12-2009

Context-dependent fusion with application to landmine detection.

Lijun Zhang
University of Louisville

CONTEXT-DEPENDENT FUSION WITH APPLICATION TO LANDMINE DETECTION

By

Lijun Zhang
B.Sc., EE, Harbin Institute of Technology, CHINA, 1999
M.Sc., EE, Shanghai Jiaotong University, CHINA, 2005

A Dissertation
Submitted to the Faculty of the
Graduate School of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

December 2009

CONTEXT-DEPENDENT FUSION WITH APPLICATION TO LANDMINE DETECTION

By

Lijun Zhang
B.Sc. EE, Harbin Institute of Technology, CHINA, 1999
M.Sc. EE, Shanghai Jiaotong University, CHINA, 2005

A Dissertation Approved on

August 18, 2009

by the Following Reading and Examination Committee:

Hichem Frigui, Ph.D., Dissertation Director

Aly A. Farag, Ph.D.

J. P. Mohsen, Ph.D.

Olfa Nasraoui, Ph.D.

Ayman El-Baz, Ph.D.

Ibrahim N. Imam, Ph.D.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor Dr. Hichem Frigui for his support, guidance and encouragement during my education at University of Louisville. His patient and enthusiasm for research went a long way to make this dissertation a reality.

I would also like to thank all my dissertation committee members Dr. Aly A. Farag, Dr. J. P. Mohsen, Dr. Olfa Nasraoui, Dr. Ayman El-Baz and Dr. Ibrahim N. Imam for being on my supervisory committee and for their comments, suggestions and assistance on this dissertation.

I would like to thank all the research members on the Multimedia Research Laboratory at University of Louisville for their friendship and help: Joshua Caudill, Joson Meredith, Aleksey Fadeev, Oualid Missaoui, Anis Hamdi, Ahmed Chamseddine, Naouel Baili, Mohamed Maher Ben Ismail, Ouiem Bchir, Andrew D. Karem and Hisham Sliman.

Also many thanks to my friends Dr. Dongqing Chen, Weizhong Zhang, Huihang Dong, Yinan Cui, and Hui Wang from the Department of Electrical & Computer Engineering, and Dr. Zhiyong Zhang from Yahoo Company for their help and support.

Finally, I would give my deepest appreciation to my parents and my girl friend Ming Wang for their continued support, unconditional love and endless encouragement.

ABSTRACT

CONTEXT-DEPENDENT FUSION
WITH APPLICATION TO LANDMINE DETECTION

Lijun Zhang

August 18, 2009

Traditional machine learning and pattern recognition systems use features to describe the sensor data and a classifier (also called "expert" or "learner") to determine the true class of a given pattern. However, for complex detection and classification problems, involving data with large intra-class variations and noisy inputs, no single source of information can provide a satisfactory solution. As a result, combination of multiple classifiers is playing an increasing role in solving these complex pattern recognition problems, and has proven to be a viable alternative to using a single classifier. In this dissertation, we introduce a novel Context-Dependent Fusion (CDF) approach, and apply this method to fuse multiple algorithms which use different types of features and different classification methods on multiple sensor data.

Our CDF approach is motivated by the observation that there is no single algorithm that can consistently outperform all other algorithms. In fact, the relative performance of different algorithms can vary significantly depending on several factors such as the extracted features and the characteristics of the target class. The CDF method is a local approach that adapts the fusion to different regions of the feature space. The goal is to take advantage of the strengths of few algorithms in different regions of the feature space

without being affected by the weaknesses of the other algorithms, and also avoid the loss of potentially valuable information provided by weak classifiers by considering their output as well.

The proposed fusion has three main interactive components. The first one, Context Extraction, partitions the composite feature space into groups of similar signatures or contexts. For this task, we explore a novel algorithm that performs clustering and feature discrimination to cluster and identify the relevant features for each cluster. The second component assigns an aggregation weight to each detector's decision in each context based on its relative performance within the context. We have developed, implemented and composed six different weight assignment methods which are embedded into this component. The third component of the CDF combines the multiple decisions with the learned weights to make a final decision.

The proposed approach was applied to the problem of landmine detection. Detection and removal of landmines is a serious problem affecting civilians and soldiers worldwide. Varieties of sensors and algorithms have been proposed or are under investigation for landmine detection. Extensive testing of these methods has shown that the relative performance of different detectors can vary significantly depending on the mine types, geographical sites, soil and weather conditions, burial depths, etc. Therefore, fusion methods that can take advantages of the strengths of different sensors and algorithms into account, overcome their weaknesses, adapt to the rapidly changing environmental conditions, and achieve a higher accuracy than any individual algorithm are needed. Thus, multi-sensor and multi-algorithm fusion which can adapt to different environments are critical components in land mine detection.

The proposed methods were tested exhaustively with several real landmine data sets collected on the field. Three different data sets were selected to be included in this dissertation to illustrate the performances of the proposed CDF method. The first data set was

collected by a Ground Penetrating Radar (GPR) mounted on a vehicle. This data covered a ground area of over $40,000m^2$. The second data set was collected by an Autonomous Mine Detection system that includes two different types of sensors: GPR and wideband Electromagnetic induction (WEMI). The third data set was collected by an Airborne Hyperspectral Imagery (AHI) data and covers approximately $145,000m^2$ of terrain. The results showed that the proposed method can identify meaningful and coherent clusters and that different expert algorithms can be identified under the different contexts. Our experiments have also indicated that our approach outperformed all individual detectors and several state of the art fusion methods significantly. More importantly, the results can be achieved with efficient computation, and can be interpreted. Consequently, the US Army is considering implementing these methods into current landmine detection system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

For complex detection and classification problems involving data with large intra-class variations and noisy inputs, perfect solutions are difficult to achieve, and no single source of information can provide a satisfactory solution. As a result, combination of multiple classifiers (or multiple experts) is playing an increasing role in solving these complex pattern recognition problems, and has proved to be a viable alternative to using a single classifier. Classifier combination is mostly a heuristic approach and is based on the idea that classifiers with different methodologies or different features can have complementary information. Thus, if these classifiers cooperate, group decisions should be able to take advantages of the strengths of the individual classifiers, overcome their weaknesses, and achieve a higher accuracy than any individual's.

Over the past few years, a variety of schemes have been proposed for combining multiple classifiers. Techniques for classifier fusion are drawn from a diverse set of more traditional disciplines including statistical estimation, digital signal processing, control theory, artificial intelligence, and classic numerical methods. The characteristics of the commonly used techniques will be examined in more details in Chapter II.

Methods for combining multiple classifiers can be classified into two main categories: classifier selection and classifier fusion. Classifier selection methods assume that the classifiers are complementary, and that their expertise varies according to the different areas of the feature space. For a given test sample, these methods attempt to predict which classifiers are more likely to be correct. Some of these methods consider the output of only

one classifier to make the final decision [2]. Others, combine the output of multiple "local expert" classifiers [3]. On the other hand, classifier fusion methods assume that the classifiers are competitive and are equally experienced over the entire feature space. For a given test sample, the individual classifiers are applied in parallel, and their outputs are combined in some manner to take a group decision.

One approach for building multiple classifiers is based on bagging and boosting [4]. Each classifier is trained using a different subset of the training set. The different subsets are obtained from the original using sampling. The final output is obtained by voting. Bagging specifically refers to the process of generating training subsets by sampling with replacement multiple times. A classifier is trained on each subset. All classifiers are used to classify a test sample. The outputs are combined via voting. Boosting generally refers to a more sequential process of building multiple classifiers on a training set. The general idea is that an initial classifier is trained on the training set. Points for which the initial classifier performs "poorly" are weighted more strongly in training a different classifier. The process is repeated multiple times in order to try and build a multi-classifier system consisting of classifiers that perform well on subsets of the training set. Boosting can cause problems by over-fitting classifiers on subsets of the training data [5].

Another way to categorize classifier combination methods is based on the way they select or assign weights to the individual classifiers. Some methods are global and assign a degree of worthiness, that is averaged over the entire training data, to each classifier. Other methods are local and adapt the classifiers' worthiness to different data subspaces. Intuitively, the use of data-dependent weights, when learned properly, provides higher classification accuracy. This approach requires partitioning the input samples into regions during the training phase. The partition can be defined from the space of individual classifier decisions, according to which classifiers agree with each other [6], or by features of the input space [7]. Then, the best classifier for each region is identified and is designated as

the expert for this region [2]. Conversely, the partitioning can be defined such that each classifier is an expert in one region [8]. This approach may be more efficient, however, its implementation is not trivial. In the classification phase, the region of an unknown sample is identified, and the output of the classifier responsible for this region is used to make the final decision. Data partition and classifier selection could also be made dynamic during the testing phase [9, 10]. In this case, the accuracy of each classifier (with respect to the training samples) is estimated in local regions of the feature space in the vicinity of the test sample. The most accurate classifier is selected to classify the test sample. This approach may be more efficient, however, entirely discarding other classifiers can be counterproductive since the potentially valuable information introduced in other classifiers may be ignored.

In this dissertation, we propose a new approach, called Context-Dependent Fusion (CDF), which is a local method and focuses on the multi-algorithm fusion problem. A multi-algorithm classification system is more general than a multi-classifier system which consists of a set of algorithms, each of which operates on feature data to ultimately produce a set of class confidence values. The features extracted by each algorithm are generally different (in fact, each algorithm could be a multi-classifier system) and could in fact be extracted from data acquired from different sensors. Performance of different algorithms in a multi-algorithm classification system can vary due to factors other than local minima of objective functions. By combining multiple algorithms, we can take advantage of their strengthens and overcome their limitations.

The proposed CDF approach has three main components. The first component, called *Context Extraction*, is completely unsupervised. In this component, the features extracted by the different algorithms (from different sensors) are combined. Then, a clustering algorithm is used to partition the training signatures into groups of similar signatures, or contexts, and learn the relevant features within each context. Here, we are assuming that

signatures that have similar response to different algorithms share some common features, and should be assigned to the same cluster. The second component of the CDF, called *Algorithm Fusion*, assigns an aggregation weight to each algorithm's confidence value in each context based on its relative performance within the context. Training data from each identified context could be used to learn the optimal fusion parameters and identify "*local experts*" for that region of the feature space. We will investigate, test and compare various weight assignment methods in this part. The third component, i.e. *Decision Making*, uses the learned weights within each context to make a final decision on a test pattern.

The proposed fusion methods are implemented and integrated within a complete landmine detection system. Detection and removal of landmines is a significant research problem [11, 12]. It is estimated that over 100 million landmines are buried in over 80 countries and that 26,000 people a year are killed or maimed by a landmine [13]. The research problem for data analysis is to determine how reliably landmines can be detected and distinguished from other subterranean objects using sensor data. Difficulties arise from the variability of landmine types, soil and weather conditions, terrains, and so on. Therefore, detection algorithms which can adopt to changing conditions are needed for detecting buried landmineds. Thus, multi-classifier, multi-algorithm, and multi-sensor fusion is a critical component in landmine detection. In this dissertation, the proposed Context-Dependent fusion methods are trained and tested with large landmine data sets collected from various regions under different conditions, including various mine types from different sensors, and better results are reported and discussed.

The organization of the next of this dissertation is as follows. Chapter I introduces the background, goals, and terminology of fusion methods. Chapter II discusses and analyzes several global fusion methods. These methods will be used for evaluation and comparison with the proposed local fusion method. Chapter III introduces the landmine detection problem and provides a literature review of different landmine detection sensors and

algorithms. The output of these algorithms will be fused using our proposed local fusion. Chapter IV introduces the proposed fusion methodology. We motivate the need for local fusion, outline the architecture of the proposed context-dependent fusion, and highlight its advantages over the global fusion. In Chapter V we propose six different methods for local weight assignment. Results of applying CDF to fuse the output of multiple landmine detection algorithms are compared on data from multiple sensors in Chapter VI. Chapter VII summarizes the contributions and outlines the potential future work.

CHAPTER II

RELATED WORK ON CLASSIFIERS FUSION

Fusion of data/information can be carried out at three levels of abstraction closely connected with the flow of the classification process: *data level fusion, feature level fusion, and decision level fusion. Data level fusion*, also called low level fusion, combines several sources of raw data to produce new raw data that is expected to be more informative and synthetic than the inputs. For example, in image processing, images presenting several spectral bands of the same scene are fused to produce a new image that ideally contains, in a single channel, all (or most) of the information available in the various spectral bands. *Feature level fusion*, also called intermediate level fusion, combines various features. These features may come from several raw data sources (e.g. several sensors) or from the same raw data. In the latter case, the objective is to find relevant features among available features that might come from several feature extraction methods. The objective is to obtain a limited number of relevant features. Examples of feature level fusion include the Principal Component Analysis (PCA) [14], and Diabolo shaped Multi-layer Perceptrons (MLP) [15] for the non-linear counterpart. *Decision level fusion*, also called high level fusion, combines decisions coming from several experts. By extension, one speaks of decision fusion even if the experts return a confidence (score) and not a decision. There are some theories about the first two levels of information fusion, for example, transforming the numerical, interval and linguistic data into a single space of symmetric trapezoidal fuzzy numbers [16, 17], and some heuristic methods have been successfully used for feature level fusion [17]. In this dissertation, we are interested in decision level fusion. Thus, the rest of this

chapter reviews existing work on decision level fusion.

Over the past few years, a variety of schemes have been proposed for combining multiple classifiers. The most representative approaches include majority vote [18], Borda count [6], average [19], weighted average [20], Bayesian [21], probabilistic [22], polling methods [23], logistic regression [6], and combination by neural networks [24], and hierarchical mixture of experts [25]. Most of the above approaches assume that the classifier decisions are independent. For instance, the Bayesian approach requires this independence assumption in order to compute the joint probabilities. However, in practice, the outputs of multiple classifiers are usually highly correlated. Therefore, in addition to assigning fusion weights to the individual classifiers, it is desirable to assign weights to subsets of classifiers to take into account the interaction between them. Fusion methods based on the fuzzy integral [26, 27] and Dempster-Shafer theory [28] have this desirable property.

Methods for combining multiple classifiers can be classified into two main categories: *classifier selection* and *classifier fusion.* Classifier selection methods put an emphasis on the development of the classifier structure. This approach assumes that the classifiers are complementary, and that their expertise vary according to the different areas of the feature space. For a given test sample, these methods attempt to predict which classifiers are more likely to be correct. Some of these methods consider the output of only a single classifier to make the final decision [8]. Others, combine the output of multiple "local expert" classifiers [3]. Classifier fusion methods, on the other hand, operate mainly on the classifiers outputs, and strive to combine the classifiers outputs effectively. This approach assumes that the classifiers are competitive and equally experienced over the entire feature space. For a given test sample, the individual classifiers are applied in parallel, and their outputs are combined in some manner to take a group decision.

A diagrammatic representation of classifier fusion methods is shown in Figure 1 [1]. From the three possible types of outputs generated by individual classifiers the crisp labels

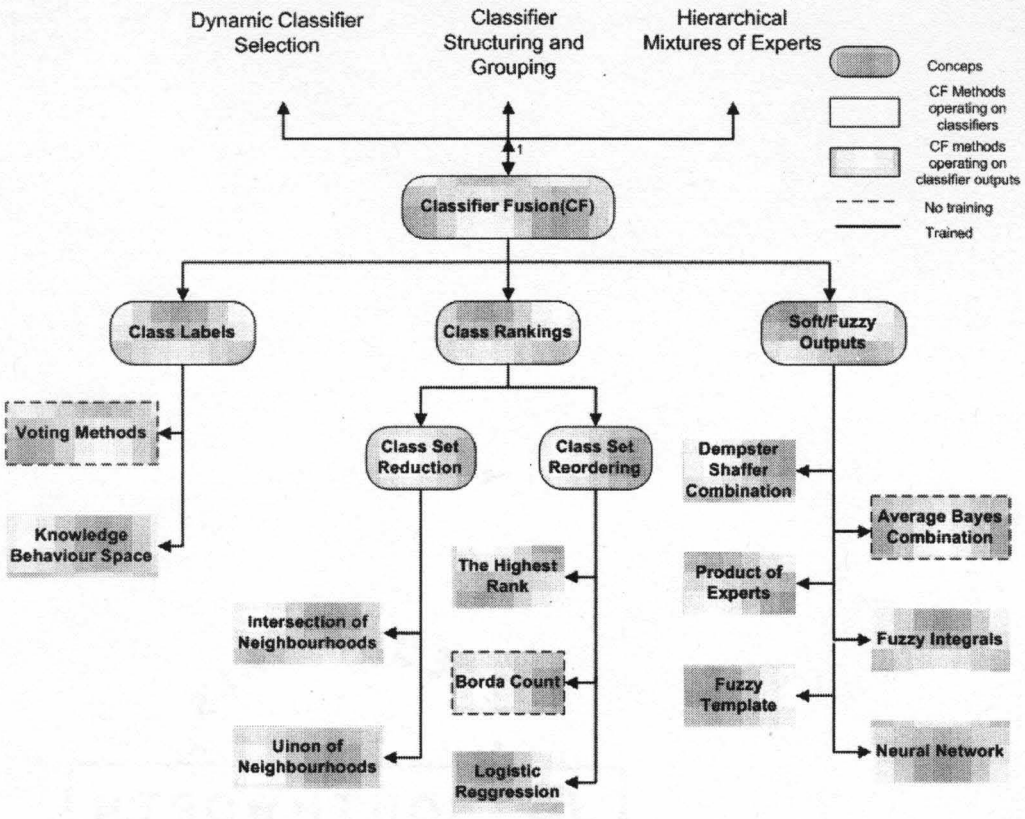


Figure 1. Categorization of different classifier fusion methods [1]

offer the minimum amount of input information for fusion methods, as no information about potential alternatives is available. Some additional useful information can be gained from classification methods generating outputs in the form of class rankings. However, fusion methods operating on classifiers with soft/fuzzy outputs can be expected to produce the greatest improvement in classification performance.

Another way to categorize classifier combination methods is based on the way they select or assign weights to the individual classifiers. Some methods are *global* and assign a degree of worthiness, that is averaged over the entire training data, to each classifier. Other methods are *local* and adapt the classifiers' worthiness to different data subspaces. Intuitively, the use of data-dependent weights, when learned properly, provides higher classification accuracy. This approach requires partitioning the input samples into regions during

the training phase. The partition can be defined from the space of individual classifier decisions [29], according to which classifiers agree with each other [6], or by features of the input space [7]. Then, the best classifier for each region is identified and is designated as the expert for this region [2]. Conversely, the partitioning can be defined such that each classifier is an expert in one region [8]. This approach may be more efficient, however, its implementation is not trivial. In the classification phase, the region of an unknown sample is identified, and the output of the classifier responsible for this region is used to make the final decision. Data partition and classifier selection could also be made dynamic during the testing phase [9, 10]. In this case, the accuracy of each classifier (with respect to the training samples) is estimated in local regions of the feature space in the vicinity of the test sample. The most accurate classifier is selected to classify the test sample.

In the following, we first outline the combining rules for multiple source or classifiers. Then, we outline several classifier fusion methods that are related to our work. These methods include Bayesian Fusion, Dempster-Shafer Theory (DST), supervised Borda-Count, Decision Template, Boosting (AdaBoost), and Random Forest.

A Combining probabilistic information

Let $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_L\}$ be a set of classifiers and let $\Omega = \{\Omega_1, \cdots, \Omega_C\}$ denote a set of class labels. Each algorithm, $\mathcal{D}_i$, extracts a set of features, x_i, and assigns a confidence value y_i to each of the C classes, i.e., $\mathcal{D}_i(x_i) = y_k$, for $i = 1, \cdots, L$, and $k = 1, \cdots, C$.

For classifier fusion, we need to compute the global posterior probability distribution $p(\Omega_k|\mathcal{D}_1(x_1), \mathcal{D}_2(x_2), \cdots, \mathcal{D}_L(x_L)), k = 1, \cdots, C$, given the information contributed by each source. This probability could be computed using a linear or non-linear methods as follows.

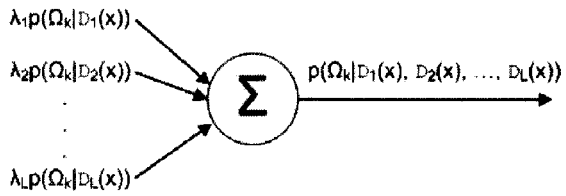


Figure 2. Linear Opinion Pool

1 Linear opinion pool

In tackling the problem of fusion, the questions of how relevant and how reliable is the information from each source should be considered. These questions can be addressed by attaching a measure of value, e.g. a weight, to the information provided by each source. Such a pool, based on the probabilistic representation of the information, was proposed by Stone [30]. The posteriors from each information source are combined linearly (see Figure 2), i.e.

$$P(\Omega_k|\mathcal{D}_1(x_1), \mathcal{D}_2(x_2), \cdots, \mathcal{D}_L(x_L))) = \sum_{i=1}^{L} \lambda_i p(\Omega_k|\mathcal{D}_i(x_i)) \tag{1}$$

where λ_i is a weight such that, $0 \leq \lambda_i \leq 1$ and $\sum_{i=1}^{L} \lambda_i = 1$. The weight λ_i reflects the significance attached to the i^{th} information source. It can be used to model the reliability or trustworthiness of an information source and to "weight out" faulty sensors.

In the case of equal weights, the Linear Opinion Pool can give an erroneous result if one sensor is dissenting even if L is relatively large. This is because the Linear Opinion Pool gives undue credence to the opinion of the i^{th} source. The need to redress this leads to the second approach.

2 Independent opinion pool

In the Independent Opinion Pool [31], it is assumed that the information obtained conditioned on the observation set is independent. More precisely, the Independent Opinion

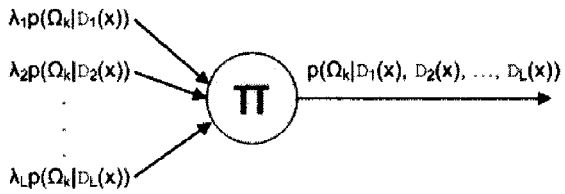


Figure 3. Independent Opinion Pool

Pool illustrated in Figure 3 is defined by the product:

$$P(\Omega_k|\mathcal{D}_1(x_1), \mathcal{D}_2(x_2), \cdots, \mathcal{D}_L(x_L))) \propto \Pi_{i=1}^{L} p(\Omega_k|\mathcal{D}_i(x_i)) \tag{2}$$

In general, this is a difficult condition to satisfy, though in the realm of measurement the conditional independence can often be justified experimentally.

B Bayesian Fusion

Bayesian fusion [32] is based on Bayesian decision theory which is a fundamental statistical approach to the problem of pattern classification. This approach is based on quantifying the tradeoffs between various classification decisions using probability and the costs that accompany such decisions. Bayesian data fusion has been studied extensively in the literature (e.g. [32, 33]). This approach has the advantage of being able to incorporate *a priori* knowledge about the likelihood of the hypothesis being tested, and when empirical data are not available, it is possible to use subjective estimates of the prior probabilities. Moreover, from a statistical point of view, the use of Bayes rule should provide the optimal decision. Unfortunately, the proper use of Bayes requires the joint probability density functions to be known. This information is usually not available and may be difficult to estimate from the data. Other disadvantages of the Bayesian approach include complexities when dealing with multiple potential hypotheses and multiple conditionally dependent events, and the inability to account for general uncertainty [33]. Thus, Bayesian fusion is best suited to applications where prior parameters are available, there is no need

to represent ignorance, and where conditional dependency can be easily modeled through probabilistic representation.

Bayesian fusion has been applied to target identification [34], image analysis [35], and many other applications [32]. It has also been applied to the problem of anti-personnel landmine detection [36, 37], and the results were compared to other fusion methods. In [36], only synthetic data were used, and in [37] a very small data set was used. Thus, the results were not conclusive.

In conventional statistical pattern recognition methods, features are extracted from objects. The features are expressed in a form of feature vectors, and the probability density function of feature vectors is estimated for each category. An unknown input pattern is assigned to the category with the maximum probability [38]. In parametric density estimation, the forms for the density function is assumed to be known, and parameters of the function are estimated using the training sample vectors.

Let $\mathbf{v}$ represent the output of all L algorithms to be fused, i.e., $\mathbf{v} = [y_1, y_2, \cdots, y_L]$. Within the Bayesian framework, $\mathbf{v}$ is considered a random variable with a distribution that depends on the state of nature. Using Bayes formula, we first compute the posterior probability using

$$p(\Omega_i|\mathbf{v}) = \frac{p(\mathbf{v}|\Omega_i)p(\Omega_i)}{p(\mathbf{v})}. \tag{3}$$

Then, $\mathbf{v}$ is assigned to the class with maximum posterior probability, i.e.,

$$\mathbf{v} \in \Omega_j \ \ if \ \ p(\Omega_j|\mathbf{v}) = \max_{i=1..K} p(\Omega_i|\mathbf{v}) \tag{4}$$

In (3), $p(\Omega_i)$ is the prior probability of class i and $p(\mathbf{v}|\Omega_i)$ is the class conditional density. The prior $p(\Omega_i)$ is usually provided by an expert, or estimated using the relative proportions of training data from each class. Similarly, $p(\mathbf{v}|\Omega_i)$ can be estimated from the training data.

The Gaussian distribution is usually used as the density function. This is because the Gaussian distribution is easy to handle, and in many cases, the distribution of the sample

vectors can be regarded as normal if there are enough samples. The mean vector and covariance matrix are calculated from the vectors.

Let d be the dimension of feature vector. The probability density function of a d-dimensional normal distribution is given by:

$$p(\mathbf{v}|\Omega_\mathbf{i}) = \frac{\mathbf{1}}{(\mathbf{2}\pi)^{\frac{\mathbf{d}}{\mathbf{2}}}|\sum_\mathbf{i}|^{\frac{\mathbf{1}}{\mathbf{2}}}}\mathbf{e}^{-\frac{\mathbf{1}}{\mathbf{2}}(\mathbf{v}-\mu_\mathbf{i})^\mathbf{T}\sum_\mathbf{i}^{-\mathbf{1}}(\mathbf{v}-\mu_\mathbf{i})} \tag{5}$$

where v is a d-component vector, μ_i is the mean vector for class i, and $\sum_i$ is the $d \times d$ covariance matrix for class i. Then, the posterior probability $P(\Omega_i|\mathbf{v})$ can be computed by Bayes formula using Eq. (3) as:

$$\begin{aligned} P(\Omega_k|\mathbf{v}) &= \frac{p(\mathbf{v}|\mathbf{\Omega_k})\mathbf{P}(\mathbf{\Omega_k})}{p(\mathbf{v})} \\ &= \frac{P(\Omega_k)e^{-\frac{1}{2}(\mathbf{v}-\mu_\mathbf{k})^\mathbf{T}\sum_\mathbf{k}^{-\mathbf{1}}(\mathbf{v}-\mu_\mathbf{k})}}{(2\pi)^{d/2}|\sum_k|^{1/2}p(\mathbf{v})} \end{aligned} \tag{6}$$

If the training data can be modeled by a mixture of Gaussian distributions, the Expectation Maximization (EM) algorithm [39] can be used first to build the multiple Gaussian model. Then this modeling can be used to make the final decision according to the above Bayes rule. The EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. Each iteration of the EM algorithm consists of two processes: The E-step, and the M-step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E-step are used in lieu of the actual missing data. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration. For the fusion problem, we can first cluster the training data using the EM into M components. Then, the posterior probability can be computed by generalizing Bayes rule in Eq. (7) to assign a test point

into different component or class.

$$P(\Omega_k|\mathbf{v}) = \sum_{i=1}^{M} \frac{p(\mathbf{v}|\Omega_{\mathbf{ki}})\mathbf{P}(\Omega_{\mathbf{ki}})}{p(\mathbf{v})} = \sum_{i=1}^{M} \frac{P(\Omega_{ki}) e^{-\frac{1}{2}(\mathbf{v}-\mu_{\mathbf{ki}})^{\mathbf{T}} \sum_{\mathbf{ki}}^{-\mathbf{1}} (\mathbf{v}-\mu_{\mathbf{ki}})}}{(2\pi)^{d/2} |\sum_k|^{1/2} p(\mathbf{v})} \quad (7)$$

C Dempster-Shafer Fusion

Dempster-Shafer Theory (DST) is a mathematical theory of evidence for representing uncertain knowledge [40, 41]. In a finite discrete space, Dempster-Shafer theory can be interpreted as a generalization of probability theory where probabilities are assigned to sets as opposed to mutually exclusive singletons. In traditional probability theory, evidence is associated with only one possible event. In DST, evidence can be associated with multiple possible events, e.g., sets of events. As a result, evidence in DST can be meaningful at a higher level of abstraction without having to resort to assumptions about the events within the evidential set. The Dempster-Shafer model collapses to the traditional probabilistic formulation when the evidence is sufficient enough to permit the assignment of probabilities to single events.

