set. The best settings and corresponding 10-fold cross-validation testing accuracy for each algorithm are recorded, and the top five algorithms are selected to advance to the next round, consisting of teams of size two.

All pairs of these top five algorithms with their best individual settings are then run. This inherently includes homogeneous and heterogeneous team compositions. Similarly, the top teams of size two are selected based on combined testing accuracy. The same process takes place for teams of size four, where the top teams of size four are selected to advance to the final round of size eight teams. Once the round of size eight teams is complete, all results are compiled and a full study can take place.

The goal of evaluating these aspects through detailed team studies is to study whether teams are better than an individual, combining decisions from multiple learners is useful, and that heterogeneous learning algorithms leads to higher task performance/accuracy (Gifford and Agah, 2009). The following sections present experimental results and discuss general aspects of team learning and collaboration discovered as part of this research.

6. Experimental results

This section presents the experimental results on various aspects of team learning and collaboration from the rock classification study. The notation used in the subsequent figures and analyses are abbreviations of learning algorithms and collaboration frequencies. The abbreviations for the 11 utilized learning algorithms are: Naive Bayes (NB) (John and Langley, 1995), Decision Tree (DT) (Quinlan, 1993), Instance-Based KNN (IBK) (Aha et al., 1991), Neural Network (NN), Logistic Regression (LGR) (le Cessie and van Houwelingen, 1992), Radial Basis Function Network (RBF), K* (KST) (Cleary and Trigg, 1995), Decision Table (DTB) (Kohavi, 1995), RIPPER Rule Learner (JRP) (Cohen, 1995), PART Rule Learner (PRT) (Frank and Witten, 1998), and Random Forest (RFT) (Breiman, 2001). The reader is referred to the referenced papers for the algorithms for detail on their underlying structure and theory.

When listing team compositions, the number of learners for each learning algorithm of the team is listed, separated by "+", followed by the collaboration frequency for that team. For example, "3ibk+2nn+2dt+1lgr (M)" represents a team of size eight composed of three IBK, two NN, two DT, and one LGR classifiers which collaborate five times (Medium) during learning. Collaboration frequencies are broken down into N (None, 0 collaborations per experiment), L (Low, 2), M (Medium, 5), and H (High, 8). Finally, Full and Independent (Indp) data learning distribution modes are also abbreviated as shown.

A total of 459 learning experiments were performed, including the variation of team size, composition, and collaboration frequency. Experiments are broken down as follows:

- 11 size one, 264 size two, 100 size four, and 84 size eight teams
- 112 teams for each collaboration frequency
- 95 homogeneous and 364 heterogeneous teams

6.1. Decision combination method

Decision combination methods are the means for combining classifications from multiple learning algorithms to formulate the team's collective decision for an instance requiring classification. Each count for a combination method represents an experiment which resulted in that combination method providing the highest testing accuracy. Ties (in terms of combined decision testing accuracy from all learners) count as a win for each method. The reported win percentages have been normalized, as there are varying numbers of teams of different sizes and more heterogeneous teams, due to heterogeneous teams generally offering higher classification accuracy.

Table 4 presents the most successful five decision combination methods over all experiments for this data set. In general, team size dictates the combination method to choose. Average, Selective Average, Multiply, Selective Multiply, and Selective Max are successful methods for the experiments. We found that Multiply performs best for size two, Selective Multiply for size four, and Average for size eight teams. Selective methods tend to work well for medium team sizes, where some team members perform poorly and should not have their decisions taken into account. As team size grows, averaging decision probabilities from all learners offers a more robust combination method. Max, Majority Vote, Weighted Vote, and Select Best consistently perform poorly for all team sizes. Average, Multiply, and Selective Multiply are good combination methods to choose for heterogeneous teams.

Combination method results for teams as a function of collaboration frequency are similar. For teams which do not collaborate or utilize Medium collaboration, the Selective Multiply methods should be selected. Low and High collaboration teams should utilize Multiply. It appears that collaborating with Medium frequency decreases the variance between combination methods. We again see that the more sophisticated decision combination methods (such as Majority Vote, Weighted Vote, and their Selective variations) perform similarly and with comparatively lower success than the more simple combination methods for this rock classification problem. These results suggest that the Multiply method is superior for this rock classification problem, while averaging performs consistently well for teams of collaborating learners.

Table 4
Most successful five decision combination methods, based on combined testing accuracy, over all experiments.

Combination method	Win count
Multiply	189
Selective multiply	176
Average	160
Selective average	160
Selective max	137

6.2. Best and worst overall team learning results

Teams of size four and eight produce the highest team classification accuracy, whereas teams of size four produce the highest accuracy for their lowest performing teams (the minimum decreases when scaling up to teams of size eight). Certain pairings of learning algorithms perform very poorly in comparison to the majority of the teams. However, as team size increases, the teams perform better and appear to level off between sizes four and eight. Thus, for this application, some learning algorithms perform very poorly when paired together, but increasing the number of learners in the team can help increase team accuracy. Team composition does make a difference in team accuracy, as each learning algorithm models the data in different manners. Combining these different models in different ways can increase accuracy.

A tradeoff is introduced between team size and classification accuracy for size four and eight teams. Team size inherently involves an increase in cumulative training and testing time. Some of the learning algorithms require more resources and time to perform their modeling effort. The K* algorithm is the most memory-intensive algorithm utilized in our experiments. For example, Decision Trees do not require the memory space and time that Neural Networks do. For homogeneous teams, training plus testing time was observed to increase linearly as team size is doubled, as each learner in the team receives the entire training set for training. The difference between Neural Networks and Decision Trees is clear, where in the worst case Decision Tree teams take seconds as opposed to hours to perform the learning and classification task. Thus, if similar accuracy can be achieved with a smaller team size and the application permits, it may not be efficient for a minimal amount of accuracy gain.

We can further investigate team compositions that are generally successful and those that are generally not. Tables 5 and 6 show the best and worst 15 teams overall, based on testing accuracy. Each is accompanied by the associated collaboration frequency and team size. Multiple teams can produce the same testing accuracy. For example, the sixth through eighth teams only differ in the last learner out of all eight. In these cases, the last learner may not be necessary, as the other seven may have

Table 5
Overall best 15 teams, ranked by combined testing accuracy.