DST fusion was applied to handwriting recognition [42], decision making [43], face detection [44], landmine detection [36, 37, 45], and more [32, 46]. One important feature of DST is its ability to cope with varying levels of precision regarding the information with no further assumptions needed to represent the information. It also allows for direct representation of uncertainty of system responses where an imprecise input can be characterized by a set or an interval and the resulting output is a set or an interval. However, DST fails to give an acceptable solution to fusion problems with significant conflict [47, 48]. Consequently, many researchers developed modified Dempster rules to represent the degree of conflict [46].

DST and Bayesian theories have been studied and compared extensively [49, 33,

50]. Both theories have initial requirements. DST theory requires masses to be assigned to alternatives in a meaningful way, including the unknown state; whereas Bayes theory requires prior probabilities. In general, the results of both methods may be comparable, but the implementations may require different amounts of effort and information. Thus, selecting one approach over the other usually depends on the extent to which prior information is available.

Let $\Theta = \{\theta_1, \cdots, \theta_K\}$ be a finite set of possible hypotheses. This set is referred to as the frame of discernment, and its power set is denoted by 2^θ. There are three important functions in Dempster-Shafer theory: the *basic belief assignment function* (BBA or m), the *Belief function* (*Bel*), and the *Plausibility function* (*Pl*).

1 Basic belief assignment Function

The basic belief assignment (BBA) function is a primitive of evidence theory. Generally speaking, the term basic belief assignment does not refer to probability in the classical sense. The basic belief assignment m assigns a value in [0, 1] to every subset $\mathcal{A}$ of Θ and satisfies the following condition:

$$m(\phi) = 0, and \sum_{\mathcal{A} \subseteq \Theta} m(\mathcal{A}) = 1 \tag{8}$$

It is worth mentioning here that, $m(\phi)$ could be positive when considering unnormalized combination rule as will be explained later. While in probability theory a measure of probability is assigned to atomic hypotheses θ_i, in DST, $m(\mathcal{A})$ is the part of belief that supports $\mathcal{A}$, but does not support anything more specific, i.e., the value of $m(\mathcal{A})$ pertains only to the set $\mathcal{A}$ and makes no additional claims about any subsets of $\mathcal{A}$. Any further evidence on the subsets of $\mathcal{A}$ would be represented by another BBA, i.e. $\mathcal{B} \subset \mathcal{A}$, $m(\mathcal{B})$ would be the BBA for the subset $\mathcal{B}$. For $\mathcal{A} \neq \theta_i$, $m(\mathcal{A})$ reflects some ignorance because it is a belief that cannot subdivide into finer subsets. $m(\mathcal{A})$ is a measure of support that will be assigned to a composite hypothesis $\mathcal{A}$ at the expense of support $m(\theta_i)$ of atomic hypotheses θ_i. A subset

$\mathcal{A}$ for which $m(\mathcal{A}) > 0$ is called a *focal element.* The partial ignorant associated with $\mathcal{A}$ leads to the following inequality: $m(\mathcal{A}) + m(\overline{\mathcal{A}}) \leq 1$, where $\overline{\mathcal{A}}$ is the compliment of $\mathcal{A}$. In other words, the Dempster-Shafer theory of evidence allows to represent only our actual knowledge without being forced to overcommit when it is ignorant.

2 Belief function

Intuitively, a portion of belief committed to a hypothesis $\mathcal{A}$ must also be committed to any hypothesis it implies. To obtain the total belief in $\mathcal{A}$, one must therefore add to $m(\mathcal{A})$ the quantities $m(\mathcal{B})$ for all subsets $\mathcal{B}$ of $\mathcal{A}$. Therefore, the belief function, *Bel*(.), associated with the BBA *m*(.) assigns a value in [0, 1] to every nonempty subset $\mathcal{A}$ of Θ. It is called "degree of belief in $\mathcal{A}$" and is defined by:

$$Bel(\mathcal{A}) = \sum_{\mathcal{B} \subseteq \mathcal{A}} m(\mathcal{B}) \tag{9}$$

$Bel(\mathcal{A})$, also called the *credibility* of $\mathcal{A}$, is interpreted as a measure of the total belief committed to $\mathcal{A}$. We can consider a basic belief assignment as a generalization of a probability density function whereas a belief function is a generalization of a probability function. It can be easily verified that the belief in some hypothesis $\mathcal{A}$ and the belief in its negation $\overline{\mathcal{A}}$ do not necessarily sum to 1, which is a major difference with probability theory.

3 Plausibility function

Plausibility function (Pl) is the sum of all the basic belief assignments of the sets ($\mathcal{B}$) that intersect the set of interest ($\mathcal{A}$) ($\mathcal{B} \cap \mathcal{A} \neq \phi$) [51]. Formally, for all sets A that are elements of the power set ($\mathcal{A} \in \Theta$).

$$Pl(\mathcal{A}) = \sum_{\mathcal{B} \cap \mathcal{A} \neq \phi} m(\mathcal{B}) \tag{10}$$

The two measures, Belief and Plausibility measures are nonadditive. This can be interpreted as it is not required for the sum of all the Belief measures to be 1 and similarly for

the sum of the Plausibility measures. It is possible to obtain the basic belief assignment from the Belief measure with the following inverse function:

$$m(\mathcal{A}) = \sum_{\mathcal{B} \subseteq \mathcal{A}} (-1)^{|\mathcal{A}-\mathcal{B}|} Bel(\mathcal{B}) \tag{11}$$

In Eq. (11), $|\mathcal{A} - \mathcal{B}|$ is the difference of the cardinality of the two sets.

In addition to deriving these measures from the basic belief assignment (m), these two measures can be derived from each other in the following way:

$$\begin{aligned} &Pl(\mathcal{A}) = 1 - Bel(\overline{\mathcal{A}}) \\ &Bel(\overline{\mathcal{A}}) = \textstyle\sum_{\mathcal{B} \subseteq \overline{\mathcal{A}}} m(\mathcal{B}) = \sum_{\mathcal{B} \cap \mathcal{A} = \phi} m(\mathcal{B}) \\ &\textstyle\sum_{\mathcal{B} \cap \mathcal{A} \neq \phi} m(\mathcal{B}) = 1 - \sum_{\mathcal{B} \cap \mathcal{A} = \phi} m(\mathcal{B}) \end{aligned} \tag{12}$$

In Eq. (12), $\overline{A}$ is the classical complement of $\mathcal{A}$. This definition of Plausibility in terms of Belief comes from the fact that all basic assignments must sum to 1.

A BBA can also be viewed as determining a set of probability distributions $\mathcal{P}$ over 2^{θ} satisfying:

$$Bel(\mathcal{A}) \leq P(\mathcal{A}) \leq Pl(\mathcal{A}), \ \forall \ \mathcal{A} \subseteq \Theta \tag{13}$$

For this reason, *Bel* and *Pl* are also called lower and upper probabilities, respectively. This fundamental imprecision in the determination of the probabilities reflects the "weakness", or the incompleteness of the available information. The above inequalities reduce to equalities in the case of a Bayesian belief function.

4 Combination rule

Consider two BBAs $m_1(.)$ and $m_2(.)$ for belief functions $bel_1(.)$ and $bel_2(.)$ respectively. Let $\mathcal{A}_j$ and $\mathcal{B}_k$ be focal elements of bel_1 and bel_2 respectively. Then $m_1(.)$ and $m_2(.)$ can be combined to obtain the belief mass committed to $\mathcal{C} \subset \Theta$ according to the following

combination or orthogonal sum formula [40],

$$m(\mathcal{C}) = m_1 \oplus m_2(\mathcal{C}) = \frac{\sum\limits_{j,k,\mathcal{A}_j \cap \mathcal{B}_k = \mathcal{C}} m_1(\mathcal{A}_j) m_2(\mathcal{B}_k)}{1 - \sum\limits_{j,k,\mathcal{A}_j \cap \mathcal{B}_k = \phi} m_1(\mathcal{A}_j) m_2(\mathcal{B}_k)}, \mathcal{C} \neq \phi \tag{14}$$

The denominator is a normalizing factor, which intuitively measures how much $m_1(.)$ and $m_2(.)$ are conflicting.

5 Weighted Combination Rule for DST

The measures of Belief and Plausibility are derived from the combined basic assignments. Dempsters rule combines multiple belief functions through their basic probability assignments. These belief functions are defined on the same frame of discernment, but are based on independent arguments or bodies of evidence. The issue of independence is a critical factor when combining evidence and is an important research subject in DST. The denominator in Dempsters rule has the effect of completely ignoring conflict and attributing any probability mass associated with conflict to the null set [47]. Consequently, this operation will yield counterintuitive results in the face of significant conflict in certain contexts.

If we have prior knowledge about reliability of the sources, we can discount the source and assign them weights before combining their belief functions, resulting in a weighted Dempster-Shafer fusion rule:

$$m(\mathcal{C}) = m_1 \oplus m_2(\mathcal{C}) = \frac{\sum\limits_{j,k,\mathcal{A}_j \cap \mathcal{B}_k = \mathcal{C}} w_1 m_1(\mathcal{A}_j) w_2 m_2(\mathcal{B}_k)}{1 - \sum\limits_{j,k,\mathcal{A}_j \cap \mathcal{B}_k = \phi} w_1 m_1(\mathcal{A}_j) w2 m_2(\mathcal{B}_k)}, \mathcal{C} \neq \phi \tag{15}$$

6 Combining several belief functions

The combination rule can be easily extended to several belief functions by repeating the rule for new belief functions. Thus the pairwise orthogonal sum of n belief functions

$bel_1, bel_2, \cdots, bel_n$, can be formed as the belief function :

$$((Bel_1 \oplus Bel_2) \oplus Bel_3) \cdots \oplus Bel_n = \bigoplus_{i=1}^{n} Bel_i \tag{16}$$

D Borda Count Fusion

In 1770, J.C. de Borda presented a new method of election to the French Royal Academy of Sciences [52]. His method involved having each voter rank all the candidates in an election. These ranks would be combined by summing, and the candidate with the best rank sum would be the winner. Soon after, the Marquis de Condorcet [53] presented an alternate method of using pairwise comparisons to generate ranked election results. Black [54], Arrow [55], and others [6] have analyzed the Borda, Condorcet, and other such methods for making communal ranking decisions. Each such ranking process involves a set of candidates and set of voters. The voters supply a schedule indicating their rankings (either total or pairwise) of the candidates, i.e. voters rank candidates in order of preference. The Borda count determines the winner of an election by giving each candidate a certain number of points corresponding to the position in which he or she is ranked by each voter. Once all votes have been counted the candidate with the most points is the winner.

The Borda count has been used for fusing the results of classifiers for the task of handwriting recognition [6, 27, 56]. In this setting, there are C classifiers and N classes. The classes correspond to words in a lexicon. Each classifier assigns a ranking of classes (possibly partial) to each object (a handwritten word). Ho, et al. [6], presented a weighted Borda count technique for this application that uses logistic regression to identify classifier weights by comparing the ranking results of each classifier with the best ranking derived by applying several different independent classification algorithms. Gader, et al. [27], employ a method in which the Borda weights are determined dynamically based on a match confidence between the object and a lexicon string. Van Erp and Schomaker [56] compared the performance of the Borda count, a variant of the Borda count, in which the median rank

(rather than sum or average) is used, and Nansons [57] election procedure (an iterative Borda scheme that deletes the candidate ranked lowest in each successive iteration).

1 General Approach

One approach [12] to implement fusion using rank weighings is to consider each discrimination algorithm to be a voter, and each observation in the training set to be a candidate. Formally, given a set of algorithms $\mathcal{D}_1, \cdots \mathcal{D}_L$ and a set of training sample objects $x_1, \cdots, x_N$, each algorithm maps each object x_j to a confidence values y_j. Algorithm $\mathcal{D}_i$ assigns rank $r_i(y_{ij})$ to candidate j if $y_{ij} = \mathcal{D}_i(x_j)$ has a confidence value greater than exactly $r_i(y_{ij}) - 1$ other candidate alarms. Thus, r_i is a map from the confidence values assigned by algorithm $\mathcal{D}_i$ into the set $\{1, \cdots, N\}$. Then, for a new candidate x_j^* with $y_i{}^* = \mathcal{D}_i(x_j^*)$, the rank can be computed using:

$$\hat{r}_i(y_j{}^*) = r_i(\bigvee_{y_{ij} \leq y_j{}^*} y_{ij}) \tag{17}$$

The rank, $\hat{r}(y_j{}^*)$ is the number of candidates in the training set having confidence value no greater than $(y_j{}^*)$.

The unweighted Borda count fusion of all L algorithms on candidate $x_j{}^*$ can be computed as $R(x_j{}^*)$ using:

$$R(x_j{}^*) = \frac{1}{LN} \sum_{i=1}^{L} \hat{r}_i(y_j{}^*) \tag{18}$$

Note that this result is normalized to yield a value in the range [0,1].

2 Weighted Borda Count

If there is evidence that algorithms $\mathcal{D}_i$ and $\mathcal{D}_j$ have differing predictive capacities, say $\mathcal{D}_i$ is more likely to be correct than $\mathcal{D}_j$, then it makes sense to assign weights w_i and w_j to these algorithms, where $w_i > w_j$. In general, a weighted Borda scheme assigns to each

algorithm $\mathcal{D}_m$, a weight w_m satisfying:

$$\sum_{m=1}^{L} w_m = 1 \tag{19}$$

Then, the weighted Borda Count assigns confidence $R(x_j{}^*)$ to new candidate alarm $x_j{}^*$ as follows:

$$R(o^*{}_j) = \frac{1}{LN} \sum_{m=1}^{L} w_m \hat{r_i}(y_{ij}{}^*) \tag{20}$$

The algorithm weight, w_m, could be assigned using prior knowledge. They could also be learned using a set of labels samples by optimizing some criteria [58].

E Decision Template Method

Decision Template (DT) [7] is a robust classifier fusion scheme that combines classifier outputs by comparing them to a characteristic template for each class. DT fusion uses all classifier outputs to calculate the final support for each class, which is in sharp contrast to most other fusion methods which use only the support for that particular class to make their decision.

In many cases, the classifier output is a C-dimensional vector with support to the C classes, i.e.,

$$\mathcal{D}_i(x_i) = [d_{i,1}(x_1), \cdots, d_{i,C}(x_C)]^T, \; i = 1, \cdots, L \tag{21}$$

where x_i is a set of feature for classifier $\mathcal{D}_i$, $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_L\}$ is a set of classifiers and $\Omega = \{\Omega_1, \cdots, \Omega_C\}$ is a set of class labels. Without loss of generality, $d_{i,j}(x_i)$ can be restricted to the interval [0,1], $i = 1, \cdots, L, j = 1, \cdots, C$. $d_{i,j}(x_i)$ is the degree of support given by classifier i to the hypothesis that x_i comes from class Ω_j (most often an estimate of the posterior probability $P(w_j|x_i)$). Combining classifiers means to find a class label for x based on the $\mathcal{D}$ classifier outputs $\mathcal{D}_1(x_1), \cdots, \mathcal{D}_L(x_L)$, i.e.:

$$\hat{\mathcal{D}}(x) = \mathcal{F}(\mathcal{D}_1(x_1), \cdots, \mathcal{D}_L(x_L)) \tag{22}$$

1 General Model for DT Classifier Fusion

DT treats the classifier outputs as the input to a second-level classifier in some intermediate feature space, and designs a new classifier for the second (combination) level. The classifier outputs can be organized in a *decision profile* (DP) [59] as a matrix:

$$\mathcal{DP}(x) = \begin{bmatrix} d_{1,1}(x) & \cdots & d_{1,j}(x) & \cdots & d_{1,j}(x) \\ \cdots & & & & \\ d_{i,1}(x) & \cdots & d_{i,j}(x) & \cdots & d_{i,j}(x) \\ \cdots & & & & \\ d_{L,1}(x) & \cdots & d_{L,j}(x) & \cdots & d_{L,j}(x) \end{bmatrix} \tag{23}$$

The entries in $\mathcal{DP}(x)$ form the *intermediate feature space*. The DT approach builds a minimum-error classifier by replacing the problem of estimating $P(\Omega_i|x)$ with one of estimating $P(\Omega_i|\mathcal{D}_1(x_1), \cdots, \mathcal{D}_L(x_L))$, or more compactly, $P(\Omega_i|\mathcal{DP}(x))$. Thus, the initial feature space with n features, $\mathbb{R}^n$, is transformed into a new space with $L \times c$ features. This treatment of the combination problem underpins the schemes in [6, 60, 61]. In a way, this idea is akin to support vector machines approach where the initial feature space is transformed in a new (generally higher dimensional) space and the classifier is built in that new space [62]. However, in the model here, the intermediate feature space has a special context-related structure [59].

2 Decision Templates (DT)

Given L (trained) classifiers in $\mathcal{D}$, C Decision Templates (DT) are calculated from the data, one per class. The decision template for class Ω_i, denoted DT_i, is the centroid of class i in the intermediate feature space. DT_i can be regarded as the expected value for class Ω_i. The support for class Ω_i offered by the combination of the L classifiers,$\mu_i(x)$, is then found by measuring the similarity between the current $DP(x)$ and DT_i. $DP(x)$ and DT_i can be viewed as two fuzzy sets defined over the set of intermediate features and use

measures of similarity from fuzzy set theory. The following algorithm [7] describes the training and the testing procedures of the DT approach.

Decision Template(Training)

1.For $i = 1, \cdots, C$, calculate the mean of the decision profiles $DP(z_j)$ of all member of Ω_i from the data set Z. Call the mean a decision template DT_i:

$$\mathcal{DT}_i = \frac{1}{N_i} \sum_{\substack{z_j \in w_i \\ z_j \in Z}} DP(z_j), \tag{24}$$

where N_i is the number of elements of Z from Ω_i;
2.Return $DT_1, \cdots, DT_C$.

Decision Template(Testing)

1.Given the input $x \in \Re^n$ construct $DP(x)$ as in Eq. (23);
2.Calculate the distance between $DP(x)$ and each DT_i, $i = 1, \cdots, c$,

$$d(\mathcal{DP}(x), \mathcal{DT}_i) = \sum_{j=1}^{c} \sum_{k=1}^{L} (d_{k,j}(x) - dt_i(k,j))^2 \tag{25}$$

where $dt_i(k,j)$ is the k, j^{th} entry in decision template;
3.Calculate the components of the soft label of x by:

$$\mu_i(x) = 1 - \frac{1}{L \cdot c} d_E(DP(x), DT_i) \tag{26}$$

If the classifier outputs are some estimates of the posterior probabilities $P(\Omega_k|x)$, $k = 1, \cdots, C$, the decision template is an unbiased estimate of the expectation of the $L \times c$ dimensional random variable $DP(x)$ given that the true class is Ω_i. Therefore, assessing the similarity between the actually occurred matrix of outputs $DP(x)$ and the expected one for Ω_i is a reasonable classification strategy.

Figure 4 illustrates how the DT scheme operates. The decision templates are calculated in advance using Eq. (24).

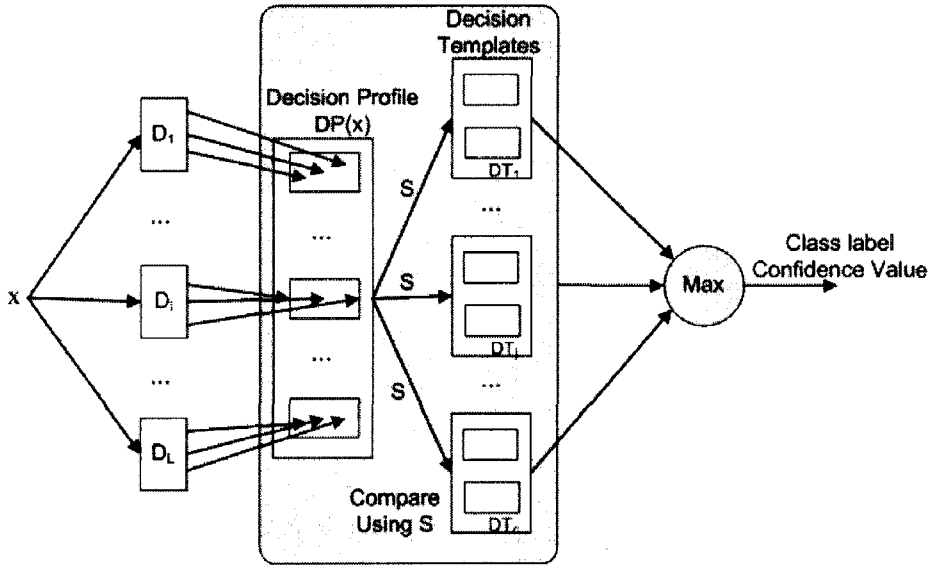


Figure 4. Architecture of the decision templates classifier fusion scheme

F Boosting

Boosting is an iterative procedure used to adaptively change the distribution of the training examples so that the base classifiers will focus on examples that are hard to classify. Boosting assigns a weight to each training sample and may adaptively change the weight at the end of the boosting round. Examples that are classified incorrectly will have their weights increased, while those that are classified correctly will have their weights decreased. This forces the classifier to focus on examples that are difficult to classify in subsequent iterations.

Over the years, several implementations of boosting have been developed [63, 5]. These algorithms differ in terms of (1) how the weights of the training example are updated at the end of each boosting round, and (2) how the predictions made by each classifier are combined. The original ones, proposed by Robert Schapire (a recursive majority gate formulation [5]) and Yoav Freund (boost by majority [63, 64]) were not adaptive and could not take full advantage of the weak learners.

While boosting is not algorithmically constrained, most boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are typically weighted in some way that is usually related to the weak learner's accuracy. After a weak learner is added, the data is reweighed: examples that are misclassified gain weight and examples that are classified correctly loose weight (some boosting algorithms actually decrease the weight of repeatedly misclassified examples, e.g., boost by majority). Thus, future weak learners focus more on the examples that previous weak learners misclassified.

AdaBoost is one of the most commonly used Boosting algorithms [65], and is summarized as follows.

AdaBoost Algorithm

1.$w = \{w_j = 1/N | j = 1, 2, , N\}$. Initialize the weights for all N examples;
2.Let k be the number of booting samples;
3.for $i = 1$ to k do,
4. Created a training set $\mathcal{S}_i$ by sampling (replacement) from $\mathcal{S}$ according to w;
5. Train a base classifier $\mathcal{D}_i$ on the bootstrap sample $\mathcal{S}_i$;
6. Apply $\mathcal{D}_i$ to all examples in the original training set $\mathcal{D}$;
7.

$$\varepsilon_i = [\sum_j w_j \delta(\mathcal{D}_i(x_j) \neq y_j)]/N \tag{27}$$

8. if $\varepsilon_i > 0.5$ then
9. $w = \{w_j = 1/N | j = 1, 2, , N\}$;
10. Go back to Step 4.
11. end if
12.

$$\alpha_i = \frac{1}{2} * \ln \frac{1 - \alpha_i}{\alpha_i} \tag{28}$$

13. Update the weight of each example according to

$$w_i^{j+1} = \frac{w_i^j}{Z_j} \times \begin{cases} \exp^{-\alpha_j} & \text{if } \mathcal{D}_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } \mathcal{D}_j(x_i) \neq y_i \end{cases} \tag{29}$$

14. end for
15.

$$\mathcal{D}^*(x) = argmax_y \sum_{j=1}^{T} \alpha_j \delta(\mathcal{D}_j(x) = y) \tag{30}$$

$\delta(.) = 1$ if its argument is true and 0 otherwise.

G Random Forest

Random Forest is a tree-based ensemble classifier that uses bagging technique to create new training sets [4]. It includes two important methods: random feature subspace

and out-of-bag estimates. The former enables a much faster construction of trees and the latter the possibility of evaluating the relative importance of each input feature. The general Random Forest algorithm is summarized as follows.

Random Forests

1. Choose T number of trees to grow;

2. Choose m number of variables used to split each node. $m \ll M$, where M is the number of input variables. m is hold constant while growing the forest;

3. Grow T trees. When growing each tree do the following.

   (a) Construct a bootstrap sample of size n sampled from S_n with replacement and grow a tree from this bootstrap sample;

   (b) When growing a tree at each node select m variables at random and use them to find the best split;

   (c) Grow the tree to a maximal extent. There is no pruning;

4. To classify point X collect votes from every tree in the forest and then use majority voting to decide on the class label.

CHAPTER III

BACKGROUND ON LANDMINE DETECTION

"Near the start of last century, 90 percent of wartime casualties were soldiers. As the century waned, 90 percent were civilians" [13]. That stunning statistic is not attributable to the landmine crisis alone. But anti-personnel(AP) landmines have added greatly to the devastating impact of modern conflict on noncombatants. These hidden killers are cheap to buy, easy to use, hard to detect and difficult to remove. They cost as little as $3 to produce, but as much as $1000 to remove. The simple fact is that more landmines are deployed in armed conflict every year than are removed by mine clearance personnel. The world is now littered with an estimated 80-110 million landmines in 64 countries, which maim or kill an estimated 500 people every week, mostly innocent civilians.The majority of these mines were deployed during the last 15 years. The burden imposed by the proliferation and indiscriminate use of these weapons is beyond calculation [13, 66]. Thus, detection and removal of landmines is a serious problem affecting civilians and soldiers worldwide.

Since the 1930s, many countries have worked on the solution to the problem of detecting nonmetallic landmines. The research has encompassed an extremely wide range of technologies and hundreds of millions of dollars have been spent. Despite these efforts, there is still no satisfactory operational detection solution. This lack of success is attributable to the extreme difficulty of the problem, such as: the large variety of landmine types, differing soil type and compaction, temperature, moisture, shadow, time of day, weather conditions, and varying terrain, to name a few.

Among these researches, a variety of sensors have been proposed or are under in-

vestigation for landmine detection. It is necessary to have a very high detection rate with a low false alarm rate. The research problem for sensor data analysis is to determine how well signatures of landmines can be characterized and distinguished from other objects under the ground using returns from one or more sensors. Metal detectors are used most frequently, unfortunately, many modern landmines are made of plastic and contain little or no metal. Ground Penetrating Radar (GPR) offers the promise of detecting landmines with little or no metal content. Unfortunately, landmine detection via GPR has been a difficult problem [67, 68]. Although systems can achieve high detection rates, they have done so at the expense of high false alarm rates. The detection problem is compounded by the large variety of explosive object types, differing soil conditions, temperature, weather conditions, and varying terrain. In particular, many systems can be significantly effected by rapidly changing environmental conditions. Therefore, detection algorithms which can adopt to changing conditions are needed for detecting buried landmines.

Because our proposed fusion method will be validated mainly by landmine detection problem, in this chapter we will give a brief overview of several sensors that have been used to detect landmines and introduce the principle, application and limitations among them to build a basic understanding of the landmine detection problem. A more detailed description can be found [69]. Additionally, we will outline few detection algorithms that will be fused by our proposed methods.

A Ground Penetrating Radar (GPR)

GPR works by emitting an electromagnetic wave covering a large frequency band into the ground through a wideband antenna. Reflections from the soil caused by dielectric variations such as the presence of an object are measured. By moving the antenna, it is possible to reconstruct an image representing a vertical slice of the soil (refer to Figure 6).

GPR is sensitive to discontinuities in the electrical properties of the interrogated

Figure 5. Wichmann/Niitek vehicle-mounted GPR at a western U.S. test site

medium, rather than to the presence of metal. Thus, GPR exploits a different phenomenology than EMI sensors (refer to Chapter III.C). Consequently, nonmetallic objects, such as wood, plastic, stone, as well as metallic objects, can be detected by a radar. Therefore, GPR offers the promise of detecting landmines with little or no metal content.

An example of a GPR system that has been developed to detect landmines include the Wichmann/Niitek GPR System [70]. This radar is a very-wide bandwidth (200Mhz - 7Ghz) bi-static GPR with very low radar cross-section that implicitly solves many of the problems typically associated with shallow-buried object detection utilizing ground penetrating radar technology.

This system, shown in Figure 5, consists of a vehicle-mounted wide-bandwidth impulse radar integrated with a marking and a GPS system. The radar is a 1.2 m wide and contains 24 antennae or channels, spaced approximately 5 cm apart. As it can be seen in Figure 5, the actual GPR is mounted some distance in front of a wheeled vehicle. As the vehicle moves in the down-track direction all 24 of the radars' channels are sampled once every 5 cm and at each down-track position each channel measures one 416-element

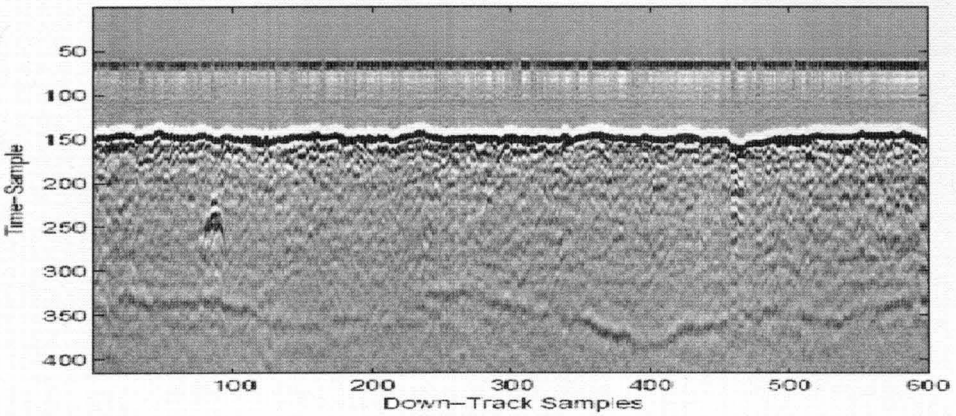


Figure 6. Sample of GPR responses. The x-axis represents down-track scan number, y-axis represents time sample. Two anomalies are visible in this data slice, one is at approximately sample 90, and another is near sample 460. Also note the high energy of ground bounce visible in all down-track scans near time sample 150. This data has been clipped to enhance contrast

time-domain vector, thus, yields a set of twenty-four 416-point time-domain vectors every 5 cm.