Team	Size	Accuracy
1ibk+1kst+1rft+1dt (H)	4	0.84301
2ibk+2kst+2nn+2rft (L)	8	0.84301
2kst+2rft+2ibk+1prt+1nn (L)	8	0.84301
4ibk+2kst+2rft (N)	8	0.84301
2ibk+1kst+1rft (M)	4	0.84169
3rft+2kst+2ibk+1dt (N)	8	0.84169
3rft+2kst+2ibk+1nn (N)	8	0.84169
3rft+2kst+2ibk+1prt (N)	8	0.84169
1ibk+1kst+1nn+1rft (N)	4	0.84037
1ibk+1kst+1nn+1rft (L)	4	0.84037
2rft+1kst+1ibk (M)	4	0.84037
2ibk+2kst+2nn+2rft (N)	8	0.84037
3ibk+2kst+2rft+1nn (L)	8	0.84037
3ibk+2kst+3rft (L)	8	0.83905
3rft+2kst+2ibk+1prt (M)	8	0.83905

Table 6
Overall worst 15 teams, ranked by combined testing accuracy.

Team	Size	Accuracy
2dtb (H)	2	0.27441
2dtb (M)	2	0.29420
2lgr (M)	2	0.36544
1lgr+1dtb (H)	2	0.39314
2lgr (H)	2	0.41161
2nb (M)	2	0.42876
2lgr (L)	2	0.43931
2dtb (L)	2	0.46306
1nb+1lgr (M)	2	0.46570
2nb (H)	2	0.46834
1lgr+1rbf (M)	2	0.47362
1nb+1dtb (H)	2	0.48285
1lgr+1dtb (M)	2	0.48681
1nb+1dtb (M)	2	0.50000
1lgr+1rbf (H)	2	0.51715

covered all of its correctly classified instances. Several team sizes and compositions also achieve identical accuracy, even when the collaboration frequency is varied. For such team compositions, collaboration may not be desired.

Two primary trends are apparent when analyzing the best performing teams. First, highly diverse teams consistently produce high team classification accuracy. Second, larger teams (dominantly size eight teams in the top 15) consistently perform better. Due to the tournament-style experimentation process, it is also observed that the same small set of learning algorithms are highly used for the most successful teams. On the other hand, small team size and more homogeneous team compositions are indicated as prominent trends for the worst performing teams. Many of the worst performing teams collaborate with Medium and High frequency. This does not necessarily mean that collaboration is bad, but rather that these teams do not collaborate well (learn examples from one another during the learning process). It is shown later in this paper that collaboration does have its advantages for specific teams and team sizes. The majority of poor-performing teams here are combinations of poorly performing individual classifiers, or homogeneous teams of the same poor individual classifier on this data set. There is a clear separation between those learning algorithms that offer high accuracy for this data set, and those that do not. Similarly, it can be concluded that teams of strong learners are outperforming collective decisions of multiple weak learners.

6.3. Team diversity: homogeneous vs. heterogeneous teams

Team diversity, or the composition of a team in terms of multiple heterogeneous learning algorithms, is central to a study of team learning dynamics. As different learning algorithms model the data and error differently, combining them can be beneficial for increasing classification accuracy. Teams composed of the same learning algorithm may not experience these advantages, especially when collaboration is involved. Here, we specifically focus on comparing results from homogeneous and heterogeneous teams. When discussing a team's diversity level, it represents the number of unique learning algorithms in that team.

One method for analyzing team diversity is to observe its effect on testing accuracy as a function of the diversity level, as shown in Fig. 3. No team existed which contained six or seven unique learning algorithms, and the testing accuracies are assumed linear between diversity levels five and eight. Teams see a steady increase in testing accuracy coupled with an increase in diversity. This is a promising result, as it supports that the more diverse a

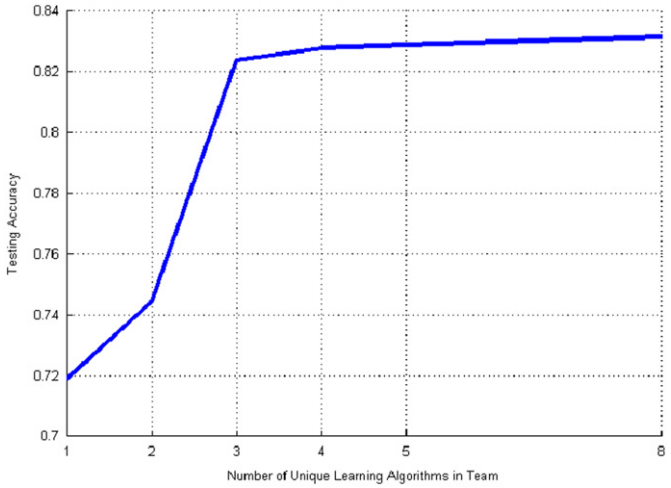

Fig. 3. Average testing accuracy as a function of team diversity.

Table 7
Number of teams of each diversity level (a count of unique learning algorithms in a team), over all, best 50, and worst 50 teams.

Diversity	All teams	Best 50 teams	Worst 50 teams
1	95	0	21
2	240	7	29
3	40	14	0
4	68	24	0
5	12	4	0
6	NA	NA	NA
7	NA	NA	NA
8	4	4	0

team is, the better its accuracy. These results suggest that a diversity level of three (i.e., a team consisting of three heterogeneous learning algorithms) is desirable for this application, as it maximizes testing accuracy.

Table 7 offers a different view of team diversity by showing the number of teams of each diversity level. Diversity level counts are shown for all teams, the best 50 teams, and the worst 50 teams (in terms of testing accuracy). The tournament-style experimentation process produced many teams of diversity levels 1, 2, and 4. By viewing the best 50 teams and their diversity levels, we can make a conclusion about what diversity level achieves highest performance overall. Here, diversity levels of 3 and 4 are observed to be best. This supports earlier results, where the majority of the best performing teams contained four or eight total learners and were dominantly heterogeneous in composition. Diversity levels of 1 and 2 proved to be of lowest performance.

Fig. 4 shows results comparing homogeneous and heterogeneous team performance as a function of team size. Heterogeneous teams perform approximately 5% better on average. Homogeneous teams produce their highest average testing accuracy when of size four, and their lowest testing accuracy when of size two. Homogeneous teams also experience a decrease in accuracy when scaling from size four to eight. These results again support that some teams of size two perform very poorly while others perform quite well. We also see that composing a team of heterogeneous learners is beneficial.