Sample unprocessed data from an U.S. site is shown in Figure 6. This image shows 600 down-track GPR responses from a central antenna channel. Clearly the largest source of GPR response energy is the dielectric discontinuity between the air and ground, seen near time sample 150 in all downtrack scans. Despite the ground response, one can still visually identify two subsurface anomalies at scans 90 and 460.

There are two types of GPR systems currently under investigation. One is the down-looking GPR system, which has its antennas placed near the surface of the earth. Though removing the strong signals reflected directly from the ground surface, referred to as the ground bounce removal, is a challenging problem [71, 72], this type of system shows a very promising detection capability. Its main drawback is that it is time consuming to use this type of system for large area interrogation, and short standoff distance is a problem as well. The other type of GPR is forward-looking system. This type has the GPR antennas

mounted on the front of a vehicle and captures radar signals at equally space positions as the vehicle moves forward, Synthetic Aperture Radar (SAR) images are formed from the received signals. The problem associated with this system, compared with the down-looking system, is that most of the radar transmitted energies are reflected off the targets and only a very small fraction can be received by the radar receiver. The deeper the burial depth of the mine, the weaker the returned signals. Moreover, due to their nearly identical dielectric coefficients to the surrounding soil, plastic mines cannot be seen convincingly in the spatial domain (SAR images) in the presence of clutter [73]. Hence, detecting buried plastic mines is extremely challenging for the forward-looking system.

B Metal Detectors (MD)

Some interesting studies have been carried out to see if it is feasible to discriminate mines from metallic clutter with metal detectors, to reduce the false alarm rate. For example, in [74], the author reported the results on using an impulse MD looking for a characteristic decay curve and comparing it to the ones stored in a library. Problems arise from the fact that the response curve depends on several factors, such as the orientation of the metallic object, and the exact metal type. Also, the matching is done only with objects that are known a priori. This approach could nevertheless be promising in specific situations. For instance in [75], the author studied the possibility of characterizing objects/mines by measuring the frequency response over a large frequency range.

Another interesting and unconventional application is represented by the Meandering Winding Magnetometer (MWM) described in [76]. The device has the characteristic of using a square wave winding conductor in order to generate a spatially periodic electromagnetic field, whose spatial wavelength depends only on the primary winding spatial periodicity. It can, in principle, detect several characteristics of a buried metallic object (size, shape, etc.), and its application to humanitarian demining is currently being investi-

gated.

The idea of using metal detectors to actually locate "cavities" in the soil, is also not new, as a (large) nonconducting target does indeed alter locally the natural ground conductivity, and has led for example to the patent ("cavity detector") described in [77]. The system should probably work best for large objects in soils with high natural conductivity ("background" signal).

C Electromagnetic Induction (EMI)

Another widely deployed metal detector (MD) for landmine detection is an electromagnetic induction (EMI) device that operates by sensing the metal present in land mines. The metal parts present in a landmine are detected by sensing the secondary magnetic field produced by eddy currents induced in the metal by a time-varying primary magnetic field. The frequency range employed is usually limited to a few tens of kHz. EMI sensors usually consist of a pair of coils, one of which is used to transmit either a broadband pulse or a continuous wideband electromagnetic waveform. The transmitted field induces a secondary current in the earth as well as in any buried conducting objects. In the case of pulsed excitation, the transmit waveform is quenched quickly and the receiving coil measures the decaying secondary field that has been induced in the earth and subsurface objects [78]. In the case of wideband excitation, the receiving coil is placed within the magnetic cavity so that it senses only the weak secondary field radiated by the earth and buried objects [79]. Present research is investigating replacement of the receive coil with magnetoresistive devices.

The most obvious and serious limitation of metal detectors used to detect landmines is the fact that they are metal detectors. A modern metal detector is very sensitive and can detect tiny metal fragments as small as a couple of millimeters in length and less than a gram in weight. An area to be demined is usually littered with a large number of such

metal fragments and other metallic debris of various sizes. This results in a high rate of nuisance alarms since a metal detector cannot currently distinguish between the metal in a landmine and that in a harmless fragment. The more sensitive a detector is, the higher the number of nuisance alarms it is likely to produce in a given location. Operating a detector at a lower sensitivity to reduce the number of such nuisance alarms may render it useless for detecting the very targets it was designed to detect, that is, the minimum metal-content landmines buried up to a few centimeters. Electromagnetic properties of certain soils can also limit the performance of metal detectors.

In addition to the above sensors, there exists several other promising techniques for landmine detection. Examples include Infrared Imaging (IR) [80, 81], neutron activation, X-ray backscatter [82], Nuclear Magnetic or Quadrupole Resonance (NMR/NQR) [83, 84, 85], and Thermal Neutron Activation (TNA) [86].

D Landmine Detection Data and Algorithms

Autonomous detection algorithms for landmine can generally be broken down into four phases: pre-processing, feature extraction, confidence assignment, and decision-making. Pre-processing algorithms perform tasks such as normalization of the data, corrections for variations in height and speed, removal of stationary effects due to the system response, etc. Methods that have been used to perform this task include wavelets and Kalman filters [87, 88], subspace methods and matching to polynomials [89], and subtracting optimally shifted and scaled reference vectors. Feature extraction algorithms reduce the pre-processed raw data to form a lower-dimensional, salient set of measures that represent the data. Principal component (PC) transforms are a common tool to achieve this task [90, 91]. Confidence assignment algorithms can use methods such as hidden Markov Models [92, 93], fuzzy logic [94], rules and order statistics [95], neural networks [96], or nearest neighbor classifiers [97] to assign a confidence that a mine is present at a point. Decision-

making algorithms often post-process the data to remove spurious responses and use a set of confidence values produced by the confidence assignment algorithm to make a final mine/no-mine decision.

In the following, we will outline four distinct feature-based algorithms on GPR data and one algorithm on WEMI data that have been applied to the landmine detection with promising results.

1 GPR Data

We use data collected by a vehicle mounted mine detection system (VMMDS). In this system, the GPR sensor [98] collects 24 channels of data. Adjacent channels are spaced approximately 6 centimeters apart in the cross-track direction, and sequences (or scans) are taken at approximately 1 centimeter down-track intervals. The system uses a V-dipole antenna that generates a wide-band pulse ranging from 200MHz to 7GHz. Each A-scan, that is, the measured waveform that is collected in one channel at one down-track position, contains 416 time samples at which the GPR signal return is recorded. Each sample corresponds to roughly 8 picoseconds. We often refer to the time index as depth although, since the radar wave is traveling through different media, this index does not represent a uniform sampling of depth. Thus, we model an entire collection of input data as a three-dimensional matrix of sample values, $S(z, x, y), z = 1, \cdots, 416; x = 1, \cdots, 24; y = 1, \cdots, N_S$, where N_S is the total number of collected scans, and the indices z, x,and y represent depth, cross-track position, and down-track positions respectively. A collection of scans, forming a volume of data, is illustrated in Figure 7.

Figure 8 displays several B-scans (sequences of A-scans) both down-track (formed from a time sequence of A-scans from a single sensor channel) and cross-track (formed from each channels response in a single sample). The objects scanned are (a) a high-metal content anti-tank (AT) mine, (b) a high-metal content anti-personnel (AP) mine, and (c) a

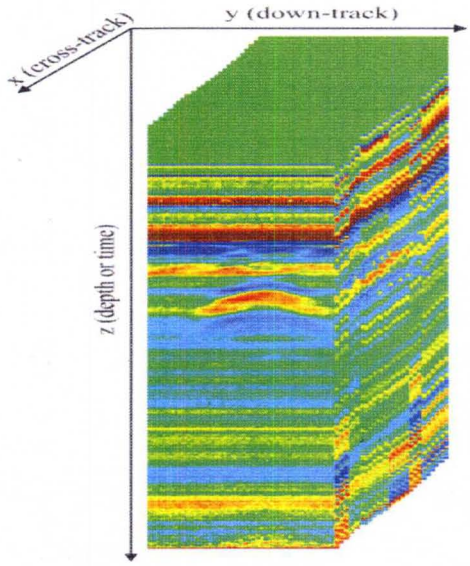


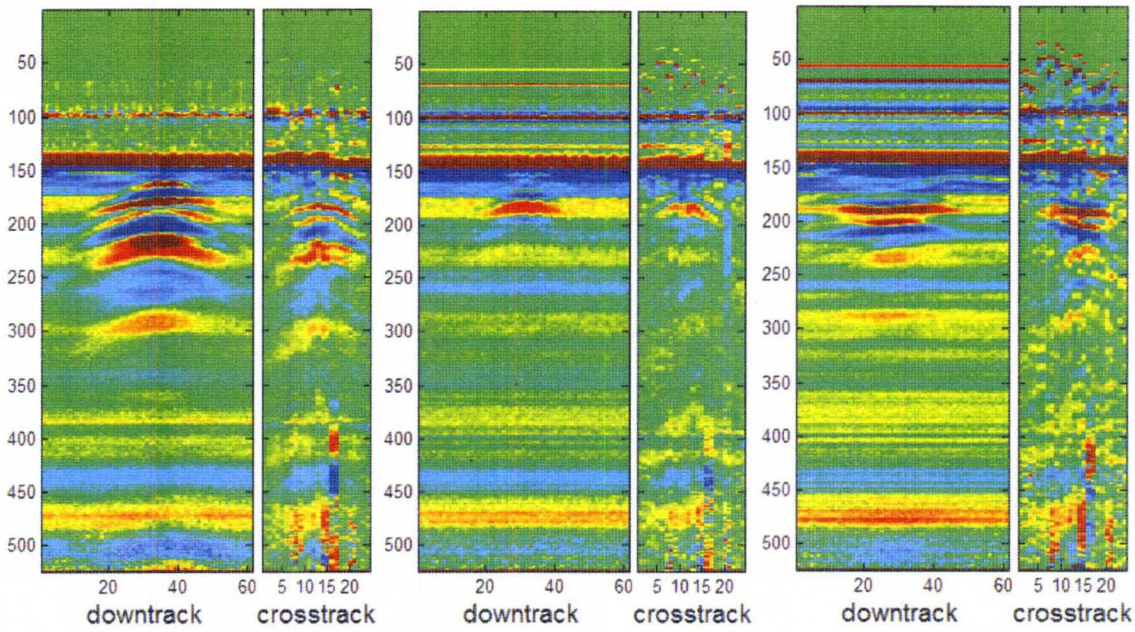
Figure 7. A collection of few GPR scans

wood block.

During data collection, the VMMDS is driven over the lanes with the GPR operating and saving data to disk. A global positioning system (GPS) on the VMMDS is used in conjunction with known locations of buried landmines to generate ground truth files that indicate the approximate locations of the landmine signatures in the GPR data files. For scoring purposes, alarms within a certain radial distance (25cm) from the edge of a landmine are considered detections and alarms more than 25cm from landmine edges are considered false alarms.

2 Hidden Markov Model (HMM) Algorithm

An Hidden Markov Model (HMM) is a model of a doubly stochastic process that produces a sequence of random observation vectors at discrete times according to an underlying Markov chain. At each observation time, the Markov chain may be in one of N_s states $\{s_1, \cdots, s_N\}$ and, given that the chain is in a certain state, there are probabilities of moving to other states. These probabilities are called the transition probabilities. An HMM



(a) AT High-metal mine (b) AP High-metal mine (c) Non-metal Clutter

Figure 8. NIITEK Radar down-track and cross-track B-scans pairs for 3 alarms

is characterized by three sets of probability density functions, the transition probabilities ($\mathcal{A}$), the state probability density functions ($\mathcal{B}$), and the initial probabilities (π). Let $\mathcal{T}$ be the length of the observation sequence (i.e.,number of time steps), let $O = \{O_1, \cdots, O_T\}$ be the observation sequence, and let $Q = \{q_1, \cdots, q_T\}$ be the state sequence. The compact notation is generally used to indicate the complete parameter set of the HMM model.

$$\lambda = (\mathcal{A}, \mathcal{B}, \pi) \tag{31}$$

In Eq. (31), $\mathcal{A} = [a_{ij}]$ is the state transition probability matrix, where $a_{ij} = Pr(q_t = j | q_{t-1} = i)$ for $i, j = 1, \cdots, N_s$; $\pi = \{\pi_i\}$, where $\pi_i = Pr(q_1 = s_i)$ are the initial state probabilities; and $\mathcal{B} = \{b_i(O_t), i = 1, \cdots, N\}$, where $b_i(O_t) = Pr(O_t | q_t = i)$ is the set of observation probability distribution in state i.

An HMM is called continuous if the observation probability density functions are continuous and discrete if the observation probability density functions are discrete. In the

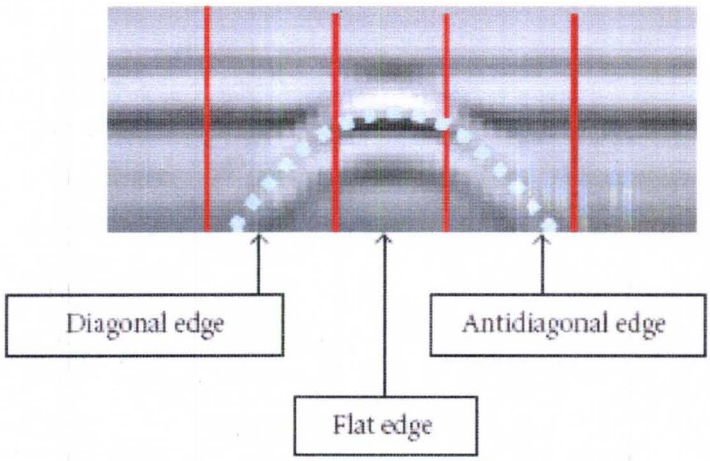


Figure 9. HMM Feature of a mine signature

case of the discrete HMM, the observation vectors are commonly vector quantized into a finite set of symbols, $\{v_1, v_2, \cdots, v_M\}$, called the codebook. Each state is represented by a discrete probability density function and each symbol has a probability of occurring given that the system is in a given state. In other words, $\mathcal{B}$ becomes a simple set of fixed probabilities for each class, that is, $b_i(O_t) = b_i(k) = Pr(v_k|q_t = i)$, where v_k is the symbol of the nearest code book of O_t . In the continuous HMM, $b_i(O_t)$ are defined by a mixture of some parametric probability density functions. The most common parametric pdf used in continuous HMM is the mixture Gaussian density:

$$b - i(O_t) = \sum_{m=1}^{M_i} c_{im} b_{im}(O_t) \tag{32}$$

where M_i is the number of components in state i, c_{im} is the mixture coefficient for the m^{th} mixture component in state i, and satisfies the constraints $c_{im} \geq 0$, and $\sum_{m=1}^{M_i} c_{im} = 1$, for $i = 1, \cdots, N$, and $b_{im}(O_t)$ is a K-dimensional multivariate Gaussian density with mean μ_{im} and covariance matrix C_{im}.

The HMM algorithm for landmine detection [92, 93] treats the down-track dimension as the time variable and produces a confidence that a mine is present at various posi-

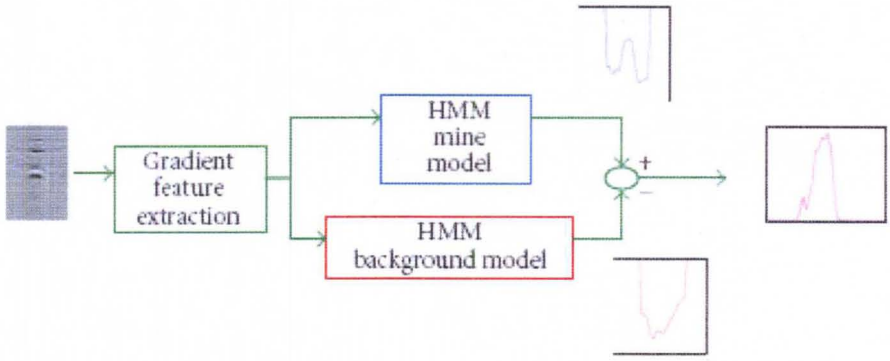


Figure 10. Illustration of the HMM-based model architecture

tions, (x, y), on the surface being traversed. In particular, a sequence of observation vectors is produced for each point. These observation vectors encode the degree to which edges occur in the diagonal and anti-diagonal directions. In particular, for every point (x_s, y_s), the strengths for the positive/negative diagonal/anti-diagonal edges is computed. Then, the observation vector at a point (x_s, y_s) consists of a set of features that encode the maximum edge magnitude over multiple depth values around (x_s, y_s). Figure 9 displays a hyperbolic curve superimposed on a preprocessed metal mine signature to illustrate the features of a typical mine signature.

The HMM classifier for landmine detection consists of two HMM models, one for mine and one for background. Each model has three states and produces a probability value by backtracking through model states using the Viterbi algorithm [99]. The mine model, λ^m, is designed to capture the hyperbolic spatial distribution of the features. λ^m has 3 states which correspond to the rising edge, flat, and decreasing edge. Each state is represented by 3 Gaussian components. The mine model is left to right model in that states are ordered and the transition probabilities for moving to a lower numbered state are zero. The background model is needed to capture the background characteristics and to reject false alarms. Each of the 24 channels is treated independently from the others, and has its own background model, λ^{bc}. In addition to allowing each channel to have a model that reflects its own

data, this decoupling allows the channels to be processed in parallel, and thus facilitating real-time operation. All λ^{b_c} (for $c = 1, \cdots, 24$) have 3 states and 3 Gaussian components per state. The probability value produced by the mine (background) model can be thought of as an estimate of the probability of the observation sequence given that there is a mine (background) present. The model architecture is illustrated in Figure 10.

3 Edge Histogram Descriptor (EHD) Algorithm

The Edge Histogram Descriptor (EHD) algorithm uses translation invariant features, that are based on the Edge Histogram Descriptor (EHD) of the 3D GPR signatures (refer to Figure 7), and a possibilistic K-Nearest Neighbors (K-NN) rule for confidence assignment [97]. The EHD is an adaptation of the MPEG-7 EHD feature [100] which captures the signature's texture as feature for recognition. For a generic image, the EHD represents the frequency and the directionality of the brightness changes in the image. Simple edge detector operators are used to identify edges and group them into five categories: vertical, horizontal, 45^o diagonal, 135^o diagonal, and isotropic (non-edges). The EHD would include five bins corresponding to the above categories. For the GPR data, we can adapt the EHD to capture the spatial distribution of the edges within a 3-D GPR data volume. To keep the computation simple, we still use 2-D edge operators, and we compute two types of edge histograms. The first one is obtained by fixing the cross-track dimension and extracting edges in the (depth, down-track) plane. The second edge histogram is obtained by fixing the down-track dimension and extracting edges in the (depth, cross-track) plane. Figure 11 displays a (depth,down-track) plane and a (depth,cross-track) plane of a sample mine signature with 60 depth values. As it can be seen, the edges in these planes and their spatial distribution constitute an important feature to characterize the mine signatures.

To generate the histogram, local edges are categorized into five types: vertical, horizontal, diagonal (45^o rising), anti-diagonal (45^o falling), and non-edges. These edges are

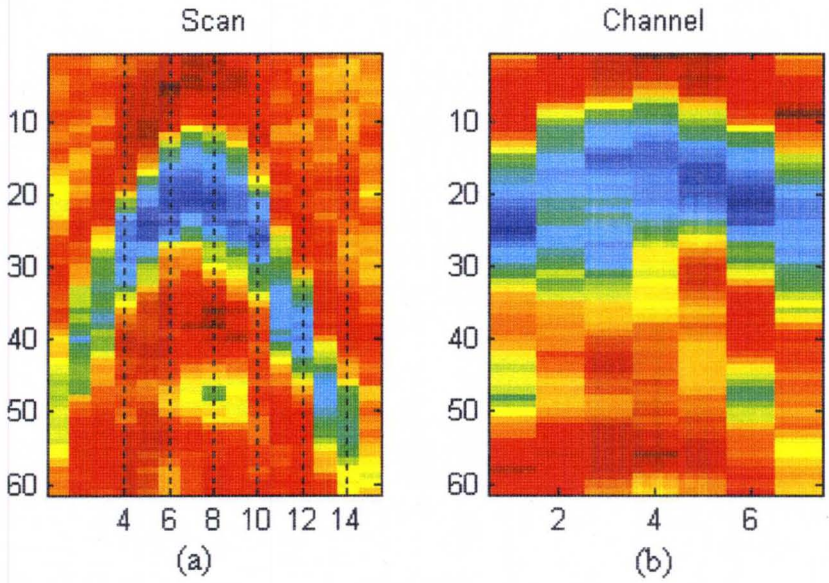


Figure 11. (a) (depth-downtrack), and (b) (depth-crosstrack) views of a sample mine signature models

superimposed on a typical mine signature in Figure 12. As it can be seen these edges represent a good approximation of the mine signature.

Let $S_{zy}^{(x)}$ be the x^{th} plane of the 3-D signature $S(x, y, z)$. First, for each $S_{zy}^{(x)}$, we compute four categories of edge strengths: vertical, horizontal, 45^o diagonal, and 135^o diagonal. If the maximum of the edge strengths exceeds a certain preset threshold, θ_G, the corresponding pixels is considered to be an edge pixel. Otherwise, it is considered a non edge pixel. Next, each $S_{zy}^{(x)}$ image is vertically subdivided into 7 overlapping sub-images $S_{zy_i}^{(x)}$, $i = 1, \cdots, 7$. For each $S_{zy_i}^{(x)}$, we compute a 5 bin edge histogram, $H_{zy_i}^{(x)}$, where the bins correspond to the 7 edge categories, and the non-edge pixels (refer to Figure 11). The down-track component of the EHD, EHD^d is defined as the concatenation of 7 five-bin histograms in Eq. 33:

$$EHD^d(S_{xyz}) = [\overline{H}_{zy_1}, \overline{H}_{zy_2}, \cdots, \overline{H}_{zy_7}] \tag{33}$$

where $\overline{H}_{zy_i}$, $i = 1, \cdots, 7$, is the cross-track average of the edge histograms of sub-image

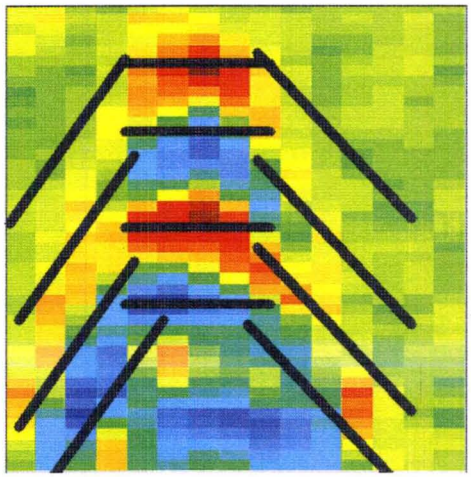


Figure 12. Diagonal, anti-diagonal, and horizontal edges superimposed on a typical mine signature

$S_{zy_i}^{(x)}$ over N_C channels, i.e.,

$$\overline{H}_{zy_i} = \frac{1}{N_C} \sum_{x=1}^{N_C} H_{zy_i}^{(x)} \tag{34}$$

To compute the cross-track component of the EHD, EHD^x, we fix the scans, and compute the 4 edge strengths on the $S_{zx}^{(y)}$, $y = 1, \cdots, NS$ (depth,cross-track) planes. Since these planes do not have enough columns (typically < 7), they are not divided into sub-images, and only one global histogram per plane, $H_{zx}^{(y)}$, is computed. Then, EHD^x is computed as the down-track average of the edge histograms over N_S scans, in Eq. 35,

$$EHD^x(S_{xyz}) = \frac{1}{N_S} \sum_{x=1}^{N_S} H_{zx}^{(y)} \tag{35}$$

The EHD of each 3-D GPR alarm is a 40-D histogram that concatenates the down-track and cross-track EHD components, i.e.,

$$EHD(S_{xyz}) = [EHD^d(S_{xyz}) \;\; EHD^x(S_{xyz})] \tag{36}$$

The extraction of the EHD is illustrated in Figure 13. Each signature s consists of a 30 (depth values) by 15 (scans) by 7 (channels) volume extracted from 7 consecutive channels extracted from channel x_s of the aligned GPR data and centered at (y_s, z_s).

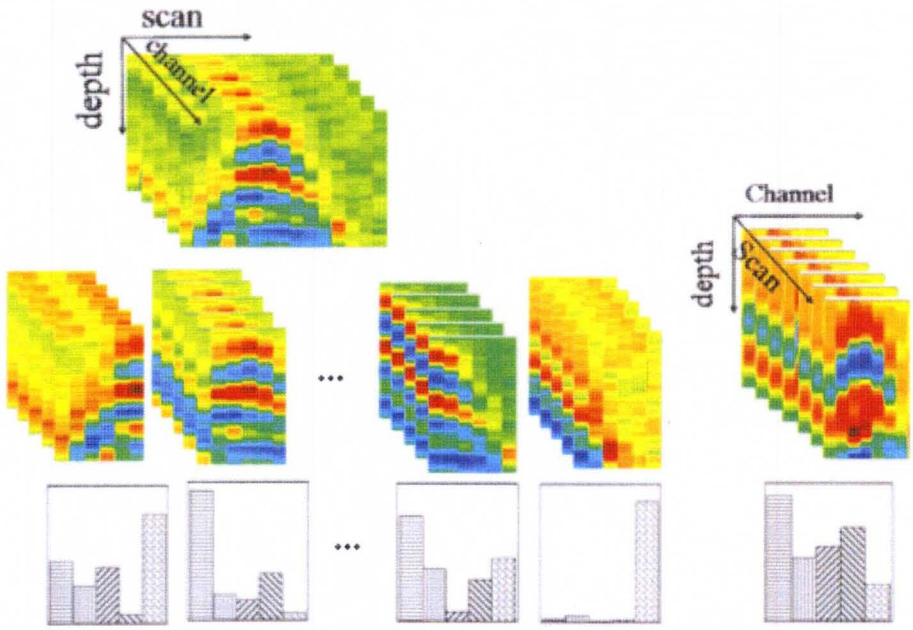


Figure 13. Extraction of the EHD for a 3-D mine signature

A set of alarms with known ground truth is used to train the decision-making process. These labeled alarms are clustered to identify a small number of representative prototypes that capture signature variations due to differing soil conditions, mine types, weather conditions, and so forth. To reduce the size of the training samples and identify few representatives that can capture these within-class variations, two self-organizing feature maps (SOFM) [101] are used to cluster the mine and false alarms signatures separately. We will refer to the clusters representatives R_i as prototypes. We use R_i^M to denote the prototypes of the mine signatures, and R_i^C to denote the prototypes of the clutter signatures.

For a given test signature, S_T , We slide a $30 \times 15 \times 7$ window size along the depth axis with a 50% overlap between 2 consecutive signatures. A maximum of 10 signatures are extracted for each target. For each signature, we extract the EHD features as described above and compute its distance to all representative prototypes, then sort these distances, and identify the top K nearest neighbors $S_T^1, \cdots, S_T^K$. The confidence value is then com-

puted using Eq. (37):

$$Conf(S_T) = \frac{\sum_{k=1}^{K} u^M(S_T^k) \times \frac{1}{dist(S_T, S_T^k)}}{\sum_{k=1}^{K} \frac{1}{dist(S_T, S_T^k)}} \tag{37}$$

where u^M is the label using Eq. (38):

$$u^M(R_i) = \frac{1/dist(R_i, R_i^M)}{1/dist(R_i, R_i^M) + 1/dist(R_i, R_i^C)} \tag{38}$$

Where R_i^M, R_i^C are mine prototype and clutter prototype respectively.

4 Geometric Feature FOWA ROCA (GEOM) Algorithm

The geometric feature FOWA ROCA algorithm, GEOM, is based on a single hidden-layer Feed-forward Order-Weighted-Average (FOWA) network [96], which is essentially a perceptron with a combination of scalar and order-weighted-average vector input features. The features presented to this network are the geometric features of the FROSAW landmine detection algorithm [95]. These features are captured in a depth-bin whitened version of the GPR data. The GPR data are segmented into a sequence of subimages that overlap in the depth dimension. To reduce noise, de-correlate time samples, and reduce computational burden, principal component analysis (PCA) is used to reduce the number of elements in depth bins on a channel-by-channel basis.