Tables 8 and 9 list the best and worst performing homogeneous and heterogeneous 15 teams, respectively. Successful teams are consistently of large size for both homogeneous and heterogeneous compositions. They are all of

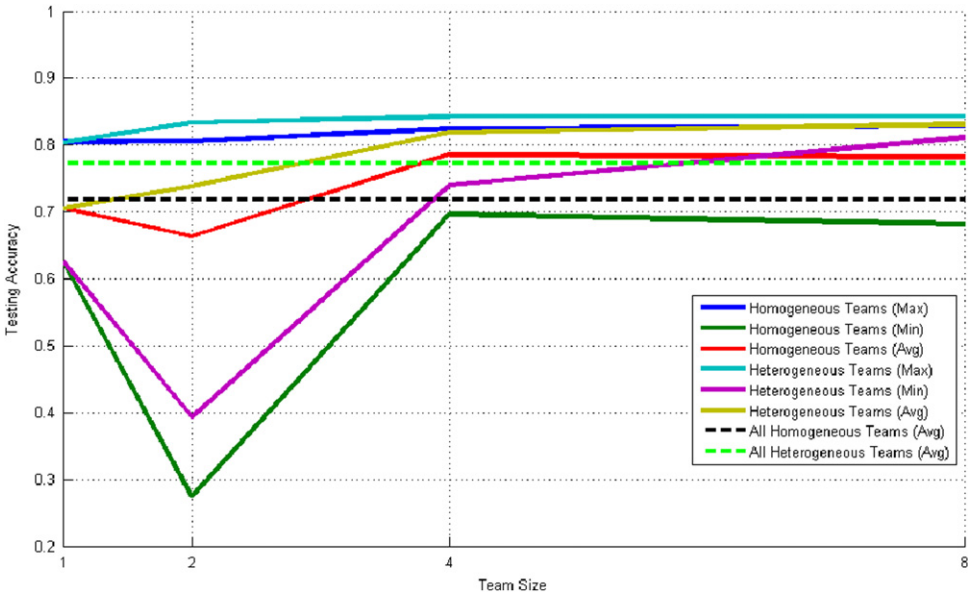

Fig. 4. Homogeneous versus heterogeneous team testing performance comparison by team size. The average testing accuracies over all homogeneous and heterogeneous teams are shown for reference.

Table 8
Overall best 15 homogeneous and heterogeneous teams, ranked by combined testing accuracy.

Homogeneous	Acc.	Heterogeneous	Acc.
8nn (H)	0.828	1ibk+1kst+1rft+1dt (H)	0.843
4nn (L)	0.825	2ibk+2kst+2nn+2rft (L)	0.843
8nn (L)	0.819	2kst+2rft+2ibk+1prt+1nn (L)	0.843
8nn (M)	0.814	4ibk+2kst+2rft (N)	0.843
8rft (H)	0.814	2ibk+1kst+1rft (M)	0.842
4nn (N)	0.811	3rft+2kst+2ibk+1dt (N)	0.842
4rft (M)	0.810	3rft+2kst+2ibk+1nn (N)	0.842
4rft (H)	0.810	3rft+2kst+2ibk+1prt (N)	0.842
4rft (L)	0.809	1ibk+1kst+1nn+1rft (N)	0.840
8nn (N)	0.809	1ibk+1kst+1nn+1rft (L)	0.840
8rft (M)	0.809	2rft+1kst+1ibk (M)	0.840
8rft (L)	0.807	2ibk+2kst+2nn+2rft (N)	0.840
2ibk (L)	0.806	3ibk+2kst+2rft+1nn (L)	0.840
4ibk (L)	0.806	3ibk+2kst+3rft (L)	0.839
2ibk (N)	0.805	3rft+2kst+2ibk+1prt (M)	0.839

Table 9
Overall worst 15 homogeneous and heterogeneous teams, ranked by combined testing accuracy.

Homogeneous	Acc.	Heterogeneous	Acc.
2dtb (H)	0.274	1lgr+1dtb (H)	0.393
2dtb (M)	0.294	1nb+1lgr (M)	0.466
2lgr (M)	0.365	1lgr+1rbf (M)	0.474
2lgr (H)	0.412	1nb+1dtb (H)	0.483
2nb (M)	0.429	1lgr+1dtb (M)	0.487
2lgr (L)	0.439	1nb+1dtb (M)	0.5
2dtb (L)	0.463	1lgr+1rbf (H)	0.517
2nb (H)	0.468	1nb+1lgr (H)	0.522
2rbf (M)	0.558	1rbf+1dtb (H)	0.524
2rbf (H)	0.574	1nb+1rbf (M)	0.533
2nb (L)	0.590	1rbf+1dtb (M)	0.587
1dtb (N)	0.625	1nb+1rbf (H)	0.595
2dtb (N)	0.632	1nb+1lgr (L)	0.631
1nb (N)	0.637	1nb+1dtb (L)	0.633
2nb (N)	0.637	1lgr+1rbf (L)	0.636

diversity levels 3 and 4, with one team exhibiting a diversity level of 5. There are several examples where an increase in collaboration frequency produced higher accuracy, but a decrease in accuracy when collaborating with High frequency. This again supports that collaborating can be beneficial to an extent, after which it degrades performance. A clear advantage is shown in favor of heterogeneous teams, which perform between 12% better in the pool of lowest accuracy teams. Low diversity levels and team sizes are foremost out of those homogeneous and heterogeneous teams that perform the worst. These results parallel those which were discussed earlier for team diversity. Specific learning algorithms consistently offer low accuracy, even when paired with other learners, such as Naive Bayes and Logistic Regression. Team heterogeneity again prevails over homogeneous team compositions.

6.4. Team size: single vs. multiple learners

A study of team learning inherently involves the question, "How large should the team be?" Specifically, it is desirable to study how scaling the team size affects different team dynamics (successful team composition, testing accuracy, collaboration frequency, etc.). The objective is to study what can be gained by using multiple collaborating learning algorithms, as opposed to a single learning algorithm for the entire data set. The following table sheds more light on these aspects, and is intended to supplement the results presented as part of other studies in this paper.

Table 10 lists the five most successful and least successful teams as a function of team size. This table shows the direct result of using a tournament-style experimentation setup, as the best individual learners are highly involved in all successful teams of larger sizes. There exists a coupling between diversity level and team size for the most successful teams. Specifically, the teams that perform best at each team size are nearly equally diverse (team size divided by diversity level). This suggests that the best way to design a team is to compose it of the best performing individual algorithms and construct it so that the team is equally diverse (team size divided by diversity level is between 1 and 2).

Table 10
Most and least successful five teams, ranked by combined testing accuracy, over each team size.