It has been consistently observed that in many of the depth bins the whitened energy signal for mines has a compact, solid, circular shape (sometimes also accompanied with outer rings). One the other hand, whitened energy signals for non-minelike false alarms (i.e., those alarms having raw GPR signatures that humans qualitatively label as non-manlike) tend to be irregular. Based on these observations, four features, i.e., compactness, eccentricity, solidity, and area/filled area ratio, are computed from whitened energy signals for discriminating mines and non-mines.

To improve the algorithm's accuracy, an iterative technique that maximizes the area under the ROC curve is used [102].

5 Spectral Feature (SPECT) Algorithm

In contrast to the geometric features and the edge histogram features, the spectral feature (SPECT) algorithm aims at capturing the characteristics of a target in the frequency domain. It extracts the alarm Spectral Correlation Feature (SCF), and formulates a confidence value based on similarity to prototypes that characterize mine objects [103].

The spectral features are derived from the Energy Density Spectrum (EDS) of an alarm declared by the pre-screener. The estimation of EDS involves three main steps: *pre-processing, whitening, and averaging.* Pre-processing estimates the ground level, aligns the data from each scan with respect to ground level, and removes the data above and near the ground surface. This step is needed to avoid an EDS that is dominated by the response of the ground bounce. The whitening step performs equalization on the spectrum from the background so that the estimated EDS reflects the actual spectral characteristics of an alarm. Averaging reduces the variance in the EDS.

6 WEMI Data and Algorithm

The Wideband Electro-Magnetic Induction (WEMI) sensor was developed by Scott [104]. This sensor measures the response of an object at 21 logarithmically spaced frequencies over the range 330 Hz to 90 KHz. The goal is to obtain characteristic spectral shapes that can help discriminate objects of interest from false alarms.

The response of the system can be modeled as

$$S(w) = A[I(w) + iQ(w)], \tag{39}$$

where w is frequency, A is magnitude and $I(w)+iQ(w)$ describes the shape of the response as a function of frequency. An input data point is composed of 21 complex responses at the following measured frequencies (in Hz.): 330, 390, 510, 690, 930, 1230, 1650, 2190, 2910, 3930, 5190, 6930, 9210, 12210, 16230, 21630, 28770, 38250, 50850, 67650, and 90030.

Before feature extraction, the I and Q values are normalized between 0 and 1. This eliminates variation in magnitude due several factors - such as the depth of the buried object to be detected as well as metal mass and content - that do not affect the shape of the response curve. The magnitude can always be measured separately. After normalization, the response models proposed by Miller et al. [58] are used to fit the curve. The 3-parameter model is given by

$$I + iQ = q\Big(s + \frac{(iw\tau)^{1/2} - 2}{(iw\tau)^{1/2} + 1}\Big) \tag{40}$$

where q, s, and τ are the three parameters describing the shape of the response curve. The value q represents the magnitude of the response curve after normalization, s does the shift in the frequency axis, and τ controls the rate of shape change. To fit the curve, we used a built-in Matlab function, *lsqcurvefit* that fits the functional form in (40) to the data. The parameters resulting from this curve fit plus the error in the fit provide 4 features. Figure 14 displays the response curves and their curve fits of metallic and non-metallic objects. We note that other researchers, such as Fails [105] and Gader [106] have also used these model parameters as features.

In addition to the 4 features provided by the model, 3 spread features [107] are used. These are defined by the following equations in which I and Q represent the In-phase (Real) and Quadrature (Imaginary) values at each frequency and N is the number of frequencies.

$$\begin{aligned} Q_{sum} &= \sum_{i=1}^{N} Q_i, \\ Q_{spread} &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |Q_i - Q_j| \\ T_{spread} &= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |Q_i - Q_j| + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |I_i - J_j| \end{aligned} \tag{41}$$

Together, these make up the 7 features used to describe a WEMI signal. Feature selection was performed using the well-known divergence measure. Four features were selected: τ,

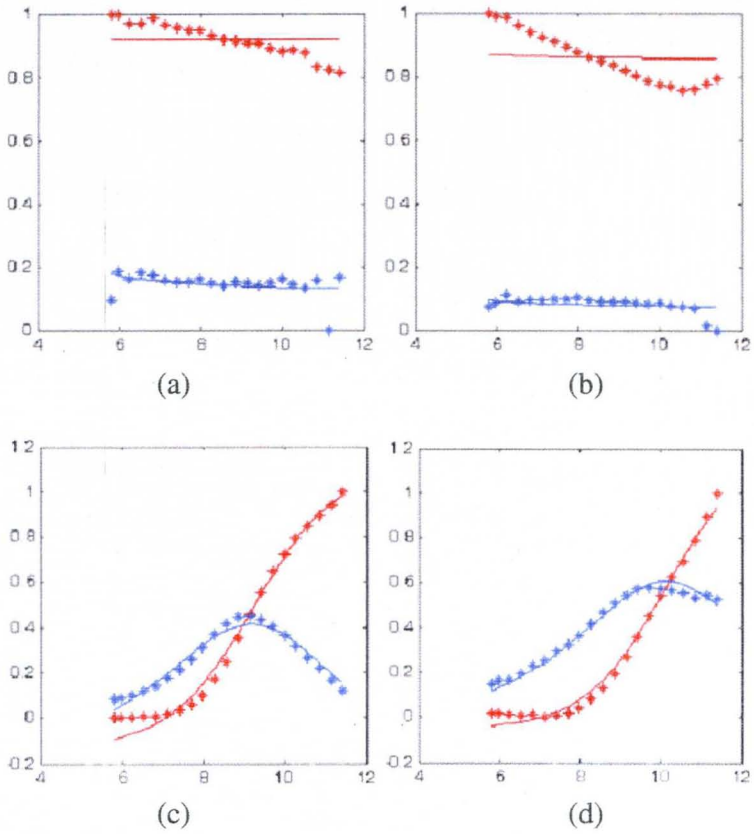


Figure 14. Response curves (sequences of dots) and their curve fits (smooth curves) from (a) blank, (b) non-metallic clutter item, (c) metallic clutter item, and (d) low-metal mine

the fitting error, Q_{spread}, and T_{spread}. A Multi-Layer Perceptron (MLP) classifier was built from these features. The MLP parameters were identified through 6-fold cross-validation. We will refer to this classifier as the WEMI detector.

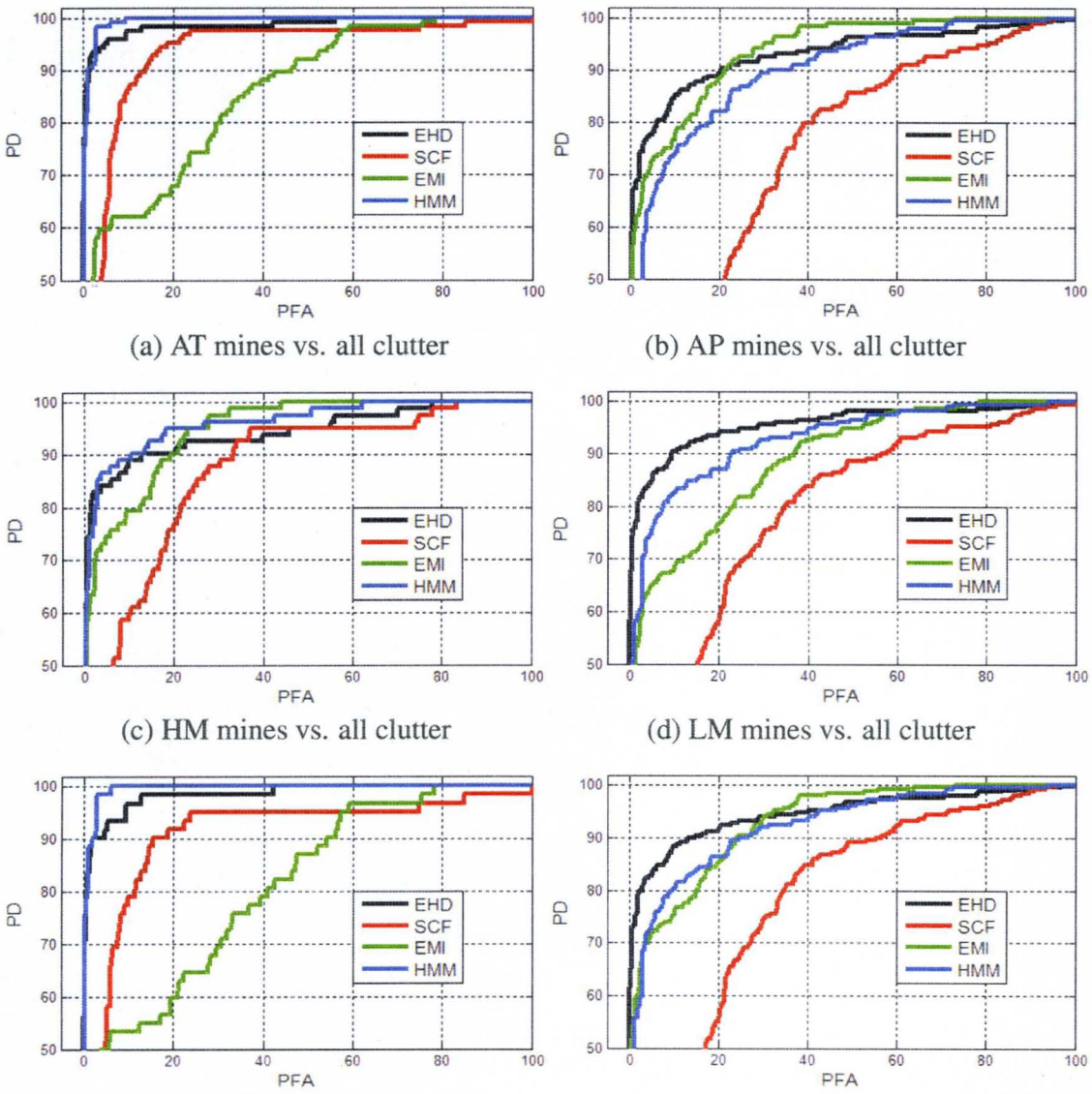
CHAPTER IV

CONTEXT-DEPENDENT FUSION

A Motivations

Our Context-Dependent Fusion (CDF) is motivated by the observation that there is no single algorithm that can consistently outperform all other algorithms. In fact, the relative performance of different algorithms can vary significantly depending on the algorithms adaption, feature types, and sensor styles. The proposed CDF is a local approach that adapts the fusion to different regions of the feature space. It can take advantages of the strengths of several algorithms in different regions of the feature space without being affected by the weaknesses of the other algorithms, and also avoid loosing potentially valuable information introduced by other weak classifiers.

For landmine detection, the relative performance of different detectors can vary significantly depending on the mine type, geographical site, soil and weather conditions, and burial depth. To illustrate the above point, in Figure 15 we show the Receiver Operation Characteristic (ROC) of the four discrimination algorithms on various subsets of data collected by the NIITEK robotic mine detection system. This data collection will be described in Chapter VI. The different ROC's display the performance of the algorithms when different types of mines are scored. For instance, in Figure 15(a), only anti-tank (AT) mines are considered. In this case, the HMM and EHD detectors have the best performance, and the WEMI has the worst. This is because AT mines are large enough to have good GPR signatures and many of them have low metal content. However, for AP mines, the WEMI detector has the best performance at high probability of detection as shown in Figure 15(b).



(a) AT mines vs. all clutter

(b) AP mines vs. all clutter

(c) HM mines vs. all clutter

(d) LM mines vs. all clutter

(e) Deep mines vs. all clutter

(f) Shallow mines vs. all clutter

Figure 15. Performance of 4 different detectors for different types of mines buried at different depths.

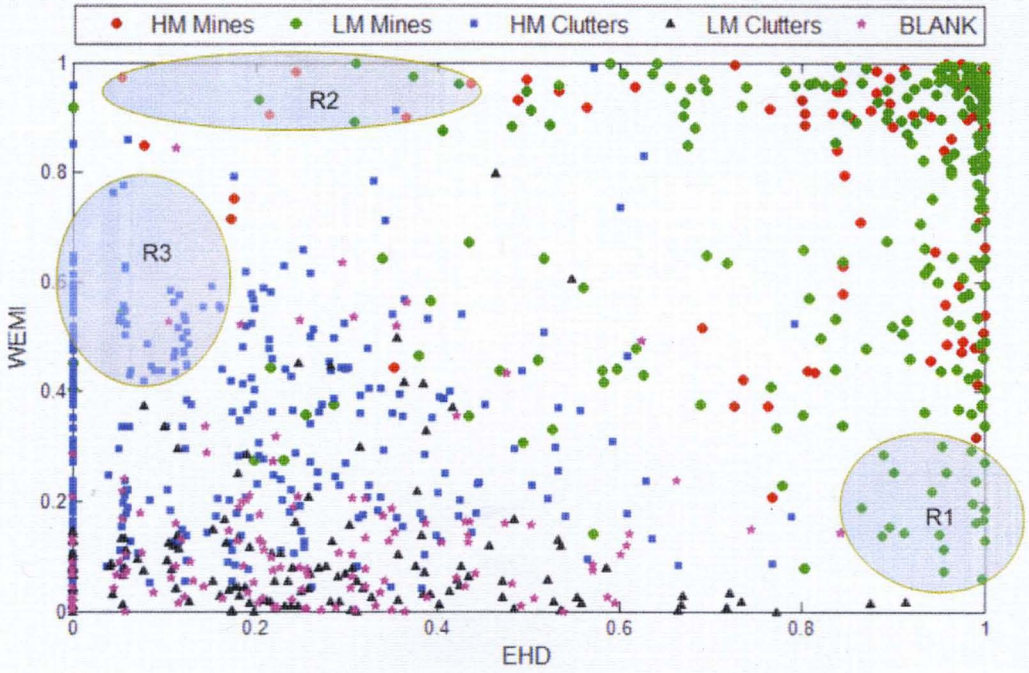


Figure 16. Comparison of the EHD and WEMI outputs for several mine and clutter signatures

In this case, several AP mines have weak GPR signatures and cannot be easily detected by any of the GPR algorithms. The relative performance of the different algorithms depend on other factors besides the mine type. For instance, as shown in Figure 15(e) and (f), the WEMI detector has a poor performance for deeply buried mines, but a relatively better performance for shallow mines. The poor performance of the GPR detectors in the latter case may be due to the difficulty in decoupling the mine signatures and the background signal around the ground bounce area.

The relative performance of the different detectors is illustrated further in Figure 16 where we display a scatter plot of the confidence values generated by the EHD and WEMI detectors for all alarms in the data collection. As it can be seen, the relative performance of the different algorithms can vary significantly. For instance, region (R1) highlights a group of mines with low metal content that are easily detected by the EHD (confidence close to 1) and not by the WEMI (confidence less than 0.25). On the other hand, region

(R2) highlights a different set of mine alarms (both HM and LM) that are easily detected by the WEMI and not the EHD. Most of these mines are buried at a depth less than 2 inches, and their GPR signatures are intertwined with the ground bounce. Region (R3) displays a group of metal clutter alarms that will be detected by the WEMI and rejected by the EHD.

The above examples suggest using different algorithms and/or features to accommodate for the different mine types, burial depths, and other conditions. However, this task may not be as simple as it sounds since it is not possible to characterize the performance of each algorithm on all possible variations. Moreover, it may not be possible to know the characteristics of the test site. Thus, the selection of the optimal subset of algorithms is not a trivial task and need to be learned in an unsupervised way.

B Proposed Approach

Motivated by the previous examples, we propose a Context-Dependent Fusion (CDF) framework that can take advantages of the strengths of few algorithms in different regions of the feature space without being affected by the weaknesses of the other algorithms. Figure 17 shows the overall architecture of the proposed CDF scheme. Some algorithms are not feature-based and they simply assign a confidence value using the raw data. Other algorithms extract their own sets of features and generate a confidence value. The different algorithms could operate on data from different sources. This figure also highlights the two main components of the training phase, namely, *context extraction* and *algorithm fusion*. In *context extraction*, the features extracted by the different algorithms from differen sources are combined. A clustering algorithm is used to partition the training data in the combined feature space into groups of similar signatures, or contexts. Here, we are assuming that signatures that have similar responses to different algorithms share some common characteristics, and would be assigned to the same cluster by the clustering algorithm. Actually, this is the main objective of any clustering algorithm.

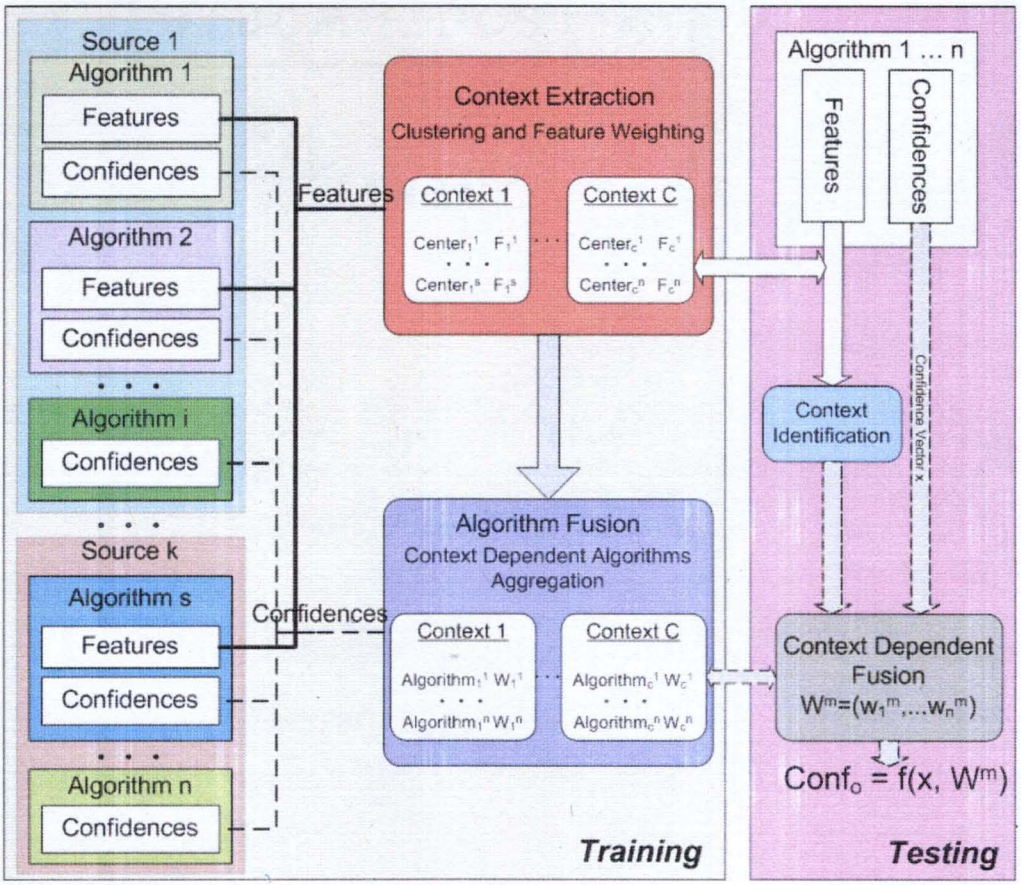


Figure 17. Architecture of the proposed Context-Dependent Fusion

After partitioning the feature space, the initial feature space with the combined features is transformed into a new space with $C \times L \times K$ confidence/decision features, where C is the number of contexts, L is the number of algorithms and K is the number of classes. Then, the training data on the confidence feature space from each identified context will be used to learn the optimal fusion parameters and identify "*local experts*" for that region in the *algorithm fusion* component.

To test a new signature using CDF, each algorithm would extract its set of features and assign a confidence value for the test pattern. The features are used to assign the test sample to the best/nearest context. The fusion parameters of this context are used to fuse the individual confidence values and obtain a final decision value.

For each context, several methods, e.g. those mentioned in Chapter II, can be used

to learn the optimal fusion parameter [108, 109, 110, 111, 112]. For instance, if we select only the best algorithm in each context, our CDF reduces to a *dynamic classifier selection local accuracy* (DCS-LA) [10]; while if we adapt *decision temple* (DT) in each context, our approach can be viewed as a generalized DT [7].

1 Context Extraction

In context extraction, the features extracted by the different individual algorithms are combined. This step can be seen as feature fusion. We assume that we have L detectors, and that each detector extracts a set of different features. The objective is to cluster the L feature sets and identify regions that correspond to homogeneous alarm signatures in various subspaces of the original space. This task can be achieved using an algorithm that performs clustering and feature discrimination simultaneously. This algorithm is described in the following section.

2 The Coarse Simultaneous Clustering and Attribute Discrimination ($SCAD_c$) Algorithm

Clustering in machine learning strives to partition a data set into groups (clusters), so that the data in each group share some common trait. A trait (feature) is defined as common through distance and similarity measures. The advantages to clustering are its unsupervised learning ability and capability to support many distance measures. However, when the features come from different algorithms, traditional algorithms such as FCM [113], Expectation-Maximization (EM) [39] are not appropriate. This is because different feature sets can vary in size, dynamic range, and should not be treated equally. Moreover, irrelevant features can adversely affect cluster definitions. Thus, it is recommended to identify cluster-dependent relevance weights for each feature subset [114, 115, 116].

For high dimensional data, learning a relevance weight for *each* feature may result

in overfitting. To avoid this, a coarse feature weighting approach to called SCAD$_c$ [117] was used. Instead of learning a weight for each feature, the set of features is divided into logical subsets, and a single weight is learned for each of these subsets.

Let $\mathcal{X} = \{\mathbf{x}_j \in \Re^p | j = 1, \cdots, N\}$ be a set of N feature vectors in an n-dimensional feature space. Let $\mathcal{B} = (\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_c)$ represent a C-tuple of prototypes each of which characterizes one of the C clusters. Each $\boldsymbol{\beta}_i$ consists of a set of parameters. Let u_{ij} represent the membership of $\mathbf{x}_j$ in cluster $\boldsymbol{\beta}_i$. The $C \times N$ fuzzy C-partition $\mathbf{U}=[u_{ij}]$ satisfies [17]:

$$\begin{cases} u_{ij} \in [0,1], & \forall i \\ 0 < \sum_{j=1}^{N} u_{ij} < N & \forall i, j \\ \sum_{i=1}^{C} u_{ij} = 1 & \forall j \end{cases} \tag{42}$$

Assume that we have L algorithms and that each algorithm extracts a set of features $FS^s, s = 1, 2, \cdots, L$, and that each subset FS^s, includes k^s features. Let d_{ij}^s be the partial distance between $\mathbf{x}_j$ and cluster i using the s^{th} feature subset. Let $\mathbf{V} = [v_{is}]$ be the relevance weight for FS^s with respect to cluster i. The total distance, D_{ij}, between $\mathbf{x}_j$ and cluster i is then computed by aggregating the partial distances and their weights, i.e.,

$$D_{ij}^2 = \sum_{s=1}^{L} v_{is} \left(d_{ij}^s\right)^2. \tag{43}$$

SCAD$_c$ [117] minimizes

$$J(\mathbf{B}, \mathbf{U}, \mathbf{V}; \mathcal{X}) = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^m \sum_{s=1}^{L} v_{is} \left(d_{ij}^s\right)^2 + \sum_{i=1}^{C} \delta_i \sum_{s=1}^{L} v_{is}^2, \tag{44}$$

subject to Eq. (42) and

$$v_{is} \in [0,1] \; \forall \; i, \; s; \quad \text{and} \quad \sum_{s=1}^{L} v_{is} = 1, \; \forall \; i. \tag{45}$$

Optimization of J with respect to $\mathbf{V}$ yields:

$$v_{is} = \frac{1}{L} + \frac{1}{2\delta_i} \sum_{j=1}^{N} (u_{ij})^m \left[D_{ij}^2 / L - (d_{ij}^s)^2 \right]. \tag{46}$$

The first term in Eq. (46), $(1/L)$, is the default value if all L feature subsets are treated equally, and no discrimination is performed. The second term is a bias that can be either positive or negative. It is positive for compact feature sets where the partial distance is, on the average, less than the total distance (normalized by the number of features). If a feature set is compact, compared to the other features, for most of the points that belong to a given cluster (high u_{ij}), then it is very relevant for that cluster.

Minimization of J with respect to $\mathbf{U}$ subject to the constraints in (42) yields:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(D_{ij}^2 / D_{kj}^2\right)^{\frac{1}{m-1}}}. \tag{47}$$

Minimization of J with respect to the prototype parameters depends on the choice of d_{ij}^s. Since the partial distances are treated independent of each other (i.e., disjoint feature subsets), and since the second term in Eq. (44) does not depend on prototype parameters explicitly, the objective function in Eq. (44) can be decomposed into L independent problems:

$$J_s = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^m v_{is} \left(d_{ij}^s\right)^2, \quad \text{for } s = 1, \cdots, L. \tag{48}$$

Each J_s would be optimized with respect to a different set of prototype parameters. For instance, if d_{ij}^s is an Euclidean distance, minimization of J_s would yield the following update equation for the centers of subset s,

$$\mathbf{c}_i^s = \frac{\sum_{j=1}^{N} u_{ij}^m \mathbf{x}_j^s}{\sum_{j=1}^{N} u_{ij}^m}. \tag{49}$$

SCAD$_c$ is an iterative algorithm that starts with an initial partition and alternates between the update equations of u_{ij}, v_{is}, and c_i^s. The SCAD$_c$ algorithm is summarized below:

Coarse SCAD

Fix the number of clusters C;
Fix m, $m \in (1, \infty)$;
Initialize the centers and fuzzy partition matrix **U**;
Initialize the relevance weights to $1/L$;
Repeat
    Compute $(d_{ij}^s)^2$ for $1 \leq i \leq C$, $1 \leq j \leq N$, and $1 \leq s \leq L$;
    Update the relevance weights v_{is} using Eq. (46);
    Compute D_{ij}^2 using Eq. (43);
    Update the partition matrix $U^{(k)}$ using Eq. (47);
    Update the centers using Eq. (49);
Until(centers stabilize)

3 Algorithm Fusion

Any of the fusion methods mentioned in Chapter II could be integrated within our context-dependent fusion. Training data from each identified context would be used to learn the optimal fusion parameters and identify "*local experts*" for that region of the feature space. In the next Chapter, we will propose six local method to calculate weights based on training performance.

4 Testing Step

To test a new signature (x_t) using CDF, each detector ($\mathcal{D}_i$) would extract its set of features F^i and assigns a confidence values y_{it}. The L sets of descriptors are then used to identify the closest context. This is achieved by comparing the features of the test sample to the centroids of the clusters representing the different contexts. The partial distances, produced by the features of each algorithm, are combined using the cluster-dependent feature relevance weights learned in the context extraction phase. Once context c has been identified as the closest context to the sample being tested, the confidence values of the individual algorithms would be aggregated using the optimal set of aggregation weights w_i^c associated with context c.

5 Computational Complexity

The proposed CDF approach is generic and does not require a specific set of features or classifiers. Thus, its overall computational complexity cannot be determined. However, we can compare it to the alternative solution with similar settings. The CDF has two additional steps over standard global fusion. The first one consists of partitioning the training data into clusters or contexts. This is an off-line step that needs to be done only once, and thus does not affect the computational complexity in the testing mode. The second step consists of identifying the closest context to a test sample. This step involves simple distance computation to identify the nearest cluster prototype. It requires $\mathcal{O}(\mathcal{C}\times p)$ computations, where C is the number of contexts and p is the dimensionality of the composite feature vector representing the alarm. Since the expected number of contexts, C, is typically small (less than 20), and the number of algorithms to be fused is less than 4, this additional computation is negligible when compared to the computation needed for feature extraction and classification.

CHAPTER V

LEARNING LOCAL WEIGHTS FOR CONTEXT-DEPENDENT FUSION

Any of the fusion methods mentioned in Chapter II could be integrated into our context-dependent fusion. Training data from each identified context would be used to learn the optimal fusion parameters and identify "*local experts*" for that region of the feature space. In this Chapter, we first define the histogram and the Receiver Operating Characteristic (ROC), and show how these measures can be used to optimize the fusion weights. Then, we propose six different methods to assign degrees of worthiness to each algorithm. These weights will be embedded into the proposed Context-Dependent Fusion (CDF) approach to perform local fusion.

A Histograms and Cumulative Histograms

A histogram is simply a mapping m_i that counts the number of observations that fall into various disjoint categories (known as *bins*). If n is the total number of observations and k is the total number of bins, the histogram m_i meets the following conditions:

$$n = \sum_{i=1}^{k} m_i \tag{50}$$

The cumulative histogram is a mapping that counts the cumulative number of observations in all of the bins up to the specified bin. That is, the cumulative histogram M_i of a histogram m_i is defined as:

$$M_i = \sum_{j=1}^{i} m_j, \tag{51}$$

TABLE 1

Contingency Table

	p	n	total
p'	True Positive	False Positive	P'
n'	False Negative	True Negative	N'
total	P	N	P+N

B Receiver Operating Characteristic (ROC) Curve

The ROC curve is a graphical plot of the sensitivity vs. specificity for a binary classifier system as its discrimination threshold is varied. The ROC can also be represented equivalently by plotting the fraction of true positives (TPR = true positive rate) vs. the fraction of false positives (FPR = false positive rate).

Consider a two-class prediction problem (binary classification), in which the outcomes are labeled either as positive (p) or negative (n) class. There are four possible outcomes from a binary classifier. If the outcome from a prediction is p' and the actual value is also p, then it is called a true positive (TP); however if the actual value is n then it is said a false positive (FP). Conversely, a true negative occurs when both the prediction outcome and the actual value are n, and false negative is when the prediction outcome is n' while the actual value is p.

Let us define an experiment from P positive instances and N negative instances. The four outcomes can be formulated in a 2×2 contingency table or a confusion matrix (refer to Table 1):

To draw an ROC curve, only the true positive rate (TPR) and the false positive rate (FPR) are needed. TPR determines a classifier or a diagnostic test performance on classifying positive instances correctly among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results while they are actually

negative among all negative samples available during the test.

An ROC space is defined by FPR and TPR as x and y axes respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). Since TPR is equivalent with sensitivity and FPR is equal to 1-specificity, the ROC graph is sometimes called the sensitivity vs (1-specificity) plot. Each prediction result or instance of a confusion matrix represents one point in the ROC space.

C Separation-Based Degree of Worthiness

Usually, the essence of any classifier is to approximate the posterior probability to arbitrary decision. However, given the fact that training data is not infinite and noisy, the true *a posteriori* is hard to estimate. From probability theory, we know that when a random variable takes values in the set of real numbers, the probability distribution $f_X(.)$ can be completely described by the cumulative distribution function (or called probability distribution function) $F_X(.)$, whose value at each real x is the probability that the random variable is smaller than or equal to x, i.e.,

$$x \mapsto F_X(x) = P(X \leq x) \tag{52}$$

Using a cumulative function, the decision space **x** can be transferred into mapped posterior probability (MPP). Here, we define the MPP function as $P(\Omega_k|\mathbf{x})$, and conditional MPP (cMPP) $p(\Omega_k|x_i)$ for each algorithm i in class Ω_k. Then the performance of each algorithm can be estimated based on their cMPP value on training data.

Suppose that we have L algorithms, and K classes, let $\boldsymbol{w}^{c_s} = \{w_1^{c_s}, w_2^{c_s}, \cdots, w_L^{c_s}\}$ be a vector of real numbers within each context c_s in CDF frame, such that:

$$\sum_{i=1}^{L} w_i^{c_s} = 1 \ \textit{and} \ 0 \leq w_i^{c_s} \leq 1 \tag{53}$$

where $i = 1, 2, \cdots L$.

Using a linear discriminant function, which has been mathematically analyzed for fusion [118], we can assign new confidence value for test pattern $\mathbf{x} = \{x_1, x_2, \cdots, x_L\}$ according to the individual confidence value $x_i, i = 1, 2, \cdots L$, which come from different algorithms using:

$$g(\mathbf{x}) = \sum_{i=1}^{L} w_i^{c_s} \cdot x_i^{c_s} \tag{54}$$

Since the discriminant functions are linear, the decision boundaries are hyperplane.

The classifiers' weights could be determined based on the relative separation between the distribution of the classes confidence values. Intuitively, algorithms with larger separation are considered more "expert" since they can discriminate between the classes boundaries, and thus should be assigned larger weights. Because the data we are using is two-category case, for this case, we define the degree of separation between two classes using algorithm i in context c_s as:

$$\tilde{w}_i^{c_s} = \frac{\sum_{j=1}^{N} p(\Omega_k|x_{ij}) \cdot 1(.)}{\sum_{j=1}^{N} 1(.)} - \frac{\sum_{j=1}^{N} p(\Omega_m|x_{ij}) \cdot \bar{1}(.)}{\sum_{j=1}^{N} \bar{1}(.)} \tag{55}$$

where $1(.) = 1(x_{ij} \in \Omega_k \cap x_{ij} \in c_s)$ and $\bar{1}(.) = 1(x_{ij} \in \Omega_m \cap x_{ij} \in c_s)$ are indicators.

Figure 18 illustrates the separation between distributions of confidence value in two classes. The red curve display the distribution in class 1, and the blue curve displays the cumulative distribution in class 2 when subtracted from 1. The centroid in the figure represents the average of the cMPP for one class within one context.

To satisfy the constriction in Eq. 53, we normalize $\tilde{w}_i^{c_s}$ in Eq. (55) using:

$$w_i^{c_s} = \frac{\tilde{w}_i^{c_s}}{\sum_{i=1}^{L} \tilde{w}_i^{c_s}} \tag{56}$$

D Overlap-Based Degree of Worthiness

The Overlap-Based Degree of Worthiness Algorithm is similar to the Separation algorithm. It uses the overlap of confidence value between cumulative histogram distribu-

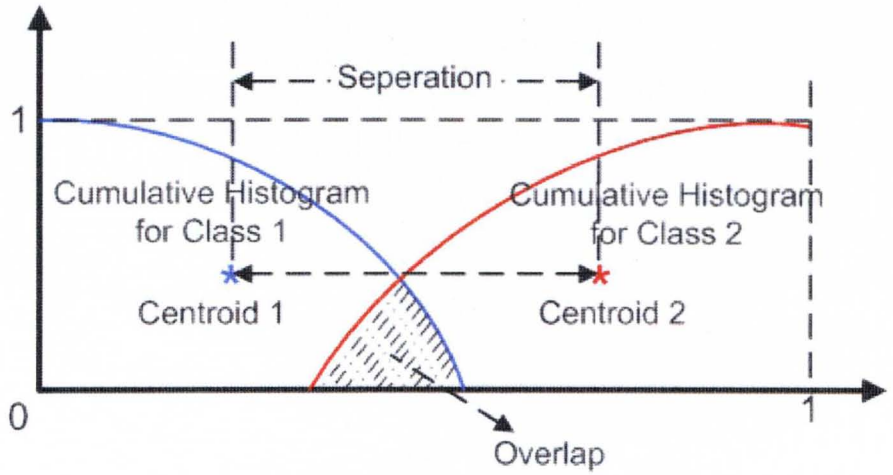


Figure 18. Cumulative Histogram Distribution of Individual Classifier. Shade area is Overlap; Separation is defined as the distance between the two classes centroids; Red curve is one class confidence cumulative histogram, Blue curve is the other class inverse confidence cumulative histogram

tions. Algorithms with smaller overlap are considered more "expert" for the cluster under consideration, and should be assigned larger weights.

The shaded area in Figure 18 shows the overlap between the two cumulative histograms.

Let O_k denote the Overlap value of two classes' cumulative histogram distribution for algorithm k. The degree of worthiness of algorithm k in context i is then computed using

$$w^i = \frac{\frac{1}{(O_k^i+\varepsilon)^m}}{\sum_{j=1}^{L} \frac{1}{(O_j^i+\varepsilon)^m}} \tag{57}$$

In Eq. (57), ε is a very small value to make sure the overlap $O_k^i \neq 0$.

E ROC Area-Based Degree of Worthiness

Recent work [58] has shown that the area under the Receiver Operating Characteristic (ROC) curve (AUC) is an unbiased estimator of discrimination accuracy. Algorithms

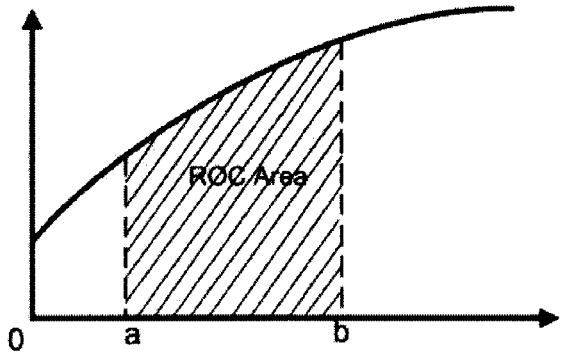


Figure 19. Area under the ROC for an interval of interest [a, b]

with larger area are considered more "expert" for the cluster under consideration, and will be assigned larger weights [110]. This area could be computed over the entire domain, or could be restricted to an interval [a, b] as shown in Figure 19.

The AUC can be computed using:

$$AUC = \int_a^b ROC d_x. \tag{58}$$

Let A_k denote the area under the ROC for algorithm k. The degree of worthiness of algorithm k in context i can then be computed using

$$w^i = \frac{(A_k^i)^m}{\sum_{j=1}^{L}(A_j^i)^m} \tag{59}$$

F Rank-Based Degree of Worthiness

Based on the cumulative distribution function, we can arrange the confidence values in a non-decreasing order, such that:

$$X_{1:N} \leq X_{2:N} \leq \cdots X_{N:N} \tag{60}$$

The i^{th} element $X_{i:N}$ is the i^{th} value in this progression. Then, the cumulative distribution function for the first and last order can be obtained by noting [118]:

$$\begin{aligned} F_{X_{N:N}}(x) &= P(X_{N:N} \leq x) \\ &= \Pi_{i=1}^{N} P(X_{i:N} \leq x) \\ &= [F_X(x)]^N \end{aligned} \tag{61}$$

and:

$$\begin{aligned} F_{X_{1:N}}(x) &= P(X_{1:N} \leq x) \\ &= 1 - P(X_{1:N} \geq x) \\ &= 1 - \Pi_{i=1}^{N} P(X_{i:N} \geq x) \\ &= 1 - \Pi_{i=1}^{N} (1 - P(X_{i:N} \leq x)) \\ &= 1 - [1 - F_X(x)]^N \end{aligned} \tag{62}$$

For the i^{th} element, the probability function of $X_{i:N}$ is given by [119]:

$$f_{X_{i:N}(x)} = \frac{N!}{(i-1)!(N-i)!} [F_X(x)]^{i-1} [1 - F_X(x)]^{N-i} f_X(x) \tag{63}$$

However, obtaining the expected value of a function of x using Eq. (63) is not always possible. Rank-based method that can be derived from Borda Count [6] is a good alternative. The Borda count is a single-winner election method in which voters rank candidates according to candidates' preference. Usually, voters give each candidate a certain number of points corresponding to the position in which the candidate is ranked. Once all votes have been counted the candidate with the most points is the winner. Because it elects broadly acceptable candidates, the Borda count is often described as a consensus-based electoral system, rather than a majority one, then it can be used for fusing different classifiers. Borda Count method has been used for non-linear combiners successfully [6], but rather than treat each class as a candidate in [6], we can treat each classifier as a candidate, and training data as voters.

Let $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iN}\}$ be a set of training data decision for algorithm i, with Borda Count method, we have:

$$X_i \mapsto R(X_i) = sort(x_{i1}, x_{i2}, \ldots, x_{iN}) \tag{64}$$

The highest decision is ranked as 1, the lowest one equals to $1/N$ and the k^{th} is k/N. This ordering transforms the decision space into a non-decreasing ordered sequence.

In our CDF which has L algorithms, and K classes, in each context c_s, let $\boldsymbol{w}_k^{c_s}$ be a vector of real numbers constrained as Eq. (53). The weight $w_i^{c_s}$ can be estimated by Borda Count rank as:

$$\tilde{w}_i^{c_s} = \frac{\sum_{j=1}^{N} R(x_{ij}^k) \cdot 1(.)}{\sum_{j=1}^{N} 1(.)} - \frac{\sum_{j=1}^{N} R(x_{ij}^m) \cdot \bar{1}(.)}{\sum_{j=1}^{N} \bar{1}(.)} \tag{65}$$

where $1(.)$ and $\bar{1}(.)$ are indicators defined in Chapter IV.C. Eq. (65) attempts to separate different classes as far as possible. This is similar to the Separation-Based method mentioned in Chapter IV.C. The algorithm with a higher degree of separation will be assigned a higher weight.

For the linear combination, we can assign a overall confidence value to an unknown patter $\{x_1, x_2, \cdots, x_L\}$ using:

$$g(\mathbf{x}) = \sum_{i=1}^{L} w_i^{c_s} R^{c_s}(x_i) \tag{66}$$

G Cumulative Separation-Based Method

The different between Separation-Based method and Cumulative Separation-based method is that, for the former, we only consider the cMPP to each class for each training data, while for the later, we consider the cMPP to all class for each training data (refer to Figure 20) [111].

Using the cumulative distribution of the two classes of an algorithm i, we define the cumulative separation as:

$$\tilde{w}_i^{c_s} = \alpha \sum_{j=1}^{N} (p_{ij}^k - p_{ij}^m) 1(.) + \beta \sum_{j=1}^{N} (p_{ij}^m - p_{ij}^k) \bar{1}(.) \tag{67}$$

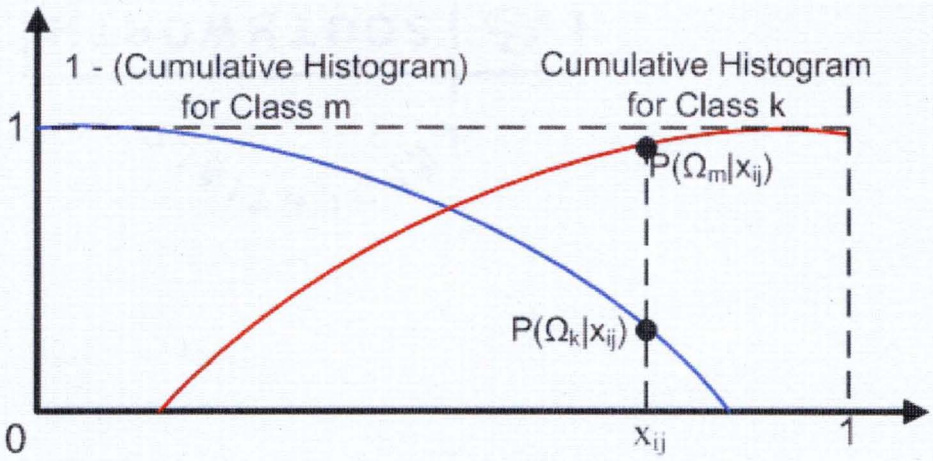


Figure 20. Cumulative Separation-based method. Red curve is one class confidence cumulative histogram, blue curve is another class inverse confidence cumulative histogram.

where $p(\Omega_k|x_{ij})$ is abbreviated as p_{ij}^k, and $p(\Omega_m|x_{ij})$ is abbreviated as p_{ij}^m, $k \neq m$, $1(.) = 1(x_{ij} \in c_s \bigcap x_{ij} \in \Omega_k)$ and $\bar{1}(.) = 1(x_{ij} \in c_s \bigcap x_{ij} \in \Omega_m)$ are indicators. α, β can be computed using:

$$\alpha = \frac{\sum_{j=1}^{N} (x_{ij} \in c_s) \cap (x_{ij} \in \Omega_k)}{\sum_{j=1}^{N} x_{ij} \in c_s} \tag{68}$$

and

$$\beta = \frac{\sum_{j=1}^{N} (x_{ij} \in c_s) \cap (x_{ij} \in \Omega_m)}{\sum_{j=1}^{N} x_{ij} \in c_s} \tag{69}$$

To satisfy Eq. (53), we normalize the weight in Eq. (67) and obtain:

$$w_i^{c_s} = \frac{\tilde{w}_i^{c_s}}{\sum_{i=1}^{N} \tilde{w}_i^{c_s}} \tag{70}$$

H MCE/GPD Based Method

In the last two decades, the minimum classification error (MCE) with the generalized probabilistic descent (GPD) method has been successfully used in pattern recognition and speech recognition tasks. This method is constructed on a direct relation between the system performance measure and the model parameters, and can conduct effective training

model even without any prior knowledge of the data distributions. The MCE/GPD learning routine can be summarized by a four-step process [120, 112]:

Step 1). Define a discriminant function $g_k(\mathbf{x}; \Lambda)$ to discriminate a data sample $\mathbf{x}$ of class Ω_k from other classes; Λ is a set of classifier parameters which will be update in step 4, and k is the given class label of $\mathbf{x}$.

Step 2). Specify the misclassification measure, $d_k(\mathbf{x})$;

Step 3). Construct the minimum error objective function, or loss function $l_k(\mathbf{x}; \Lambda)$;

Step 4). Use a GPD method to estimate and update the model parameters Λ.

The first step determines the formulation of the objective function and is the foundation of the MCE framework. The second step evaluates all training samples, and enumerates the decision rule. The third step defines a loss function which is typically a translated sigmoid function as it is smooth and suitable for gradient based optimization. The last step uses a GPD method to update the parameters used in step 1. This method converges, with probability equal to 1, to Λ^* which is at least a local minimum of $l(\Lambda)$, i.e.:

$$\Lambda(t+1) = \Lambda(t) + \delta\Lambda(x, \Omega_k, \Lambda) \tag{71}$$

$$\delta\Lambda(x, \Omega_k, \Lambda) = -\varepsilon\nabla f(\Lambda) \tag{72}$$

where δ and ε are small positive real number.

Within our CDF framework, suppose that we have C contexts, D algorithms, and K classes. Within each context, C_s, let $g_k(\mathbf{x}; \Lambda_k)$ be a discriminant function, where $\mathbf{x} = (x_1, ..., x_D)$ is D-dimensional confidence feature space, and Λ_k $(k = 1, ..., K)$ denotes a set of parameters of the discriminant function. The discriminant function can be any reasonable type of measure, such as distance or similarity or probability function.

In general, it is hard to estimate the true posterior probability for real data which lack a functional form. This is particularly true for our CDF since within each context, the training data is insufficient. Moreover, the context is constructed via the feature space, and we want to estimate the posterior probability in the decision space. Thus, it cannot be

assumed that the data have a Gaussian or a Uniform distribution, and a mapped posterior probability (MPP) function, $g_k(\mathbf{x}; \Lambda_k)$ would be a good alternative.

First, we define the misclassification measure, $d_k(\mathbf{x})$ which can be described as [112]:

$$d_k(\mathbf{x}) = -g_k(\mathbf{x}; \Lambda_k) + \ln \Big[\frac{1}{K-1} \sum_{c,c \neq k} exp[g_c(\mathbf{x}; \Lambda_k)\eta]\Big]^{\frac{1}{\eta}}, \tag{73}$$

where η is a positive number. A positive $d_k(\mathbf{x})$ implies misclassification while a negative $d_k(\mathbf{x})$ implies correct decision. Thus, the decision rule becomes a judgement on a scalar value. As $\eta \rightarrow \infty$, the term in the brackets reduces to $\max_{i,i \neq k} g_i(\mathbf{x}; \Lambda_i)$, and:

$$d_k(\mathbf{x}) = -g_k(\mathbf{x}; \Lambda_k) + \max_{c,c \neq k} g_c(\mathbf{x}; \Lambda_c) \tag{74}$$

Next, we define the loss function, $l_k(\mathbf{x}; \Lambda_k)$ as a smooth function of the misclassification measure $d_k(\mathbf{x})$. That is,

$$l_k(\mathbf{x}; \Lambda_k) = l(d_k(\mathbf{x})) \tag{75}$$

where l is a translated sigmoid function:

$$l(d_k) = \frac{1}{1 + exp(-\xi d_k + \theta)}, \;\; \xi > 0, \theta > 0 \tag{76}$$

In (76), ξ is used to control the sensitivity in defining the decision boundary. As ξ increase, the sensitivity increases and fewer training patterns can dominate the shape and location of the boundary. The optimal value of ξ could be learned using a regularization method [121]. In our work, we derive the necessary condition when ξ is fixed to a positive value larger than 1.

Using the loss function for a set of training samples $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, we define the empirical average cost function as:

$$L(\Lambda|X) = \frac{1}{N} \sum_{j=1}^{N} \sum_{k=1}^{K} l(d_k(\mathbf{x}_j)) 1(\mathbf{x} \in \Omega_k) \tag{77}$$

where $1(.)$ is the indicator function. Here, we use a batch search type which aims to minimize the average empirical loss. An adaptive search that minimizes the individual loss for a set of samples can also be used. However, this can make the optimization more sensitive to the training samples.

In order to ensure that the estimated weights satisfy the constraint in (53), we map these parameters using:

$$w_{ik} \rightarrow \tilde{w}_{ik} = \ln w_{ik} \tag{78}$$

Then, the parameters are updated with GPD w.r.t to $\tilde{\Lambda}$ in Step 4. After updating, the weights are mapped back using:

$$w_{ik} = \frac{exp(\tilde{w}_{ik})}{\sum_{i=1}^{D} exp(\tilde{w}_{ik})} \tag{79}$$

Using the update rules in (71) and (72), it can be shown that the weights need to be updated using:

$$\tilde{w}_{ik}(t+1) = \tilde{w}_{ik}(t) - \epsilon \frac{\partial L(\tilde{\Lambda})}{\partial \tilde{w}_{ik}}|_{\tilde{\Lambda}=\tilde{\Lambda}(t)} \tag{80}$$

where

$$\frac{\partial L(\tilde{\Lambda})}{\partial \tilde{w}_{ik}} = \sum_{j=1}^{N} \sum_{k=1}^{K} \sum_{i=1}^{D} \frac{\partial l_k(\mathbf{x}_j;\Lambda)}{\partial d_m(\mathbf{x}_j)} \frac{\partial d_m(\mathbf{x}_j)}{\partial g_k(\mathbf{x}_j;\Lambda)} \frac{\partial g_k(\mathbf{x}_j;\Lambda)}{\partial \tilde{w}_{ik}} \tag{81}$$

In the above, the partial derivatives are defined as:

$$\frac{\partial l_k(\mathbf{x}_j;\Lambda)}{\partial d_m(\mathbf{x}_j)} = \xi l_k(\mathbf{x}_j;\Lambda)(1 - l_k(\mathbf{x}_j;\Lambda)) \tag{82}$$

$$\frac{\partial d_m(\mathbf{x}_j)}{\partial g_k(\mathbf{x}_j;\Lambda)} == \begin{cases} -1 & \text{if } m = k \\ \frac{exp[g_m(\mathbf{x}_j;\Lambda)\eta]}{\sum_{c,c\neq k} exp[g_c(\mathbf{x}_j;\Lambda)\eta]} & \text{if } m \neq k \end{cases} \tag{83}$$

and

$$\frac{\partial g_k(\mathbf{x}_j;\Lambda)}{\partial \tilde{w}_{ik}} = p(\Omega_k|x_i) \tag{84}$$

In the above, $x_i \subset \mathbf{x}_j$, i.e. x_i is the i^{th} scalar value in the j^{th} training vector $\mathbf{x}_j$, and $i = 1, \cdots D, j = 1, \cdots N$.

To test a new alarm, first it is processed by each of the individual algorithms, and the extracted features are used to assign the alarm to the closed context. Finally, the confidence value of the individual algorithms and their degree of worthiness in the cluster of assigned are aggregated to compute the final confidence value. Assuming that alarm **x** is assigned to context c, its aggregated confidence value is computed using:

$$Conf(\mathbf{x}) = \sum_{i=1}^{L} w_{iM}^{(c)} * p(\Omega_M|x_i) - \sum_{i=1}^{L} w_{iF}^{(c)} * p(\Omega_F|x_i) \tag{85}$$

where $p(\Omega_M|x_i)$ is the cMPP confidence value assigned by algorithm k.

I Application to Benchmark Data

In order to illustrate the performance of CDF, we set up two experiments on the Phoneme benchmark data. The first experiment uses the same classifier method, but different subsets of features. The second one uses the same features but different classifiers. The Phoneme data is a benchmark data in the ELENA database [122] which was used in the European ROARS ESPRIT project aimed to the implementation of a real time analytical system for French speech recognition [123]. It consists of 5404 five-dimensional vectors (the amplitudes of the five first harmonics AH_i, normalized by the total energy Ene (integrated on all the frequencies) AH_i/Ene, three observation moments) characterizing two classes of phonemes: 1) nasals (70.65%) and 2) orals (29.35%).

1 Experimental Setup

We perform a five-fold cross validation with randomly splitting the data into a training set and a testing set for each cross validation. The training set is used to train the individual classifiers and learn the aggregation weights for each one. The proposed CDF approach requires partitioning the feature space into clusters of similar training samples. To achieve this task, we use the $SCAD_c$ clustering algorithm [117]. We do not address the problem of finding the optimal number of clusters, and simply fix this number to 20. We

choose local fusion with ROC area based, Cumulative Separation based and MCE based methods to illustrate the performance of the proposed CDF.

In the first experiment, we fix the type of classifiers to K-NN and vary the features. In particular, we train six K-NN classifiers using different subsets of features. Each subset includes 3 of the 5 original features (i.e. feature [1, 2, 3], [1, 2, 4], [1, 2, 5], [2, 3, 4], [2, 3, 5], and [3, 4, 5]). The outputs of the six classifiers are combined using the proposed local fusion methods where different weight sets are learned for each cluster using six local weighting approaches. For comparison purposes, we also combine the output of the six classifiers globally using a weighted average fusion (i.e., without partitioning the training data into clusters), and additionally, we score one K-NN classifier using all 5 features.

In the second experiment, we use all 5 features and vary the classification strategy. We train classifiers using a K-NN ($K = 20$); a linear discriminant analysis (LDA); a quadratic discriminant analysis (QDA); and a multilayer linear perceptron (MLP) neural network with 1 hidden layer and 10 nodes. All of the above classifiers generate a soft confidence value. As in the previous experiment, we compare the results when these classifiers are combined globally and locally using the proposed CDF with six local weighting approaches.

2 Experimental Results

The results of the first experiment are shown in Figure 21 where we display the Receiver Operation Characteristic (ROC) curves. These curves display the probability of true positive versus the probability of false positive. As it can be seen, the performance of the K-NN with different subsets of features varies, implying that these subsets encode different information. Also, the global fusion of all six K-NN classifiers outperforms the K-NN classifier with all features. This behavior is consistent with the results reported in the literature [118], that the fusion of "weak" classifier usually outperforms a single classifier.

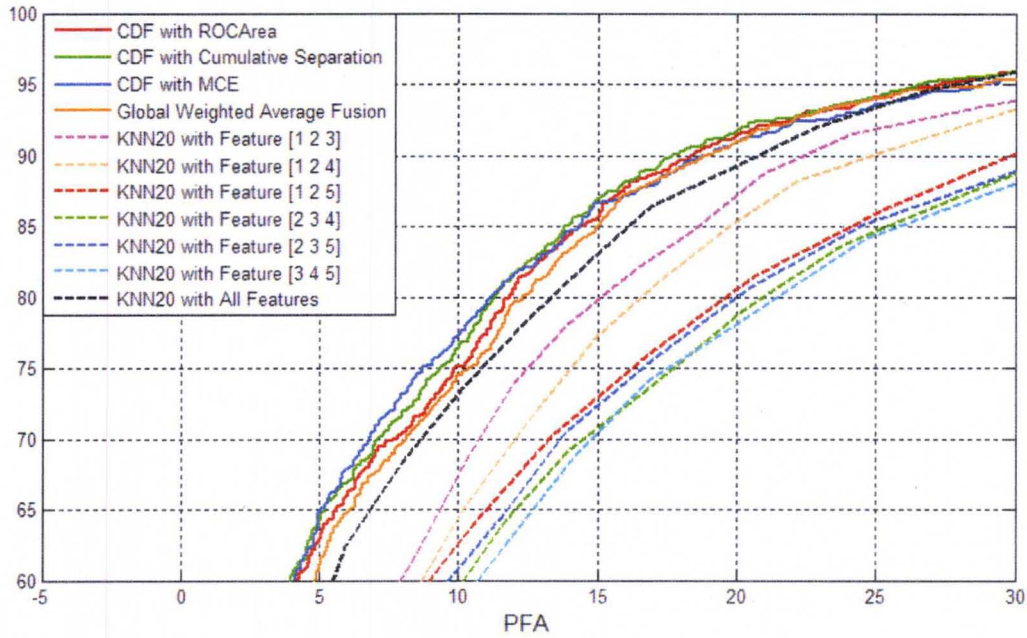
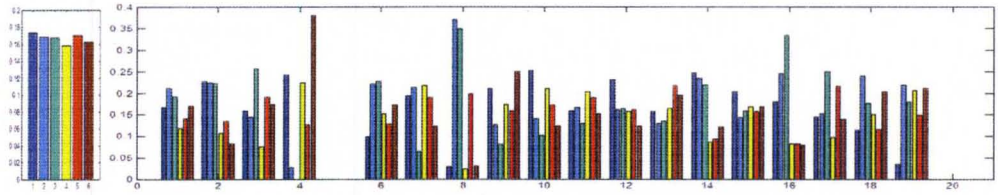


Figure 21. Comparison of the Context-Dependent Fusion with individual K-NN classifiers and global fusion methods

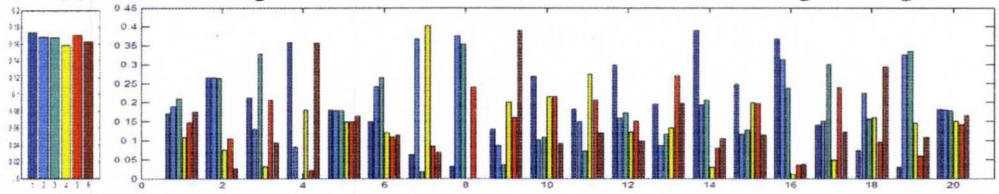
More importantly, almost all the performances of the proposed six local fusions outperform the global fusion. In other words, better results could be achieved when the different classifiers are combined in different ways according to the different contexts extracted from the training data.