Best teams		Worst teams	
Size 1 team	**Acc.**	**Size 1 team**	**Acc.**
1ibk (N)	0.805	1dtb (N)	0.625
1kst (N)	0.796	1nb (N)	0.637
1nn (N)	0.790	1jrp (N)	0.642
1rft (N)	0.782	1lgr (N)	0.642
1dt (N)	0.697	1prt (N)	0.666
Size 2 team	**Acc.**	**Size 2 team**	**Acc.**
1ibk+1kst (N)	0.834	2dtb (H)	0.274
1ibk+1kst (L)	0.834	2dtb (M)	0.294
1ibk+1kst (M)	0.834	2lgr (M)	0.365
1ibk+1kst (H)	0.834	1lgr+1dtb (H)	0.393
1kst+1rft (M)	0.831	2lgr (H)	0.412
Size 4 team	**Acc.**	**Size 4 team**	**Acc.**
1ibk+1kst+1rft+1dt (H)	0.843	4dt (N)	0.697
2ibk+1kst+1rft (M)	0.842	4dt (H)	0.704
1ibk+1kst+1nn+1rft (N)	0.840	4dt (M)	0.719
1ibk+1kst+1nn+1rft (L)	0.840	4dt (L)	0.723
2rft+1kst+1ibk (M)	0.840	2lgr+2kst (M)	0.740
Size 8 team	**Acc.**	**Size 8 team**	**Acc.**
2ibk+2kst+2nn+2rft (L)	0.843	8dt (L)	0.682
2kst+2rft+2ibk+1prt+1nn (L)	0.843	8dt (N)	0.697
4ibk+2kst+2rft (N)	0.843	8dt (H)	0.704
3rft+2kst+2ibk+1dt (N)	0.842	8dt (M)	0.707
3rft+2kst+2ibk+1nn (N)	0.842	8kst (H)	0.743

As we have observed certain advantages from homogeneous and heterogeneous teams, constructing a team in this manner combines aspects of both. For large teams, this would translate to having a heterogeneous mixture of homogeneous teams. Additionally, benefits of collaboration are observed, but largely at Low and Medium frequencies. Teams of size four and eight, which happen to contain the best overall teams, generally exhibit all of these aspects: heterogeneous mixture of homogeneous learning teams, with Low to Medium collaboration.

Conversely, the least successful teams over all team sizes generally do not follow this equation for success. They exhibit low diversity equality (high value for team size divided by diversity level) and are almost entirely composed of homogeneous learning algorithms. These tables further support the notion that adding individual learners to a team (increasing the team's diversity level) can help fill niches of the data and positively contribute to team success. For example, combining IBK and K* offers a significant increase in team testing accuracy. Accuracy and diversity are observed to increase with team size. Collaboration appears to offer an advantage for teams of size two and four. Teams of size four appear to be the best choice, in terms of testing accuracy and time requirements for this application. Teams of size 5, 6, and 7 were not tested as part of this study.

6.5. Collaborative vs. independent learning

Collaboration allows a learner in a team to distribute difficult training instances to all other learners, increasing the chance that one or more of them will properly learn it and be able to correctly classify that instance as a team once learning has finished. In this respect, it is expected that highly heterogeneous teams will benefit more from collaboration, as each learning algorithm models the data in different ways. Thus, if one learning algorithm cannot correctly classify an instance using its classification model, one or more others would. Homogeneous teams would likely not be able to take advantage of this, contributing to the estimate that collaboration could be detrimental to homogeneous teams. Additionally, collaboration can potentially contribute to overfitting, especially for large teams which collaborate with very high frequency on a small data set. Collaboration was intended to be used on large data sets with moderate team sizes. It was expected that Independent mode teams would require additional collaboration, to gain exposure to more of the training set (albeit the most difficult instances found by other learners). It was expected that some collaboration would be beneficial, but high-frequency collaboration would become detrimental to team success.

Fig. 5 presents collaboration results, comparing homogeneous and heterogeneous collaborating and non-collaborating teams by team size. On average, both collaborating and non-collaborating teams increase in accuracy with team size. Heterogeneous non-collaborating teams perform the best over all team sizes, and heterogeneous collaborative teams perform better than all homogeneous teams. Homogeneous collaborative teams outperform non-collaborative homogeneous teams only for teams of size four. Also, the average testing accuracy over all collaborative teams becomes slightly better than the average over all non-collaborative teams for teams of size eight. The advantage of collaboration over self-learning becomes minimal with teams larger than size four, as only 1% is gained in accuracy when scaling to eight learners.

Fig. 6 illustrates how collaboration frequency affects average testing accuracy as functions of team size and team heterogeneity. There are only two cases where collaboration causes an increase in testing accuracy: size eight homogeneous teams which collaborate with Medium frequency, and size four homogeneous teams which collaborate with Low frequency. This suggests that collaboration is more effective for larger teams which are homogeneous in composition. For other team sizes, regardless of team composition, collaboration causes a reduction in testing accuracy. These results support that collaborating can be beneficial at Low to Medium frequencies.

Another important aspect of collaboration to investigate is the improvement, if any, that it provides over not collaborating. Fig. 7 presents results on the effect of collaboration frequency, and how it is tied to team size and team heterogeneity. Improvement is calculated by dividing the average testing accuracy of a collaborative team over the average testing accuracy of a non-collaborative team of the same size. A value above 1.0 signifies that collaborating with that frequency improves accuracy for that team size. Therefore, each collaboration frequency's effect on team accuracy can be analyzed for various team sizes.

There exists two primary cases where collaboration offers an improvement in testing accuracy: homogeneous teams of size four (for each collaboration frequency), and homogeneous teams of size eight which collaborate with Medium frequency. Heterogeneous teams of size eight collaborating with Low frequency are similar, on average, to the same teams which do not collaborate. All other collaboration frequencies and associated team compositions resulted in a decrease in testing accuracy.

Table 11 lists the best and worst 10 collaborative and non-collaborative teams. This table illustrates what team compositions perform best for collaborative teams and non-collaborative teams. Of the most successful teams, both collaborative and non-collaborative teams are of larger team sizes and more equally diverse. Overall performance is nearly identical for collaborative and non-collaborative teams; however, more best-accuracy teams utilize collaboration. Examining the lowest accuracy teams, non-collaborative teams perform