Figure 22 shows the global weights and local weights for CDF with the selected methods mentioned above. For better view, we chose the global weights range as [0, 0.2], and local weights range according to the maximum value. From this figure, we can also find that for the local fusion methods, the weights vary significantly, and that makes the local fusion performance better than the global fusion. For CDF with MCE/GPD, the weights assigned to each class as shown in Figure 22 (d).

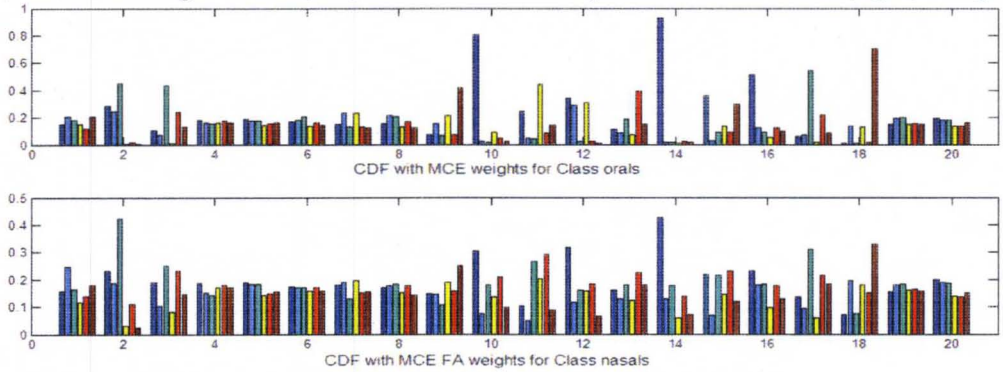
The results of the second experiment are shown in Figure 23. As it can be seen, the performance of the different classifiers can vary significantly. We should point out here that no attempt was made to optimize the performance of any classifier. We simply used the default settings for each one. Our goal is to compare the global and local fusion and



(a) Global average fusion (left) and ROC area based CDF (right) weights



(c) Global average fusion (left) and cumulative separation based CDF (right) weights



(d) MCE/GPD based Context-Dependent Fusion weights

Figure 22. Global fusion and Context-Dependent Fusion (CDF) weighs on Phoneme data with KNN classifier on various feature sets

not the individual classifiers. The global fusion of all classifiers outperforms all individual classifiers, and the performance of the proposed local fusion has the best overall performance.

The global weights and local weights for CDF with selected methods are shown in Figure 24. From this figure, we can also note that in the MCE based local fusion method, the weights to the QDA and the LDA algorithm are smaller because the performances of this two algorithms are quite worse than the other algorithms.

From Figure 21 and Figure 23, we observe that both local and global fusion methods do not improve the results significantly. This is because the different algorithms use the same source of information and the same features. This is not the case for the next chapter's

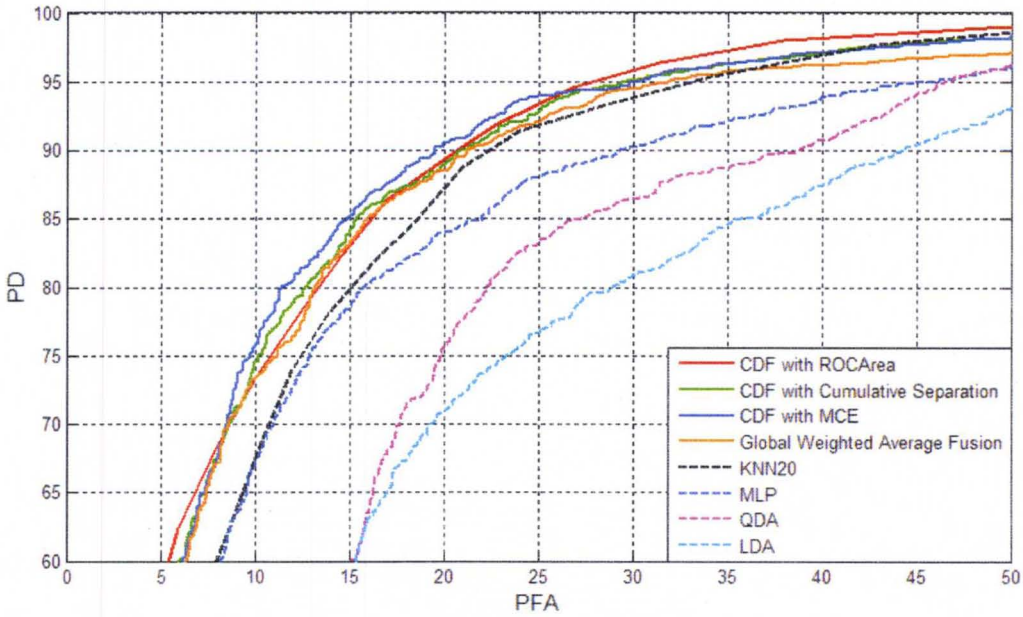
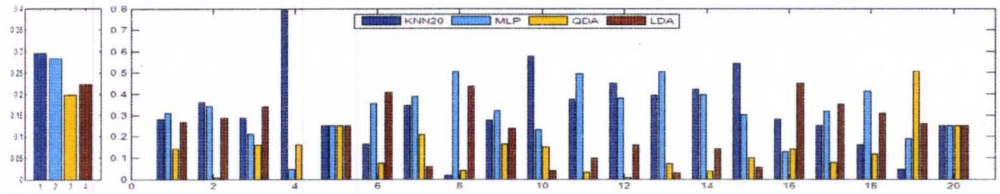
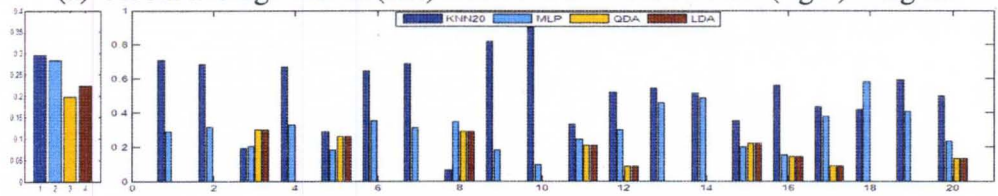


Figure 23. Comparison of the Context-Dependent Fusion with individual classifiers and global fusion

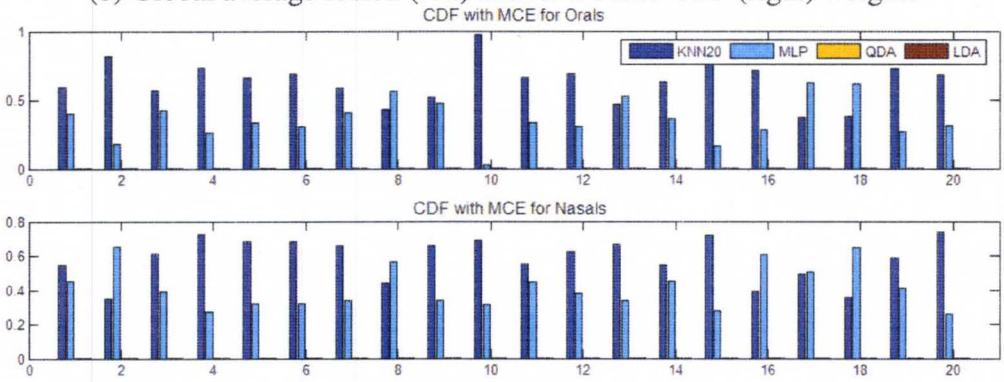
experiments on the landmine detection which involve data from different sensors, and each classifier preprocesses the data differently, and each has its own set of features.



(a) Global average fusion (left) and ROC area based CDF (right) weights



(b) Global average fusion (left) and rank based CDF (right) weights



(d) MCE/GPD based CDF weights

Figure 24. Global fusion and Context-Dependent Fusion (CDF) weighs on Phoneme data with KNN classifier on various feature sets.

CHAPTER VI

APPLICATIONS TO LANDMINE DETECTION

In this Chapter, we apply the proposed Context-Dependent Fusion (CDF) method to the problem of land mine detection. We fuse the output of multiple detection algorithms which have different preprocessing, different features, and different classification approaches. In these experiments, we apply the CDF method with six weighting methods to multiple data sets collected by Vehicle Mounted Ground Penetrating Radar (VMGPR), Autonomous Mine Detection Multi Sensors System with GPR and Wideband Electromagnetic Induction (WEMI), and Airborne Hyperspectral Imagery (AHI) systems to illustrate the performance of our proposed fusion methods. Moreover, in the first experiment within the VMGPR, in addition to evaluate the performance of CDF with a set of fusion methods, we are also interested in their suitability and scalability with respect to the number of discrimination algorithms. Thus, we compare these fuison methods when 5, 6, and 8 discrimination algorithms are considered.

A Experiment 1: Land Mine Detection Using a Vehicle Mounted GPR System

The GPR data (refer to Chapter III.D.1) used in this experiment consists of a sequence of raw GPR signatures collected using a NIITEK Vehicle Mount GPR landmine (VMGPR) system [98] as it travels forward. Figure 25 shows this VMGPR system. This system comprises a 24-channel GPR array which is put in front of the vehicle. Adjacent channels are spaced approximately 5 centimeters apart in the cross-track direction, and sequences (or scans) are taken at approximately 5 centimeter down-track intervals.

Figure 25. Niitek vehicle-mounted GPR system

Since the set of potential false alarm locations is infinite (limited only by the precision of the marking system), we cannot consider typical receiver operating characteristic (ROC) curves comparing probability of detection (PD) vs. probability of false alarm (PFA) because the denominator in the PFA calculation is not well defined. For this reason, algorithm scores are typically reported in PD vs. false alarm rate (FAR) where false alarm rates are measured in number of false alarms per meter squared.

1 Data Collection

The dataset used in this experiments was collected between November 2002 and July 2006 from 4 geographically distinct test sites. We will refer to this collection as NTK4. Sites A, B, and D are temperate climate test facilities with prepared soil and gravel lanes. Site C is an arid climate test facility with prepared soil lanes. Site B has the largest number

of collections and the largest number of alarms. The four sites have a total of 17 different lanes with known mine locations. Most lanes at these sites have both metal and non-metal non-emplaced clutter objects. All mines are Anti-Tank (AT) mines. Overall, there are 19 distinct mine types that can be classified into 3 categories: anti-tank (ATM), anti-tank with low metal content (ATLM), and simulated mines (SIM). The targets were buried up to 6 inches deep. Multiple data collections were performed at each site at different dates, covering a ground area of $41,807.57/m^2$, resulting in a large and diverse collection of mine and false alarm signatures. False alarms arise as a result of radar signals that present a mine-like character. Such signals are generally said to be a result of clutter. In this experiment, clutter arises from two different processes. One type of clutter is emplaced and surveyed in an effort to test the robustness of the algorithms. Other clutter result from human activity unrelated to the data collection or as a result of natural processes. We refer to this second kind of clutter as non-emplaced. Non-emplaced clutter includes objects discarded or lost by humans, soil inconsistencies and voids, stones, roots and other vegetation, as well as remnants of animal activity.

TABLE 2

Statistics of the NTK4 dataset

	Site A	Site B	Site C	Site D	Total
No. Collections	3	6	2	1	12
No. Mine Types	9	15	9	5	19
No. Mine Alarms	183	821	62	494	1560
No. Clutter Encounters	0	15	0	196	211
No. Clutter Alarms post prescreener	0	4	0	46	50
Area(m^2)	14813	15631	4054	7310	41808

The statistics of this collection are shown in Table 2. The data collected from Sites B and D have emplaced buried clutter. Although the lanes at Sites A and C are prepared, they still contain non-emplaced clutter objects. Both metal and non-metal non-emplaced

TABLE 3

Number of Metal and Plastic Cased Mines and Mine Simulants and their burial depths in NTK4 dataset

	Depth								Total
	-1"	0"	1"	2"	3"	4"	5"	6"	
Metal	12	37	124	68	151	34	119	77	777
Low-Metal	6	92	90	204	122	134	47	76	616
Simulants	48	0	20	47	23	29	0	0	167
Total	66	129	234	319	296	197	166	153	1560

clutter objects such as ploughshares, shell casings, and large rocks have been excavated from these sites. The emplaced clutter objects include steel scraps, bolts, soft-drink cans, concrete blocks, plastic bottles, wood blocks, and rocks. In all, there are 12 collections having 19 distinct mine types. Many of these mine types are present at several sites. The prescreener detected $1,560$ of the $1,593$ mines encountered in the data, yielding a 97.9% probability of detection. It rejected 161 of 211 emplaced clutter objects encountered, and yielded a total of $3,435$ false alarms associated with non-emplaced clutter objects. The number, type, and burial depth of the mines are given in table 3. As it can be seen, the mines buried at 1 inch through 6 inches occupy 87.5% of the total targets encountered vs. 12.5% surface-laid or flush-buried mines.

2 Data Pre-processing

Preprocessing is an important step to enhance the mine signatures for detection. In general, preprocessing includes ground-level alignment and signal and noise background removal. First, we identify the location of the ground bounce as the signal's peak and align the multiple signals with respect to their peaks. This alignment is necessary because the vehicle-mounted system cannot maintain the radar antenna at a fixed distance above the ground. The early time samples of each signal, up to few samples beyond the

ground bounce are discarded. The remaining signal samples are divided into N depth bins, and each bin would be processed independently. The reason for this segmentation is to compensate for the high contrast between the responses from deeply buried and shallow anomalies.

Next, the adaptive least mean squares (LMS) pre-screener proposed by Torrione et al. [124] is used to focus attention and identify regions with subsurface anomalies. The goal of a pre-screener algorithm in the framework of vehicle-mounted real-time landmine detection is to flag locations of interest utilizing a computationally inexpensive algorithm so that more advanced feature-processing approaches can be applied only on the small subsets of data flagged by the pre-screener. The LMS is applied to the energy at each depth bin and assigns a confidence value to each point in the cross-track, down-track plane based on its contrast with a neighboring region. The components that satisfy empirically pre-determined conditions are considered as potential targets. Their cross-track x_s, and down-track y_s positions of the connected component center are reported as alarm positions for further processing by the feature-based discrimination algorithm to attempt to separate mine targets from naturally occurring clutter.

3 Evaluation Methods

To provide an objective and consistent evaluation of the different algorithms and their fusion, we use a Testing/training Unified Framework (TUF) system. This system supports creation of supervised learning algorithms that perform discrimination between targets and non-targets in data collected at a variety of different regions (mine or lanes) in a variety of different sites. The framework employs algorithms implemented in Matlab using a control flow that incorporates a user-programmed pre-screener which processes raw data files into Alarms with associated Universal Transverse Mercator (UTM) coordinates and confidence values. The alarms are then processed by extracting signatures. These

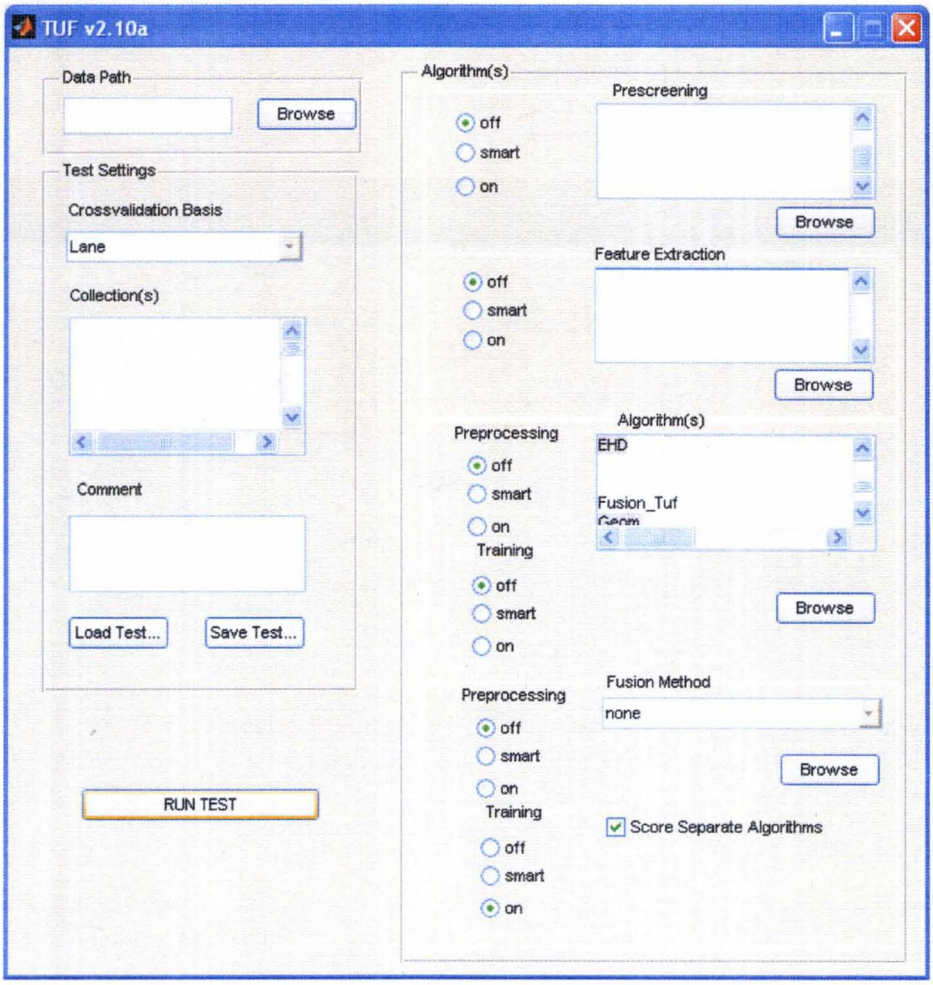


Figure 26. Interface of the TUF evolution system

signatures are passed to a user-specified feature extractor. The features resulting from the feature extractor are presented along with the alarms to a discrimination algorithm, which produced a confidence for each alarm. TUF system performs n-way cross-validation testing using either lane-based cross-validation (in which each mine lane is in turn treated as a test set with the rest of the lanes used for training) or site-based cross-validation (in which each data collection site is treated in turn as a test set). A snapshot of the TUF GUI system is shown in Figure 26.

The results of this process are scored using the Mine Detection Assessment and

Scoring (MIDAS) system developed by the Institute for Defense Analysis [125]. The scoring is performed in terms of Probability of Detection (PD) vs. False Alarm Rate (FAR). Confidence values are threshold at different levels to produce ROC curve. For a given threshold, a mine is detected if there is an alarm within 0.25 meters from the edge of the mine with confidence value above the threshold. Given a threshold, the PD is defined to be the number of mines detected divided by the number of mines. The FAR is defined as the number of false alarms per square meter.

It is often the case that a single dominating classifier (one producing statistically lower FAR at every PD value), does not exist. Furthermore, in many practical cases such as humanitarian demining, the best algorithm may be the one at which 100% detection is achieved with the lowest false alarm rate, no matter what other properties the ROC may display. For other time-critical demining applications where some level of missed mines is not considered as great a cost, the best ROC may be the one at which the probability of detection is highest at a giver constant false alarm rate.

Our algorithm development efforts have been geared toward developing algorithms suitable for an autonomous vehicle-based mine detection system. In such system false alarms will delay the progress of the system. Knowing that any single property of the ROC may be inappropriate to evaluating the algorithms, we have chosen to consider a number of measurable properties of these ROCs. the metrics chosen for algorithm evaluation are the following:

i) PD85: FAR at the threshold yielding PD .85.

ii) PD90: FAR at the threshold yielding PD .90.

iii)PD90: FAR at the threshold yielding PD .95.

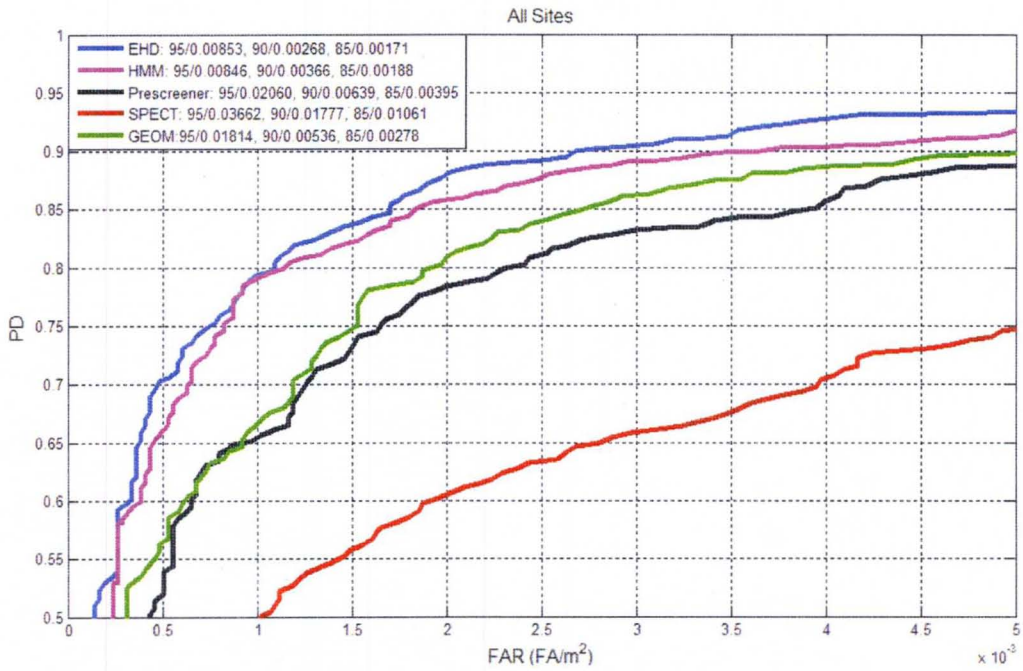


Figure 27. Algorithm ROCs for All Sites

4 Experimental Results

In this first experiment, we utilized five detection algorithms (HMM, EHD, GEOM, SPECT and Prescreener) outlined in Chapter III.D which were implemented and tested within the TUF system. The GEOM and EHD algorithms are trained in this cross-validation manner. The HMM was based on a model trained using a different radar system [93] and SPECT employs a single static mine model that does not require training.

First, we compare the performance of the individual detectors and justify the need to fuse their results to improve the overall performance of the system. Figure 27 displays the ROC's obtained by applying these five detection algorithms to the entire NTK4 VMGPR data collection with lane-based cross validation. As it can be seen, the EHD detector has the best overall performance. However, this does not necessarily mean that the EHD is consistently the best algorithm. For instance, Figure 29 displays the results averaged over site B only of NTK4. For this subset, the HMM is the best algorithm and the EHD is the

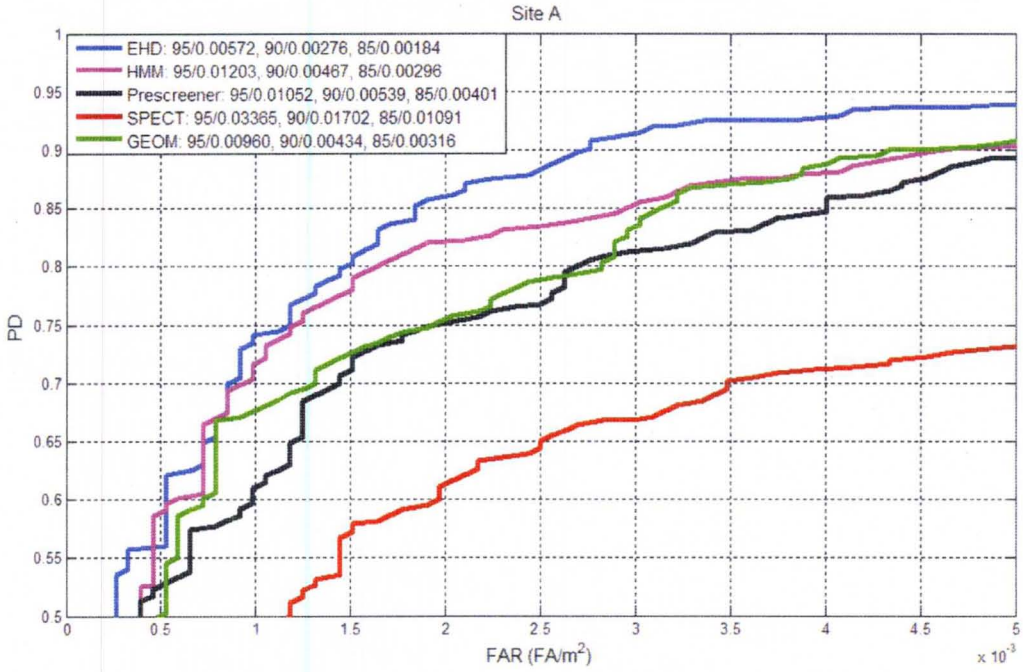


Figure 28. Algorithm ROCs for Site A.

second best one. Our evaluation also shows that the two edge-based algorithms, EHD and HMM provided the best overall performance in the range of detection probabilities of interest in our entire multi-site data collection. At a 90% probability of detection, the difference FAR GEOM (0.00536) is roughly double that of EHD (0.00268). The EHD algorithm was somewhat more consistent in achieving high rankings with respect to our evaluation criteria, however, the performance of the algorithms varied from site to site. In particular, the EHD algorithm outperformed HMM at Site A, while HMM performed better at Site B. Consulting Figure 29, we see that at Site B, the HMM algorithm has a large number of false alarms from lower PDs than EHD.

Figure 30 shows a 2-D scatter plot of the EHD and HMM confidence values for the NTK4 data. Comparing Figure 30 (a) and (b), we observe that in (a), there are more mines (Red points) that have higher confidence value for the EHD than the HMM. On the other hand, in Figure 30 (b), there are more mines (Red points) that have higher confidence value

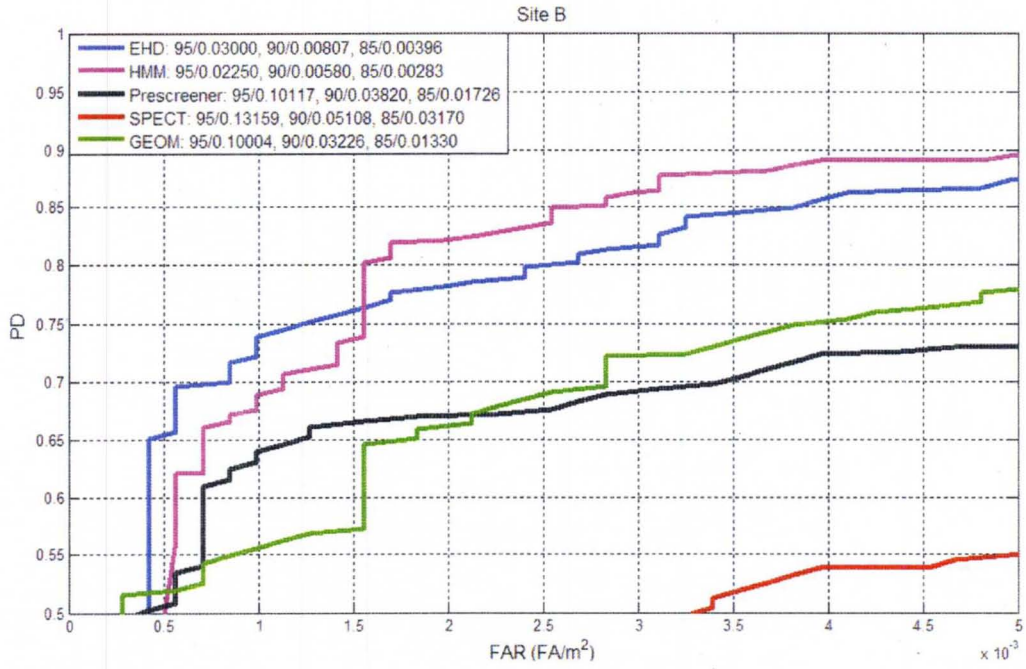
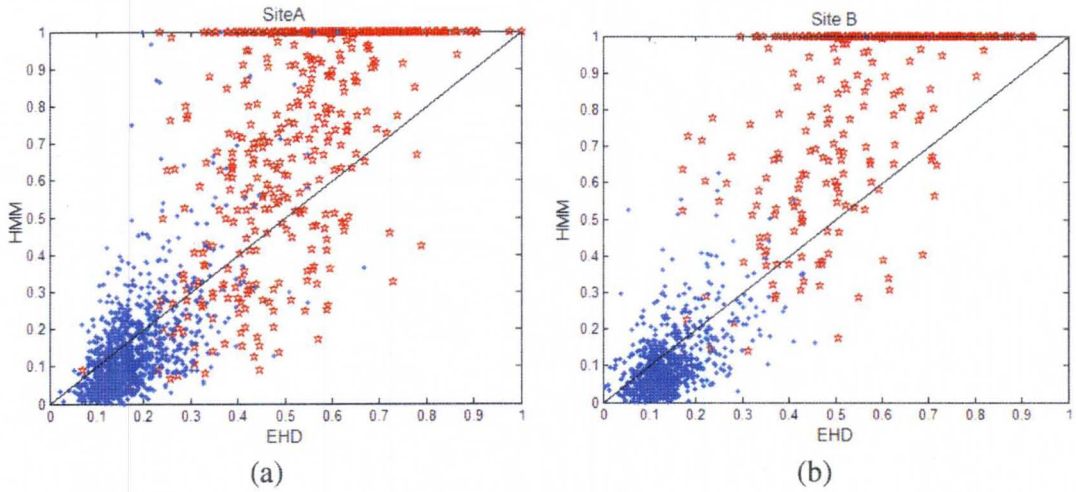


Figure 29. Algorithm ROCs for Site B.



(a) (b)

Figure 30. HMM and EHD Confidence value scatter plot for NTK4, Red stars are Mine, and Blue dots are FA. (a) Site A, (b) Site B

for the HMM than the EHD. This means that even within a small subset of the same site, the relative performance can vary significantly.

From the above analysis, we can conclude that when one algorithm performs better on one site, that does not necessarily mean it will perform better on the other sites, and also there is no single algorithm that can consistently outperform all others detectors. This observation has motivated us to adopt the context dependent fusion algorithm to such problem.

In our first experiment, the training data consists of a set of alarms reported by the LMS pre-screener. Each alarm is processed by these four algorithms (EHD, HMM, SPECT and GEOM) outlined in Chapter III.D. The features extracted from these alarms by the different algorithms are combined, and the $SCAD_c$ (refer to Chapter IV) clustering algorithm is used to partition the training signature into groups of similar signatures, or contexts, and learn the relevant features within each context. In our experiment, the EHD algorithm has 40 features, the HMM has 20 features, the SPECT has 20 features, and the GEOM has 12 features. Thus, in total we have a 92 dimension feature space to be clustered by the $SCAD_c$ algorithm into 20 clusters. Within each cluster, the five algorithms are scored and a degree of worthiness is assigned to each one based on its relative performance. Algorithms with better performance are considered more "expert" for the cluster under consideration, and will be assigned larger weights. The worthiness of all algorithms are constrained to sum to 1.

Figure 31 displays the distribution of the NTK4 4 collection (A, B, C and D) data in 20 clusters. The first figure is the mine distribution in 20 clusters, and the second is the Fa distribution, while the last figure is the summary of the alarms distribution. Theoretically, all Mines or Fa from the same collection should cluster together. From the first figure in Figure 31, we observe that Site C is clustered into Clusters 1, 2, and 8, and Site A is mainly clustered into Clusters 1, 2, 3 and 7, due to the consistent of the alarm. Site B has the

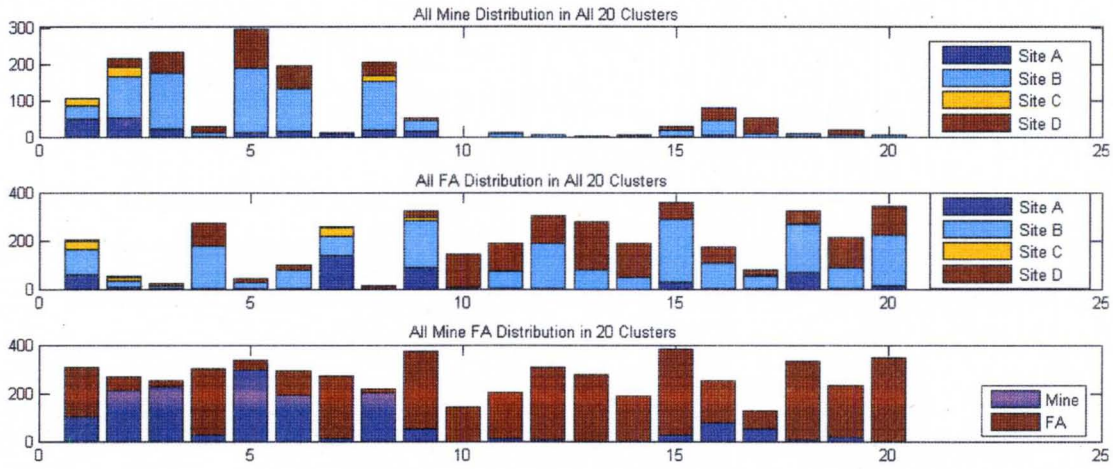


Figure 31. NTK4 data distribution in 20 clusters

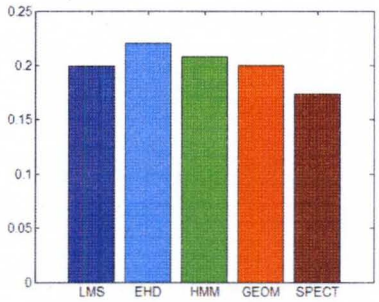


Figure 32. Global fusion weights assigned to the five detections in CV1 for the NTK4 data

largest number of mines and mine types. Thus, it is reasonable for the alarms from this site to be scattered over multiple clusters.

Figure 33 shows the CDF weights for each algorithms with different local weight methods in the first cross validation (CV). For comparison purposes, we also assign a global weight using the entire training collection (i.e. treat all data as one cluster). These weights are shown in Figure 32. As it can be seen, overall, the EHD has the best performance followed by the HMM and then the WEMI. However, the performance of the different algorithms can vary significantly from one context to another as shown in Figure 33.

Figure 34 shows the distribution of all the alarms included in the first CV (CV1).

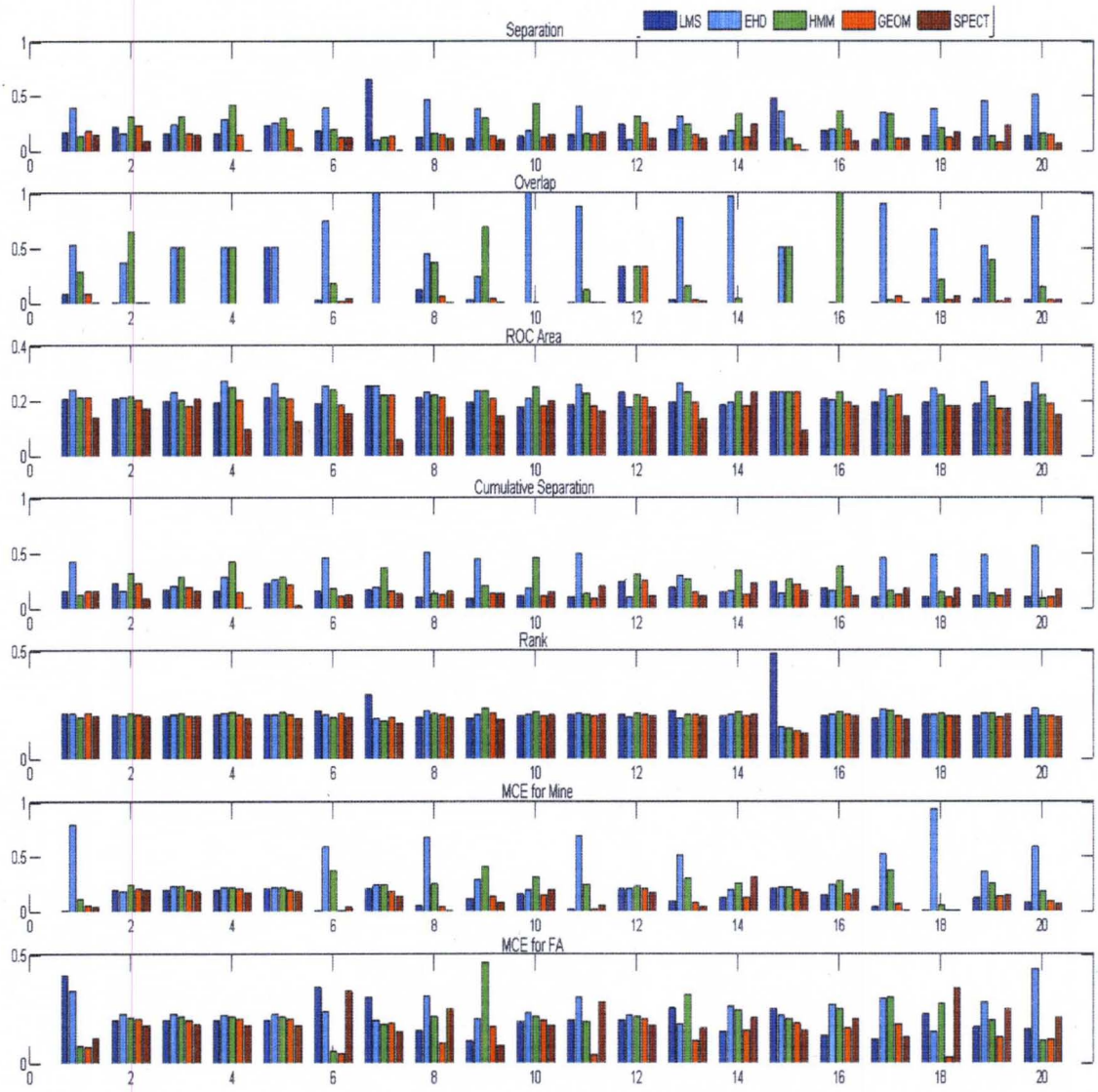
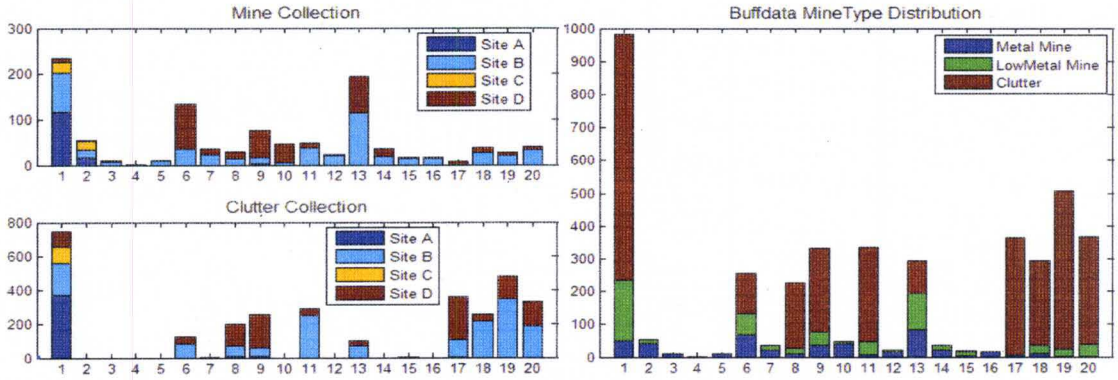


Figure 33. Context-Dependent Fusion weights assigned to the five detections in 20 clusters in CV1 for the NTK4 data



(a) Distribution of the mine and clutter per site (b) Distribution of the alarms per type

Figure 34. Distribution of the alarms included in CV1 for NTK4 data

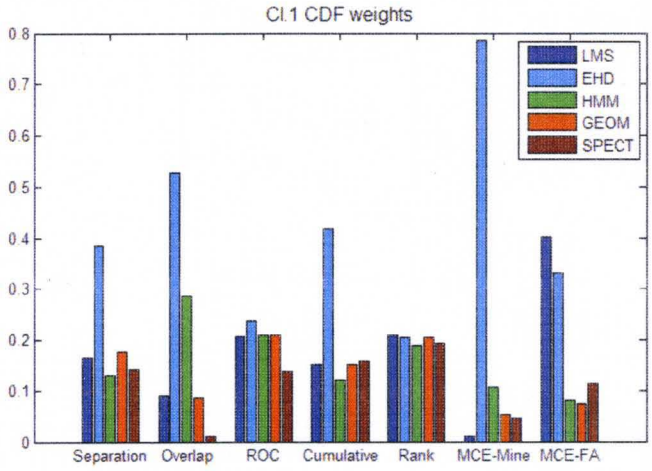
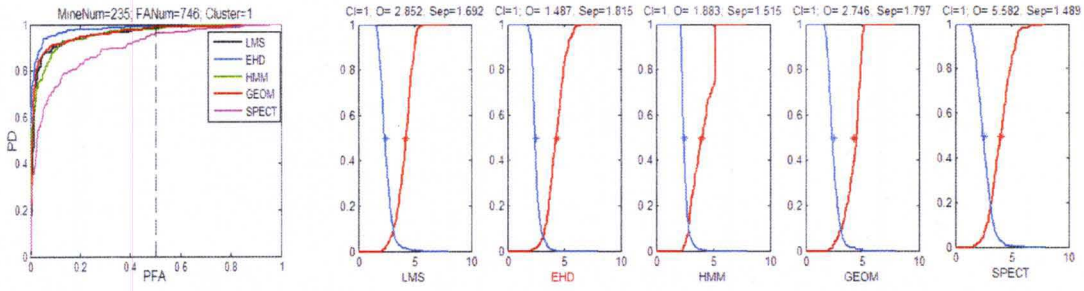


Figure 35. Context-Dependent Fusion weights assigned to the five detections in Cluster 1 in CV1 for NTK4 data



(a) ROC for Cluster 1 in CV1 (b) Separation and Overlap for Cluster 1 in CV1

Figure 36. Local performance of Cluster 1 in CV1 for NTK4 data

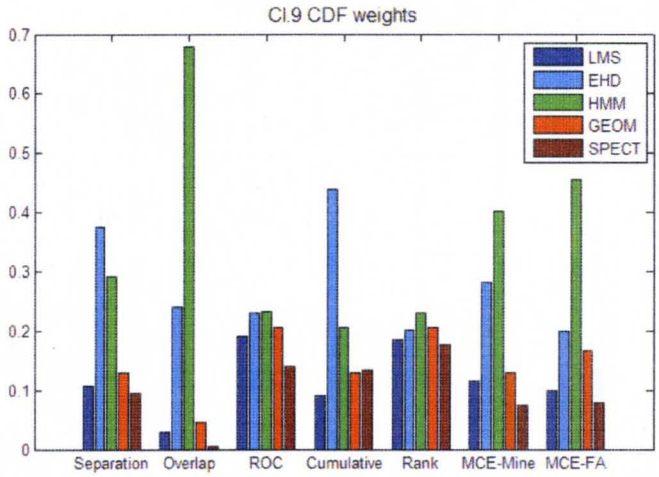
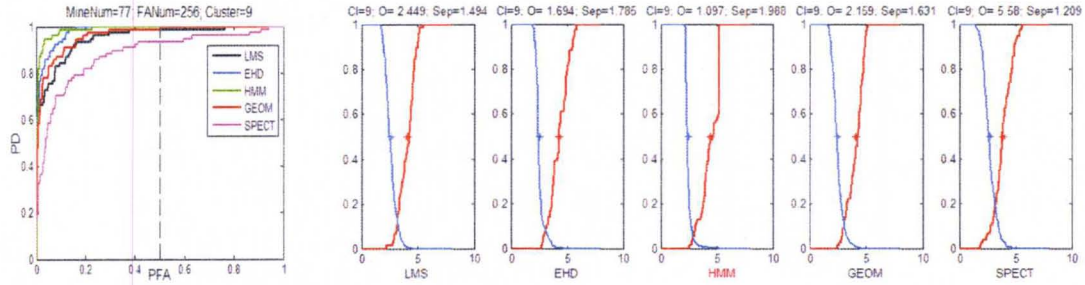


Figure 37. Context-Dependent Fusion weights assigned to the five detections in Cluster 9 in CV1 for the NTK4 data



(a) ROC for Cluster 9 in CV1 (b) Separation and Overlap for Cluster 9 in CV1

Figure 38. Local performance of Cluster 9 in CV1 for NTK4 data

As it can be seen, in this CV, alarms from Site A and C are distributed in cluster 1 and 2, and alarms from Site D are distributed in clusters 6-10, 13, 17-20. Figure 35 displays the weights assigned by CDF to each detection in Context 1. As it can be seen, the EHD is assigned the largest weight based on its performance. To justify the weights assigned by CDF to the algorithms, in Figure 36, we display the distribution and ROCs of the confidence values assigned by each algorithm. At it can be seen, the EHD has the best Separation ($Sep = 0.1815$), the best Overlap ($O = 1.487$), and the best ROC area (Blue curve in Figure 36 (a)) in cluster 1. Thus, the EHD should be considered more "expert" for Cluster

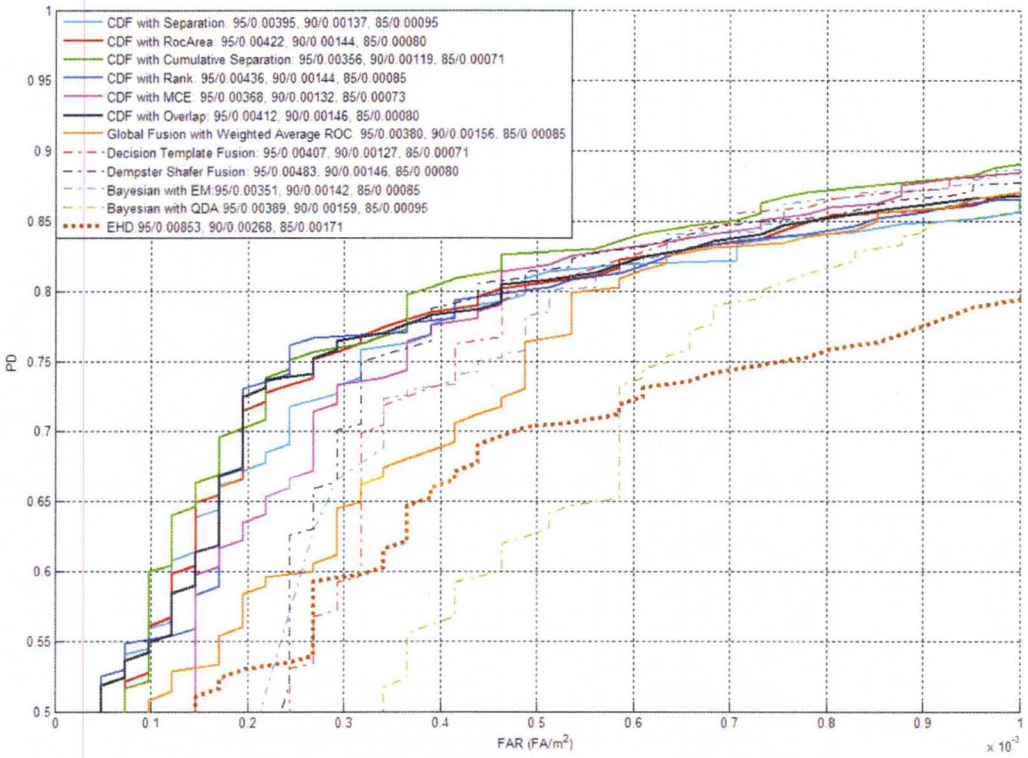


Figure 39. Performance of the Context-Dependent Fusion and the global fusion on the entire collection of the NTK4 data

1, and is assigned the largest weight. Figure 37 shows the CDF weights for Cluster 9, and Figure 38 compares the ROCs, separation, and overlap of the five algorithmsance in this clustet. As it can be seen, for Cluster 9, HMM has the best Separation ($Sep = 1.988$), the best overlap ($O = 1.097$), and the best ROC area (Green curve in (c)). Thus, the HMM is assigned has the highest degree of worthiness in this cluster.

Figure 39 displays the results of the CDF with all six weight assignment methods. We also compare the results with several state-of-the-art fusion methods including: Bayes-based with QDA and EM, Dempster Shafer Theory (DST), Decision Template (DT), and global weighted average fusion based on ROC area. We also include the ROC of the EHD (best overall individual discrimination algorithm) as a reference. As it can be seen, the ROCs of all fusion methods are clustered together, and thus all methods have comparable

performances. All fusion methods improve the PD results over the best discrimination algorithm by an average of 10% for FAR around 0.0007. At low PD ($< 80\%$), the Bayes-base with QDA fusion result is not as good as the other methods. This is due mainly to the fact that one single Gaussian component may not be sufficient to model the distribution of the confidence values of the individual discriminators in the 5-dimensional confidence space. The Bayes-based EM method, which is similar to the QDA based, does not exhibit this behavior because multiple Gaussian components (M was estimated to be 4) were used to model the distribution of each class. It is also interesting to note that the QDA based fusion outperforms EM at higher PD. This is because the former method is optimized to minimize the average FAR for $PD \in [92\%; 96\%]$.

The CDF has the best overall performance. This is due to the fact that this method is local and strives to take advantage of the different detectors in different contexts. For any cluster (or context) the detectors are ranked based on the overlap between the mine and clutter confidence distribution. This ranking can ignore (by assigning low aggregation weights) many of the discrimination algorithms. It could also assign a significant weight to discrimination algorithms that are good for the given context, but globally, are not as good as other algorithms.

5 Scalability with Respect to the Number of Algorithms Fused

This experiment is designed to evaluate the performance of various fusion methods, and test their scalability with respect to the number of discrimination algorithms. In this experiment, we use only one weighting method for CDF, ROC Area based CDF. In addition to the five discrimination algorithms used in the previous experiment (HMM, EHD, GEOM, SPECT and Prescreener), we also use the Texture Feature Classification Method (TFCM) detector [126], the Gaussian Fit (GFIT) detector [127] and the Gaussian Markov Random Field (GMRF) detector [128] highlighted below:

- **TFCM Detector:** The Texture Feature Classification Method (TFCM) detector [126] is a three-dimensional extension of the algorithm by Horng [129]. The algorithm transforms a block of GPR data into a block of integer codes. The code at each point in a block is generated by considering several differences in GPR intensity values over a $3\times3\times3$ window centered at the point. The differences are thresholded producing a string of zeros and ones, which are then mapped to the integer codes, the details of which are described in the references. Statistical textures features, such as entropy, variance, co-occurrence, etc., are then computed on the blocks of codes and transformed into feature vectors. Relevance Vector Machines use the features to produce a confidence that an alarm represents a landmine.

- **GMRF Detector:** The Gaussian Markov Random Field (GMRF) detector [128] is based on a transmission line model of the time-domain GPR response to the subsurface. The model represents the GPR as a sequence of dielectric discontinuities. Each discontinuity is parameterized by a location and a gain parameter. These parameters are characterized statistically using a Gaussian-Markov Random Field. A generalized likelihood ratio test is then used to assign a confidence that an alarm represents an antitank landmine.

- **GFIT Detector:** The Gaussian Fit (GFIT) detector [127] calculates the parameters of a Gaussian pulse which best fits the spatial energy distribution of target responses to GPR. The output features are the goodness of fit, the pulse width, and pulse gain. More specifically, the spatial shape of the summed energy from a cross-track scan is compared to the shape of a Gaussian pulse. If x represents position in down-track scans, and E represents the energy, we find the σ, x_0, α to minimize the root mean square error between $E(x)$ and $f(x)=\alpha * exp(-(x_0 - x)/\sigma 2)$. The output features are then $\sqrt{sum_x(E(x) - f(x))}$, σ, x_0, and α.

The above discrimination algorithms were developed by researchers at the Universities of Missouri, Louisville, Florida, as well as Duke University. They are independently developed and have many differences in GPR Preprocessing and normalization, feature extraction, and classification methodologies. For example, in feature extraction alone one can see many differences. The anomaly prescreener detector simply looks for locations that are different from the background. It uses masks oriented in the C-scan direction. The HMM detector looks at variable length sequences of edges. The EHD detector looks at fixed length representations of edges. All these three algorithms used the down-track and cross-track time domain GPR. The SPECT detector looks at features in the frequency domain. The GEOM detector calculates feature based on geometric shape in C-scans. The TFCM detector looks for texture features in three-dimensional blocks of time domain data, GMRF, and the GFIT detector looks at energy in the cross-track direction. Thus, in the feature extraction process alone, one can see that these algorithms vary widely in the focus and processing. Each of the eight detection algorithms (EHD, HMM, Prescreener, SPECT, GEOM, TFCM, GFIT, and GMRF) and the fusion methods (Context-dependent, Bayes, Decision Template, Dempster-Shafer, and Borda count) were implemented with the TUF system.

First, we compare the performance of the individual detectors and justify the need to fuse their results to improve the overall performance of the system. Figure 40 displays the ROC's obtained by applying the 7 detection algorithms and the prescreener to the entire data collection. As it can be seen, the EHD detector has the best overall performance. However, this does not necessarily mean that the EHD is *consistently* the best algorithm. For instance, Figure 41 displays the results averaged over site A of the collection only. For this subset, the EHD is the best algorithm and the HMM is the second best one. However, in Figure 42, which displays the results averaged over site B only, the HMM is the best algorithm and EHD is the second best one.

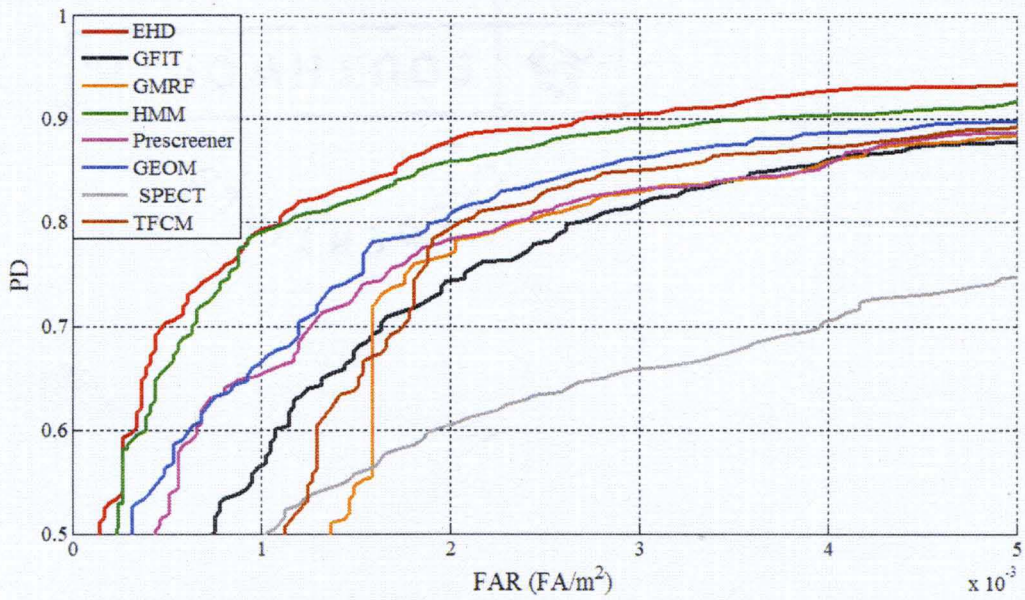


Figure 40. Performance of the 8 different detectors on the entire NTK4 data collection