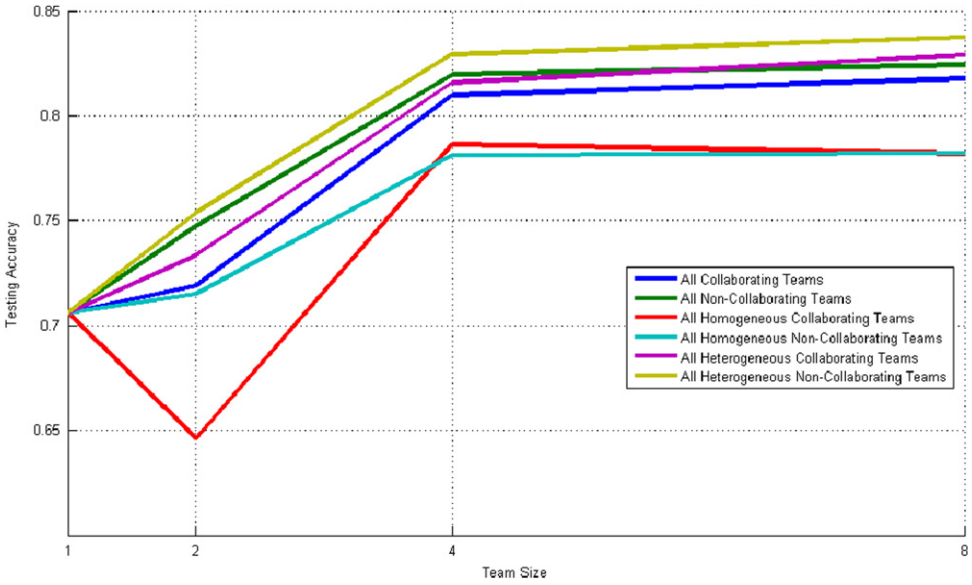

Fig. 5. Collaboration versus independent learning average testing accuracy by team size.

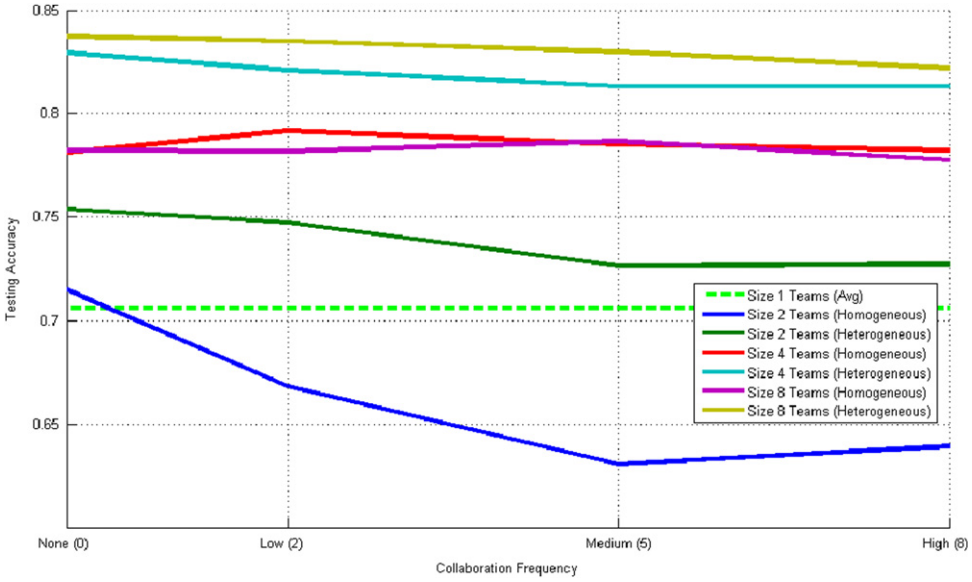

Fig. 6. Collaboration versus independent learning average testing accuracy by collaboration frequency for homogeneous and heterogeneous teams.

substantially better than those which collaborate. Results indicate that collaboration frequency is coupled with both team size and learning mode. We therefore conclude that collaboration with Low or Medium frequency will lead to the best results.

6.6. Individual learner contribution

The contribution and success of individual machine learning algorithms were investigated. By analyzing a learning algorithm's contribution to success, we can extract which learning algorithms are generally successful for this application, and those which also perform well together. If teams that contain a certain learning algorithm consistently perform better than others not containing that learning algorithm, then that algorithm can be considered valuable (or a major contributor) to a team's combined testing classification accuracy.

This is measured in two primary ways. First, as shown in Fig. 8, average testing accuracy is compared for each learning algorithm over all teams in which that algorithm participated. On average, NB, LGR, RBF, DTB, and JRP perform the worst, whereas IBK, KST, NN, RFT, and DT perform the best. Teams containing DTB, LGR, and NB learners are less stable, as they can produce very poor classification accuracy when included in a team. On the other hand, learning algorithms such as IBK, KST, and NN are stable learning algorithm choices, as their minimum team accuracies are still comparably high.

Each algorithm has a similar maximum accuracy, meaning that it was involved in one or more teams which contained heterogeneous learning algorithms that collectively produced high testing accuracy. These results indicate that for this application learning algorithms such as IBK, KST, NN, RFT, and DT are reliable when constructing a team of multiple classifiers. Learning algorithms such as NB, LGR, RBF, DTB, and JRP are not as reliable. Combining these algorithms is encouraged, to produce a team

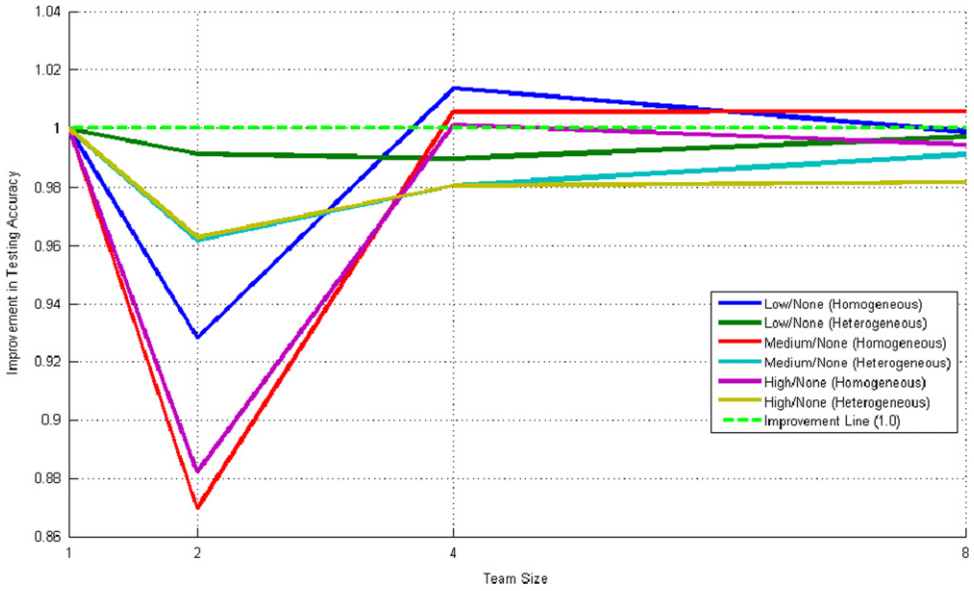

Fig. 7. Effect/improvement of collaboration on average testing accuracy versus non-collaborating teams by team size for homogeneous and heterogeneous teams.

Table 11
10 best and worst performing collaborating and non-collaborating teams, based on average testing accuracy.

Best teams		**Worst teams**	
Collaborative	**Acc.**	**Collaborative**	**Acc.**
1ibk+1kst+1rft+1dt (H)	0.843	2dtb (H)	0.274
2ibk+2kst+2nn+2rft (L)	0.843	2dtb (M)	0.294
2kst+2rft+2ibk+1prt+1nn (L)	0.843	2lgr (M)	0.365
2ibk+1kst+1rft (M)	0.842	1lgr+1dtb (H)	0.393
1ibk+1kst+1nn+1rft (L)	0.840	2lgr (H)	0.412
2rft+1kst+1ibk (M)	0.840	2nb (M)	0.429
3ibk+2kst+2rft+1nn (L)	0.840	2lgr (L)	0.439
3ibk+2kst+3rft (L)	0.839	2dtb (L)	0.463
3rft+2kst+2ibk+1prt (M)	0.839	1nb+1lgr (M)	0.466
4rft+2kst+2ibk (L)	0.839	2nb (H)	0.468
Independent	**Acc.**	**Independent**	**Acc.**
4ibk+2kst+2rft (N)	0.843	1dtb (N)	0.625
3rft+2kst+2ibk+1dt (N)	0.842	2dtb (N)	0.632
3rft+2kst+2ibk+1nn (N)	0.842	1nb (N)	0.637
3rft+2kst+2ibk+1prt (N)	0.842	2nb (N)	0.637
1ibk+1kst+1nn+1rft (N)	0.840	1nb+1lgr (N)	0.640
2ibk+2kst+2nn+2rft (N)	0.840	1jrp (N)	0.642
1ibk+1rft+1lgr+1kst (N)	0.838	1lgr (N)	0.642
1kst+1rft+1ibk+1prt (N)	0.838	2lgr (N)	0.642
2kst+2rft+2ibk+1prt+1dt (N)	0.838	1lgr+1dtb (N)	0.657
2kst+2rft+2ibk+1prt+1nn (N)	0.838	1dtb+1jrp (N)	0.657

that has a heterogeneous mixture of strong homogeneous classifiers.

As shown in Fig. 9, the number of total teams which each learning algorithm participated in can be compared. This figure shows individual learning algorithm accuracies on the entire data set (i.e., size one teams), and learning algorithm participation over all team learning experiments as a percentage. The byproduct of the tournament-style experimental setup is the use of the best performing algorithms in more teams. It is also evident that the IBK, KST, and RFT classifiers are used in 20% more teams than the next best set of classifiers. This is a result of combining teams of size two to form teams of size four, and combining teams of size four to form teams of size eight. These three classifiers comprised more size two and four teams in comparison. The NB, DTB, JRP, and RBF classifiers were utilized the least over all experiments.

7. Conclusion

Using the proposed Multi-Agent Collaborative Learning Architecture, we were able to obtain approximately 84.5% absolute accuracy, an improvement of 6.5% over the best results that KGS was able to achieve on the same rock facies classification data set. Several novel heuristics were discussed for constructing teams of multiple collaborative classifiers. A tradeoff between team size and team composition also became apparent, including the time required to complete the training, testing, and combination processes.

Highly diverse and heterogeneous team compositions are preferred, due to their robustness, complementary nature, and general good performance. Heuristics for team construction were also developed. A heterogeneous mixture of strong homogeneous teams, especially those compositions that are equally diverse (team size divided by diversity level being 1 to 2), allows a team of multiple agents to classify with high combined accuracy. Furthermore, collaboration was found to be most beneficial for homogeneous teams of size four, and generally produced an increase in testing accuracy when utilized with Low to Medium frequency. Size four and eight teams proved to be well-suited for this data set. Combination methods of Average, Selective Average, Multiply, and Selective Multiply were observed to offer the highest combined team testing accuracy. Lastly, the IBK, KST, NN, RFT, and DT algorithms performed consistently better than the others. For this difficult classification problem, utilizing a team of multiple heterogeneous collaborative machine learning algorithms was necessary to maximize accuracy.

Collaboration proved to be beneficial in certain situations, but only at Low to Medium frequency. Collaboration is more useful for distributed learning problems, and more challenging data sets. Collaboration increases the learning rate (i.e., knowledge is gained more rapidly) and distributes the responsibility of learning challenging instances to each member of the team. The repetition and multiple exposures of instances that collaboration offers, allows the teams of classifiers the opportunity to spend less learning effort on simple instances and a more dedicated effort on

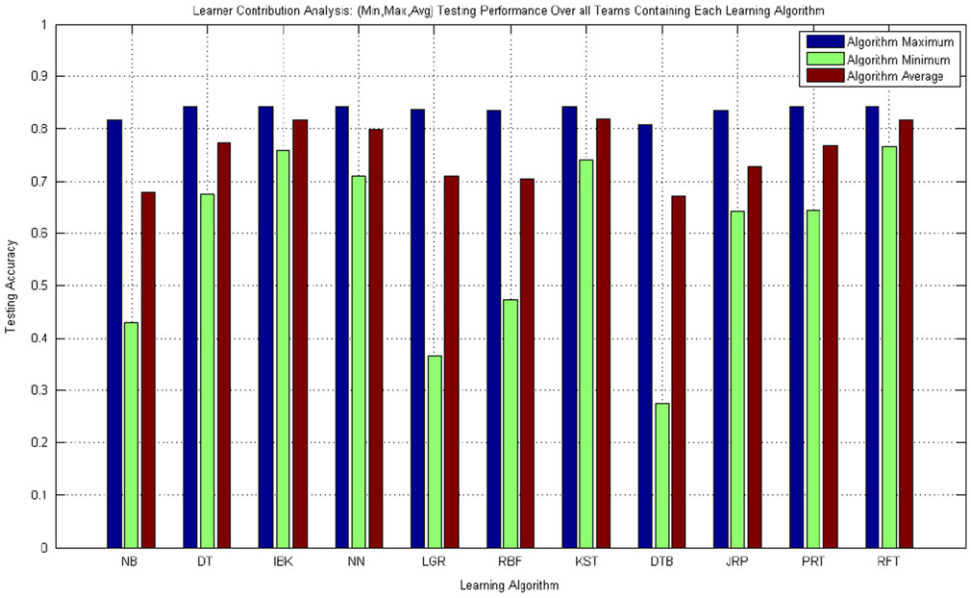

Fig. 8. Minimum, maximum, and average testing accuracies over all teams containing one or more of each learning algorithm.

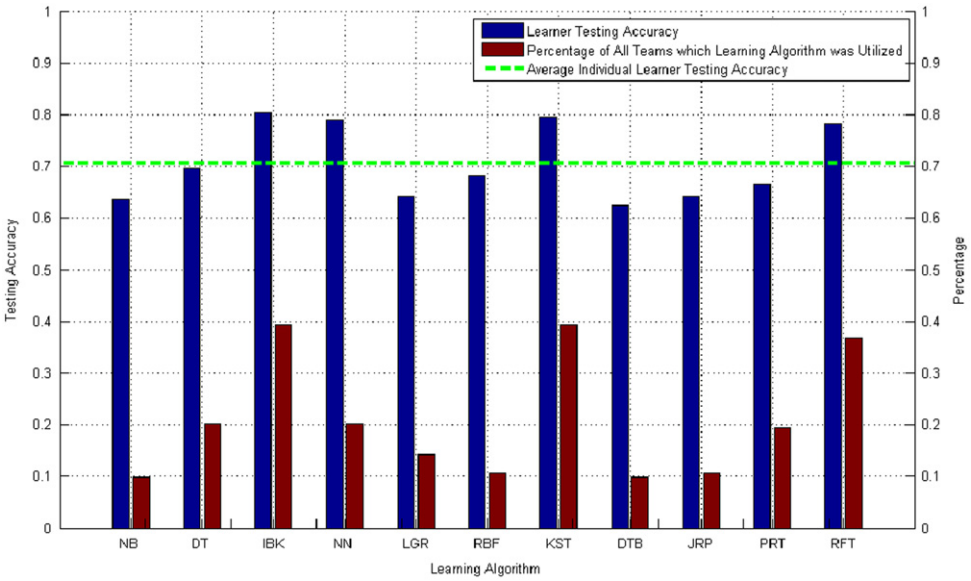

Fig. 9. Individual learning algorithm testing accuracies for entire data set (team size of 1), accompanied by each learning algorithm's overall participation by the percentage of teams in which it was a member.

those instances that one or more learning algorithms are misclassifying during the learning process. Multiple collaboration events integrates an interactive aspect to the learning process, which appears to be required to achieve the best testing accuracy for the data set.

The current Multi-Agent Collaborative Learning Architecture is a serial process: the learning algorithms are not learning and collaborating in a truly parallel fashion. One possible extension to this architecture is to incorporate parallel processing for large-scale, massively-parallel classification problems. Grid WEKA (Zuo, 2004) and WEKA-Parallel (Celis et al., 2003) are currently available for parallel machine learning applications. Efficient collective communication for sharing knowledge and collaborating could be achieved through the use of a parallel cluster, employing the message passing interface (MPI) (Thakur et al., 2005). As communication is a primary source of overhead and requires significant computing and energy resources for hardware systems, introducing parallel aspects to multi-agent machine learning can aid in its application to larger problems.

Results presented in this paper support the application of multi-agent machine learning and collaboration to challenging, real-world classification problems. The Multi-Agent Collaborative Learning Architecture is currently being applied to other domains, such as stand-off device classification, novelty detection in space imagery, and medicine. Multi-agent collaborative learning and classification can be applied to many different problems and domains, namely, decision support, bioinformatics, medicine, gene sequencing, etc. Military and medical institutions are increasingly discovering needs which only advanced machine learning and collaborative techniques can address.

There are many facets of team learning that require further investigation. Team learning introduces social interaction, team-building, and knowledge transfer that are inherent in human teams. In order to achieve true team-based learning and collaboration, models are needed for how humans cognitively learn, interact, and teach one another. New and novel machine learning methods are also needed for information sharing and collaboration. Incorporating agent trust, confidence, and negotiation, along with using such concepts to intelligently reason to arrive at a collective decision, may allow for higher accuracy classification or task performance. Investigating other next-generation learning architectures, such as deep learning (incorporating both substance and meaning) and hierarchical systems, is also important to advance the state-of-the-art in multi-agent machine learning.

Machine learning and pattern recognition will continue to be applied to other disciplines to advance the state-of-the-art. With the growing importance of resources and prediction of events, sophisticated classification techniques for determining location of subsurface reservoirs, disease outbreaks, and the change in weather and climate will experience increased attention. Many applications exist that can benefit from use of such technology, especially with the significant amount of data available from today's information systems. Finding new ways in interdisciplinary fields to leverage these approaches should be a focus for the future.

Acknowledgements

The authors would like to thank Geoffrey Bohling at Kansas Geological Survey for providing the wireline well log data set for this study. Author C.M. Gifford performed this research while part of the Department of Electrical Engineering and Computer Science at the University of Kansas. This material is based upon work supported by the National Science Foundation under Grant no. ANT-0424589. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors are grateful to the reviewers for their time and constructive criticism in preparing this article for publication.

References

Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Machine Learning 6, 37–66.

Al-Faraj, S.A.A., 1998. Reservoir formation facies identification using decision tree learning. Master's Thesis, King Fahd University of Petroleum and Minerals.

Bhatt, A., 2002. Reservoir properties from well logs using neural networks. Ph.D. Thesis, Department of Petroleum Engineering and Applied Geophysics, Norwegian University of Science and Technology.

Bhattacharya, B., Solomatine, D., 2006. Machine learning in soil classification, neural networks. Earth Sciences and Environmental Applications of Computational Intelligence 19 (2), 186–195 (Special issue).

Bhatt, A., Helle, H.B., Ursin, B., 2001. Application of committee machines in reservoir characterisation while drilling: a novel neural network approach in log analysis. In: Proceedings of the Nordic Symposium on Petrophysics, Trondheim, Norway, 2001, pp. 1–15.

Bohling, G., Dubois, M., 2003. An integrated application of neural network and markov chain techniques to prediction of lithofacies from well logs. Technical Report 2003-50.

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

Celis, S., Musicant, D.R., 2003. Weka-parallel: machine learning in parallel. Technical Report, Department of Mathematics and Computer Science, Carleton College, uRL: ∠http://sourceforge.net/projects/weka-parallel/〉.

Chen, K., Wang, L., Chi, H., 1997. Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification. International Journal of Pattern Recognition and Artificial Intelligence 11 (3), 417–445.

Cleary, J.G., Trigg, L.E., 1995. K*: an instance-based learner using an entropic distance measure. In: Proceedings of the 12th International Conference on Machine learning, 1995, pp. 108–114.

Cohen, W.W., 1995. Fast effective rule induction. In: Proceedings of the 12th International Conference on Machine Learning, pp. 115–123.

dos Santos, R.V., Artola, F., da Fontoura, S., Vellasco, M., 2002. Lithology recognition by neural network ensembles. In: Lecture Notes in Computer Science: Advances in Artificial Intelligence, vol. 2507. Springer, London, UK, pp. 302–312.

Dubois, M., Byrnes, A., Bohling, G., Seals, S., Doveton, J., 2003. Statistically-based lithofacies predictions for 3-D reservoir modeling: examples from the Panoma (Council Grove) Field, Hugoton Embayment, Southwest Kansas. In: Proceedings of the American Association of Petroleum Geologists Annual Convention, vol. 12. A44, Salt Lake City, Utah.

Dubois, M., Bohling, G., Chakrabarti, S., 2004. Comparison of rock facies classification using three statistically based classifiers. Technical Report 2004-64.

Dubois, M.K., Bohling, G.C., Chakrabarti, S., 2007. Comparison of four approaches to a rock facies classification problem. Computers & Geosciences 33 (5), 599–617.

Dzeroski, S., Zenko, B., 2004. Is combining classifiers better than selecting the best one? Machine Learning 54 (3), 255–273

Ebrahimpour, N.H.R., 2007. Combining multiple classifiers: diversify with boosting and combining by stacking. International Journal of Computer Science and Network Security 7 (1), 127–131.

Frank, E., Witten, I.H., 1998. Generating accurate rule sets without global optimization. In: Proceedings of the 15th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA, pp. 144–151.

Gifford, C.M., 2009. Collective machine learning: team learning and classification in multi-agent systems. Ph.D. Thesis, Electrical Engineering and Computer Science Department, University of Kansas, Lawrence, KS (November 2009).

Gifford, C.M., Agah, A., 2009. Sharing in teams of heterogeneous, collaborative learning agents. International Journal of Intelligent Systems 24 (2), 173–200.

John, G.H., Langley, P., 1995. Estimating continuous distributions in Bayesian classifiers. In: Jaya, S. (Ed.), Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Mateo CA, pp. 338–345.

KGS, Kansas Geological Survey, 2009. 〈http://www.kgs.ku.edu/〉.

Kim, J.H., Kim, K.K., Suen, C.Y., 2000. Hybrid schemes of homogeneous and heterogeneous classifiers for cursive word recognition. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, Amsterdam, pp. 433–442.

Klein, D., Toutanova, K., Ilhan, H.T., Kamvar, S.D., Manning, C.D., 2002. Combining heterogeneous classifiers for word-sense disambiguation. In: Proceedings of the ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, vol. 8. pp. 74–80.

Kohavi, R., 1995. The power of decision tables. In: Proceedings of the European Conference on Machine Learning, pp. 174–189.

Kotsianti, S., Kanellopoulos, D., 2007. Combining bagging, boosting and dagging for classification problems. In: Lecture Notes in Computer Science: Knowledge-Based Intelligent Information and Engineering Systems, Springer, Berlin, Heidelberg, pp. 493–500.

Kuncheva, L.I., 2002. Switching between selection and fusion in combining classifiers: an experiment. IEEE Transactions on Systems, Man, and Cybernetics 32 (2), 146–156.

le Cessie, S., van Houwelingen, J., 1992. Ridge estimators in logistic regression. Applied Statistics 41 (1), 191–201.

LaBelle, D., Bares, J., Nourbakhsh, I., 2000. Material classification by drilling. In: Proceedings of the International Symposium on Robotics and Automation in Construction, Taipei, Taiwan.

Liu, Y., Sacchi, M.D., 2003. Mapping rock mechanical properties with seismic attribute-based support vector machine (SVM) technique. In: Proceedings of the CSPG/CSEG Joint Convention.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

Rogova, G.L., Bursik, M.I., Hanson-Hedgecock, S., 2007. Interpreting the pattern of volcanic eruptions: intelligent system for tephra layer correlation. In: Proceedings of the International Conference on Information Fusion, pp. 1–7.

Saggaf, M., Nebrija, E.L., 2000. Estimation of lithologies and depositional facies from wire-line logs. American Association of Petroleum Geologists Bulletin 84 (10), 1633–1646.

Santos, R., Vellasco, M., Artola, F., da Fontoura, S., 2003. Neural net ensembles for lithology recognition. In: Lecture Notes in Computer Science: Multiple Classifier Systems, vol. 2709. Springer, London, UK, pp. 246–255.

Shang, F., Zhang, X., Zhao, T., 2008. Application of a mixed kernel in the oil water-flooded layer identification. International Journal of Computer Science and Network Security 8 (1), 102–106.

Shi, Z., Luo, P., Hao, Y., Li, G., Stumptner, M., He, Q., Quirchmayr, G., 2005. In: International Federation for Information Processing, Intelligent Information Processing II, vol. 163. London, UK, pp. 373–382.

Tartakovsky, D., Wohlberg, B., 2004. Delineation of geologic facies with statistical learning theory. Geophysical Research Letters 31 (18), L18502.

Thakur, R., Rabenseifner, R., Gropp, W., 2005. Optimization of collective communication operations in MPICH. International Journal of High Performance Computing Applications 19 (1), 49–66.

Ting, K., Low, B., 1996. Theory combination: an alternative to data combination, Technical Report 96/19.
Tsoumakas, G., Angelis, L., Vlahavas, I., 2005. Selective fusion of heterogeneous classifiers. Intelligent Data Analysis 9 (6), 511–525.
Tumer, K., Ghosh, J., 1996. Error correlation and error reduction in ensemble classifiers. Connection Science 8 (3), 385–404.
van der Baan, M., Jutten, C., 2000. Neural networks in geophysical applications. Geophysics 65 (4), 1032–1047.
Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, second ed. Morgan Kaufmann, San Francisco.
Wohlberg, B., Tartakovsky, D., Guadagnini, A., 2006. Subsurface characterization with support vector machines. IEEE Transactions on Geoscience and Remote Sensing 44 (1), 47–57.
Wong, P., Gedeon, T., Taggart, I., 1995. An improved technique in porosity prediction: a neural network approach. IEEE Transactions on Geoscience and Remote Sensing 33 (4), 971–980.
Zuo, X., 2004. High Level Support for Distributed Computation in WEKA, Master of Computational Science, University College, Dublin, Ireland, URL: 〈http://userweb.port.ac.uk/khusainr/weka/index.html〉.