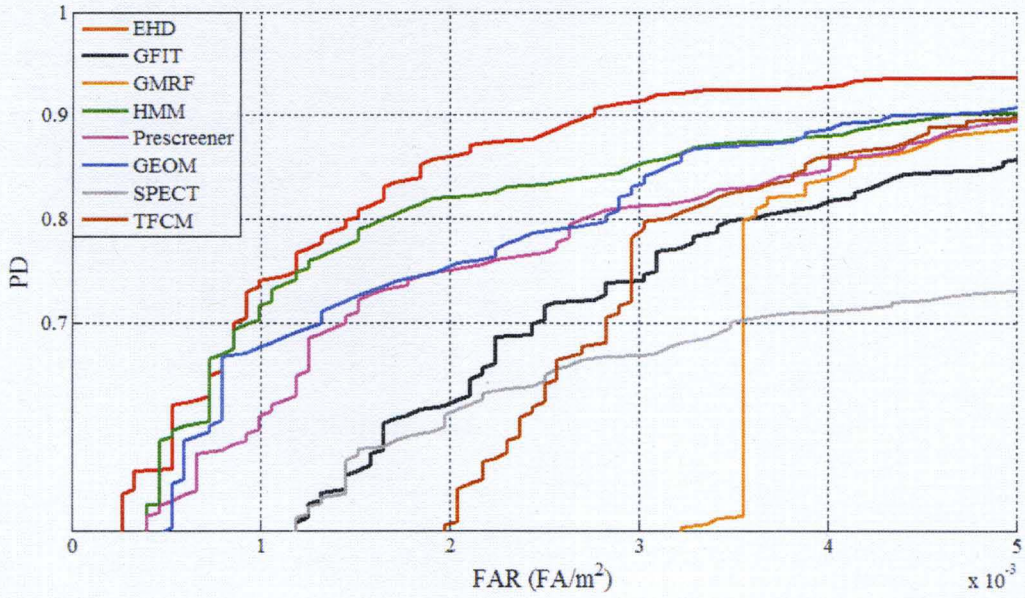


Figure 41. Performance of the 8 detectors on Site A only

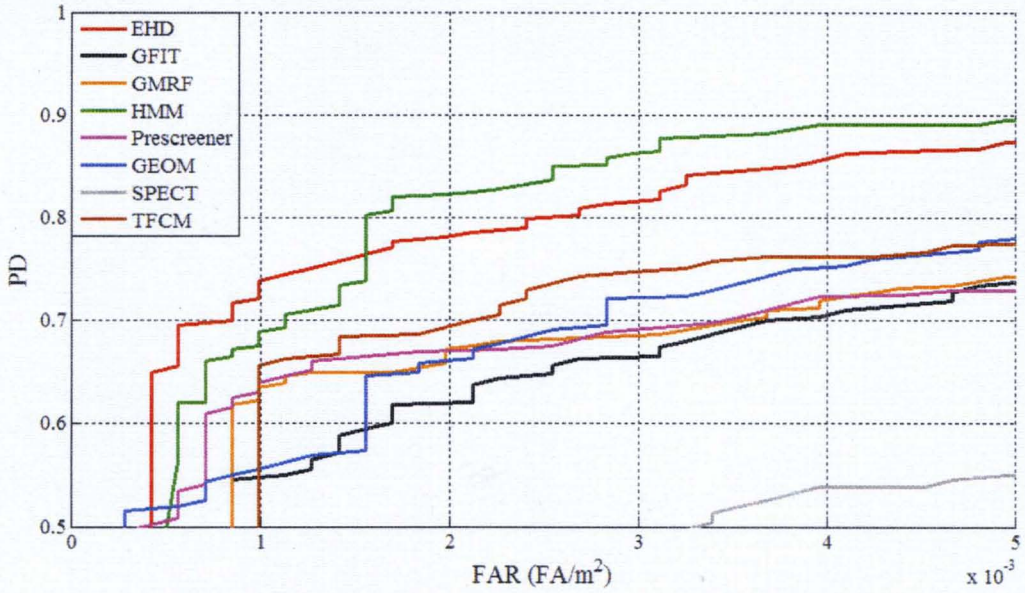


Figure 42. Performance of the 8 detectors on Site B only

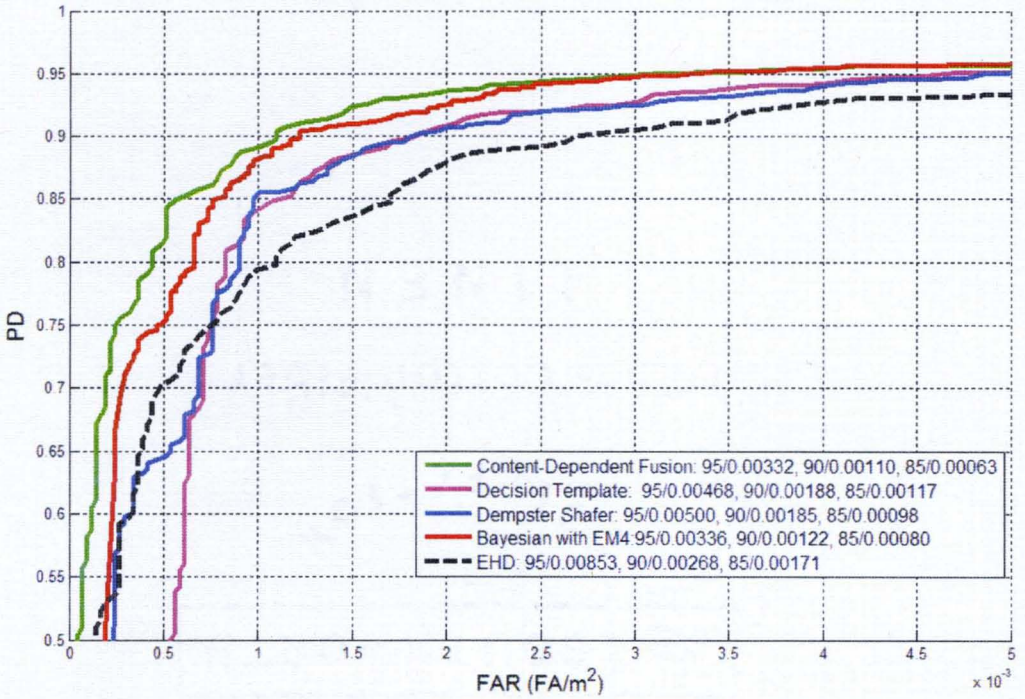


Figure 43. Comparison of 4 fusion methods when 6 discrimination algorithms (EHD, HMM, SPECT, Prescreener, GEOM, and TFCM) are combined

Figure 43 displays the results of the 4 fusion algorithms (CDF, DT, DST, and Bayesian with 4 components EM) when only 6 discrimination algorithms (EHD, HMM, SPECT, Prescreener, GEOM, and TFCM) are fused. For comparison, we also put the best individual detecting algorithm EHD on the figure. First, we notice the addition of the TFCM algorithms did not improve the results of any of the fusion methods. Two possible reasons may explain this behavior. First, the added TFCM algorithms is based on edge, texture, and statitics features that are already used (in a different way) by the other discrimination algorithms. Second, it is possible that for the data collection that was used is not possible to improve the results further.

Comparing the results in Figure 43 to those in Figure 39, we observe that for some fusion methods, the performance has degraded. In particular, the performance of the DST and the DT methods have dropped significantly at low PD ($< 80\%$) and have become even worse than the EHD discriminator. Investigation of this problem has revealed that these two fusion methods generate confidence values that have a distribution close to binary. This behavior is due to the way the basic belief functions are aggregated (refer to Eq. (16)). In particular, adding more algorithms will require more multiplications. For the DT method, the dimension of the decision template matrix increases, and this may drive the distances in Eq. (25) to a bimodal distribution. Due to these nearly binary distributions, weak mines will be assigned confidence values close to zero, and this would explain the lower PD at low FAR. Also, strong false alarms will be assigned confidence values close to 1, and this would explain the relatively lower PD at higher FAR.

Figure 44 compares the results of the 4 fusion algorithms (CDF, DT, DST, and Bayesian with 4 components EM) when 8 discrimination algorithms (EHD, HMM, SPECT, Prescreener, GEOM, TFCM, GFIT, and GMRF) are fused. First, we note that the performance of the DT and DS degraded further as the confidence values become closer to binary. Second, the performance of Bayesian fusion methods has degraded compared to the fusion

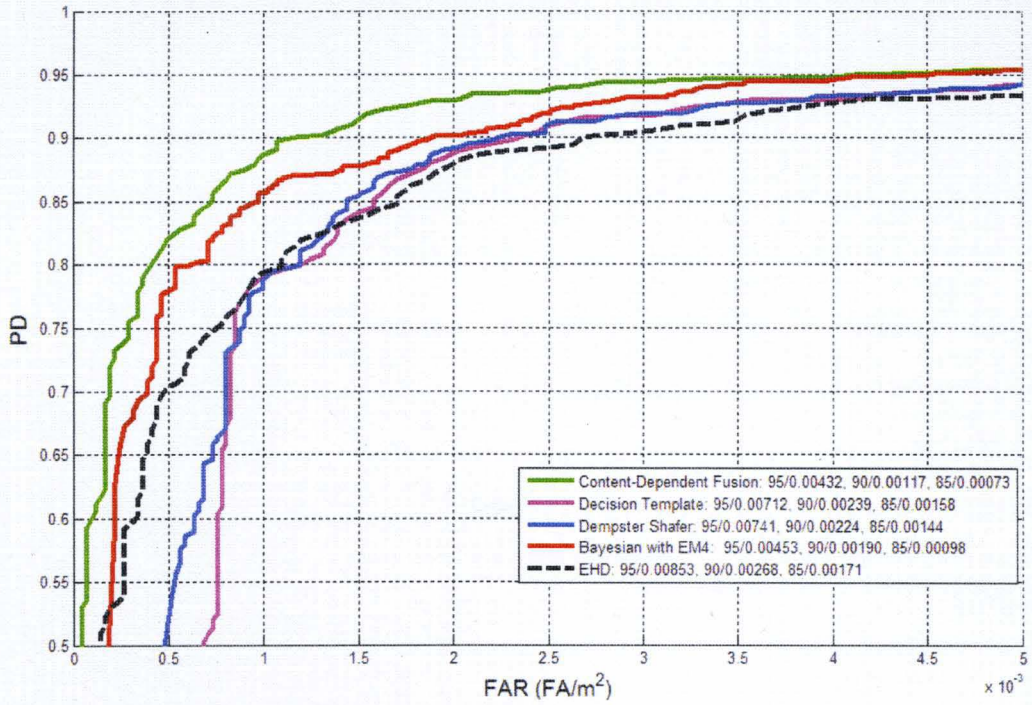


Figure 44. Comparison of 4 fusion methods when 8 discrimination algorithms (EHD, HMM, SPECT, Prescreener, GEOM, TFCM, GFIT, and GMRF) are combined

of 5 algorithms only. This may be due to the fact that the 3 added algorithms have lower performances (refer to Figure 40), and when all 8 algorithms are fused globally, the added algorithms have a negative impact. Third, we note that the dependency assumption does not seem to be an issue. In fact, the best fusion methods (CDF) assume that the eight discrimination algorithms are independent.

The CDF has the best overall performance. Moreover, the addition of discrimination algorithms did not degrade its performance. In fact, for certain FAR values, its performance has improved. Again, this is due to the fact that this method is local and strives to take advantage of the different detectors in different contexts. We have observed that on average, this fusion assigns significant aggregation weights to 3 to 5 discrimination algorithms. These algorithms differ from one cluster to another.

These experimental results also show that although the fusion algorithms were all quite similar when a small number of algorithms were fused, the performance was more

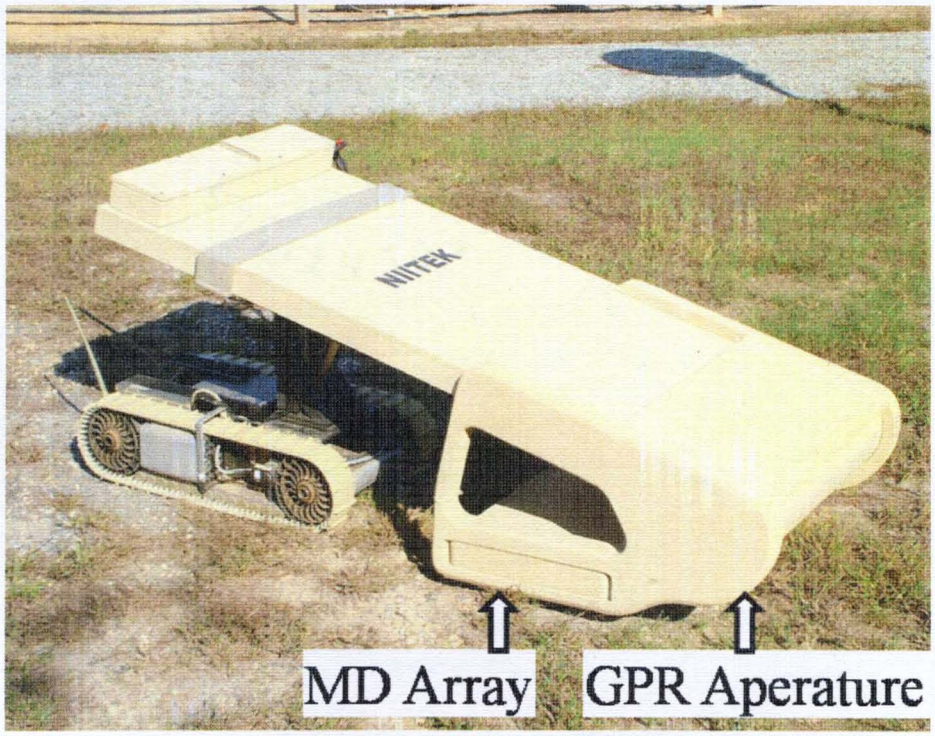


Figure 45. NIITEK Autonomous Mine Detection System

varied as the number of algorithms increased. Context dependent fusion appears to outperform the other methods.

B Land Mine Detection Using an Autonomous Mine Detection System

In this experiment, we use the data collected using NIITEK Inc. robotic mine detection system to illustrate and validate the proposed CDF fusion methods on multiple sensors. This system includes a GPR and a WEMI sensor and is shown in Figure 45. It was used to acquire large collections of co-located GPR and WEMI data from geographically distinct test sites.

We use three distinct detection algorithms that were developed for GPR data (EHD, HMM, and SPCET described in Chapter III.D), and one algorithm for the EMI data [130, 131] described in Chapter III.D6.

1 Data Statistics

The data sets used in our experiment were collected in May 2007 from 2 geographically distinct test sites (site A and site B). These sites have several emplaced mines of various types including Anti-Person (AP) mines, Anti-Tank (AT) mines, High Metal (HM) mines, and Low Metal (LM) mines. The two sites are partitioned into grids with known mine locations. In all, there are 28 distinct mine types that can be classified into 4 categories: anti-tank metal (ATM), anti-tank with low metal content (ATLM), anti-personnel metal (APM), and anti-personnel with low metal content (APLM). These sites also include various clutter objects such as steel scraps, bolts, soft-drink cans, concrete blocks, and wood blocks. The clutter objects are emplaced and surveyed in an effort to test the robustness of the detection algorithms. Mines and clutter objects were buried up to 5 inches deep. This data collection includes a total of 308 mine signatures and 556 False Alarm signatures.

In our data collection, false alarms arise as a result of sensor signals that present a mine-like character. Such signals are generally said to be a result of clutter. Clutter arises from two different processes. One type of clutter is emplaced and surveyed. Objects used for this clutter can be classifier into 2 categories: High Metal Clutter (HMC) and Non-Metal Clutter (NMC). High metal clutter such as steel scraps, bolts, soft-drink cans, is emplaced and surveyed in an effort to test the robustness of the detection algorithms, and in particular the WEMI algorithm. Non-metal clutter such as concrete blocks and wood blocks is emplaced and surveyed in an effort to test the robustness of the GPR based detection algorithms. The other type of clutter, referred to as blank, is caused by disturbing the soil.

Overall, the data collection includes a total of 311 mine signatures and 564 False Alarm (FA) signatures. The statistics of these collections are shown in Table 4. The depth distribution for all objects are shown in Table 5.

TABLE 4

Statistics of the data collection

Type	Content	Site A	Site B	Total/Category	Total/type
AP	HM	16	40	56	187
	LM	38	93	131	
AT	HM	6	20	26	124
	LM	28	70	98	
FA	HMC	224	68	292	564
	NMC	72	68	140	
	BLANK	52	80	132	
Total		436	439	875	875

TABLE 5

Burial Depth of All Objects in the Data Collection

	Mine			Clutter		
Depth	Site A	Site B	Total	Site A	Site B	Total
Surface	0	27	27	52	80	132
(0 1"]	12	104	116	70	46	116
(1 2"]	36	48	84	78	44	122
(2 3"]	28	34	62	88	18	106
(3 4"]	12	0	12	60	20	80
(4 5"]	0	10	10	0	8	8
Total	88	223	311	348	216	564

2 Motivations

The proposed CDF is motivated by the observation that there is no single detection algorithm that can consistently outperform all other detection algorithms for landmine detection. In fact, the relative performance of different algorithms can vary significantly depending on the algorithms adaption, feature types, and sensor styles. Figure 46 shows the individual detection algorithms on all data sites, it can be seen, EHD is the best algorithm for the entire data, while Figure 15 in Chapter IV.A show that the ROC of the four

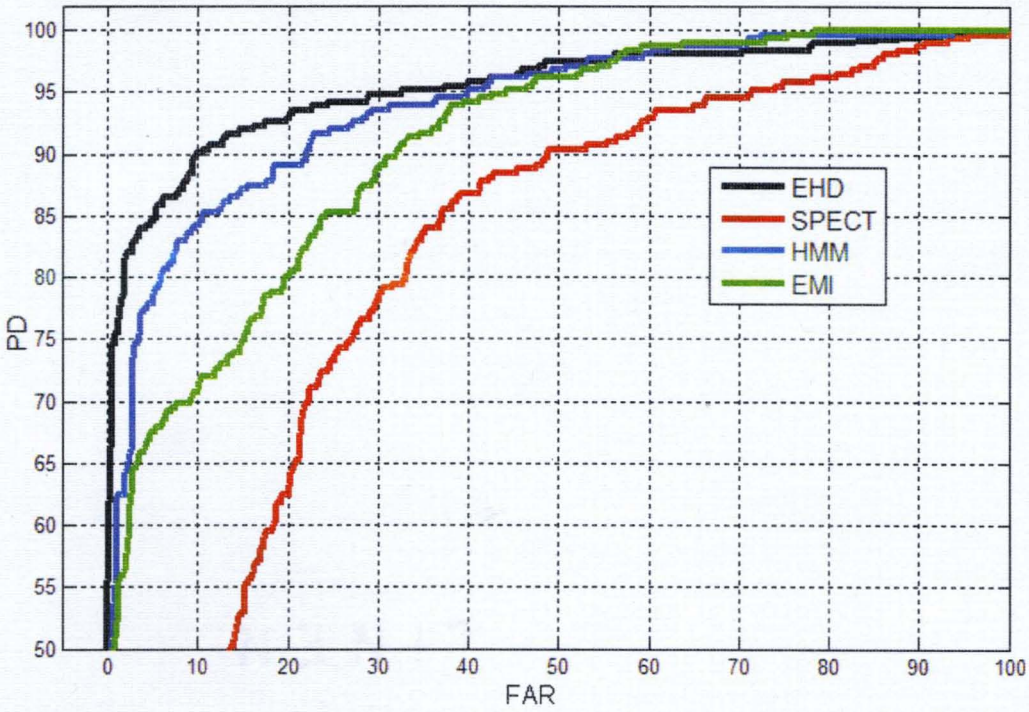


Figure 46. Individual algorithms ROC on all data sites

discrimination algorithms on various subsets of data collected by this NIITEK robotic mine detection system vary to different geographical site, soil type, mine type, bury deepth etc.), and that the CDF should work well on landmine detection because CDF will take advantages of the strengths of few algorithms in different regions of the feature space without being affected by the weaknesses of the other algorithms.

3 Context Extraction

For each cross validation, the training data consists of a set of co-located GPR and WEMI alarms. Each alarm is processed by the four discrimination algorithms (EHD, HMM, SPECT, and WEMI) outlined in Chapter III.D. The features extracted from these alarms are then fed to $SCAD_c$ to partition the aggregated feature space into C clusters. The choice of the number of clusters is not critical for this application. This number should be large enough so that most clusters contain only similar alarms. However, it should not

TABLE 6

Distribution of the alarms among the 10 clusters for one cross validation set

Cluster	ATHM	ATLM	APHM	APLM	HMC	NMC	Blank	Total
1	0	18	0	0	0	20	0	38
2	16	0	11	0	6	0	0	33
3	0	0	12	1	28	0	0	41
4	0	0	5	15	34	0	0	54
5	0	0	0	16	39	0	0	55
6	0	0	0	0	9	31	93	133
7	4	4	24	20	54	0	0	106
8	0	3	0	45	31	1	1	81
9	2	27	0	0	0	1	0	30
10	0	42	0	22	24	51	19	158
Total	22	94	52	119	225	104	113	729

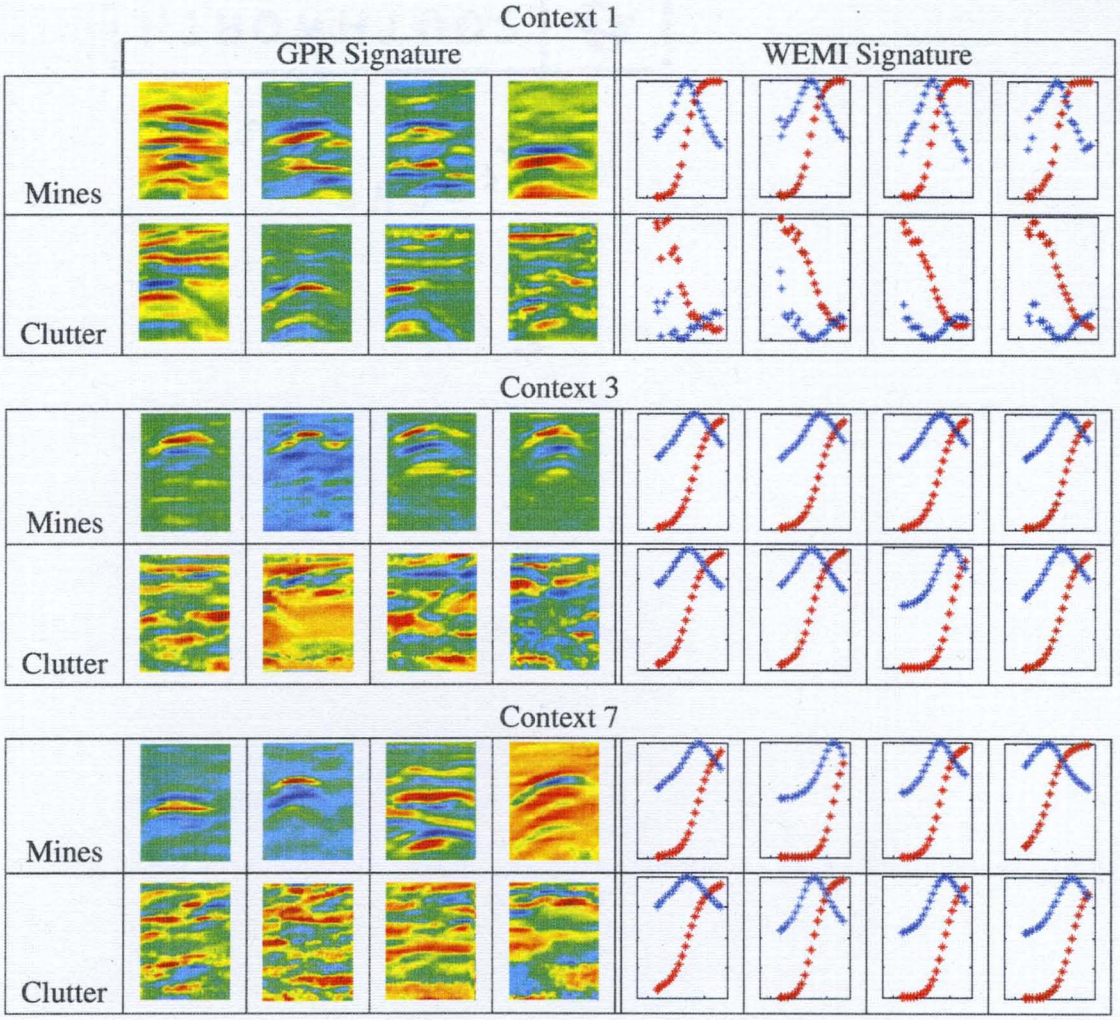
be too large to avoid using too many small clusters that do not include enough samples to learn the optimal algorithm fusion weights. Here, we let $C = 10$.

Table 6 displays the content of the 10 identified clusters. As it can be seen, most clusters include alarms of similar types, and thus may be considered as a homogeneous context. For instance, some clusters are dominated by high metal mines and high metal clutter. Others are dominated by AT mines or AP mines. Also, some clusters include mainly mine or clutter alarms. Others, include a mixture of both. Alarms that are grouped into the same context share common GPR and/or WEMI features.

Table 7 displays few representative mine and clutter alarms from three contexts. For instance, context 1 includes only AT mines with low metal content and non-metal clutter (refer to Table 6). Alarms from both classes have strong GPR signatures, and the GPR sensor by itself may not be sufficient to discriminate between mines and clutter within this context. The WEMI sensor, on the other hand, can easily discriminate between these samples. For context 3, which includes mainly AP mines with high-metal content and high-metal clutter, the GPR sensor is more reliable. For this context, the WEMI sensor cannot

TABLE 7

Samples of representative mine and clutter alarms from three different contexts



discriminate between mines and clutter objects since both have high metal content.

The different contexts do not always correspond to alarms of the same type. If this is the case, the ground truth could be used to partition the training data into contexts. Context 7 is an example of one cluster that includes mine signatures from all 4 types. One alarm from each type is displayed in Table 7. In this case, other factors such as burial depth and soil properties can affect the signatures. For instance, for the GPR sensor, some shallowly buried AP mines can have signatures as strong as the deeply buried AT mines.

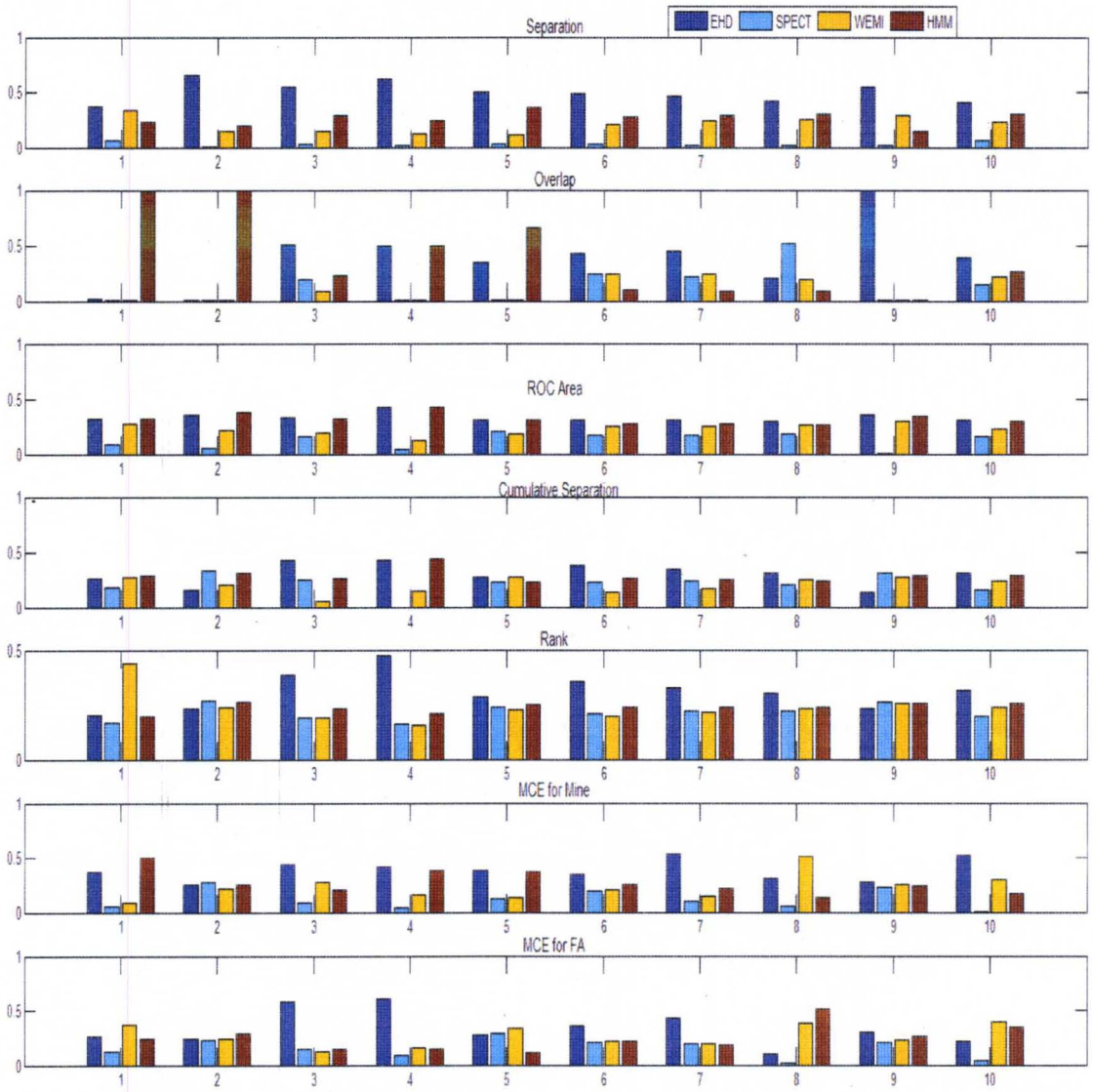


Figure 47. Context-Dependent Fusion weights of CV1 in 10 clusters

4 Learning Detectors Aggregation Weights

After the context extraction step, the performance of each detector is evaluated within each context based on the degree of worthiness proposed in Chapter V between the mine and clutter alarms assigned to it. Then, an aggregation weight is assigned to each detector in each context. These weights are shown in Figure 47. For comparison purposes,

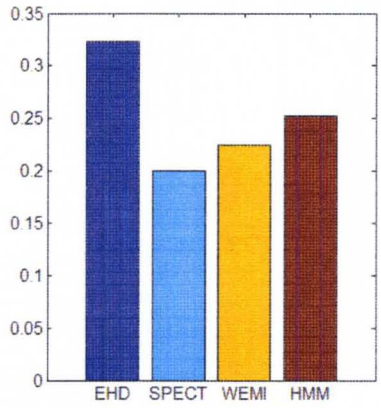


Figure 48. Global weighted average weights of CV1

we also assign a global weight using the entire training collection, i.e. we treat all data as one cluster, and weighted the algorithms according to the ROC area. These weights are shown in Figure 48. As it can be seen, overall, the EHD has the best performance followed by the HMM and then the WEMI. However, the performance of the different algorithms can vary significantly from one context to another. For instance, the WEMI detector has the least weight in context 1 for Rank-based, Cumulative Separation-based and MCE-based methods which consistent with Table 7. This is because this context includes mainly mines and non metal clutter, and the WEMI can discriminate between these objects easily. On the other hand, context 3 (refer to Table 6) includes mainly AP mines with high metal content and clutter with high metal content. From Figure 15(c) we know that the EHD and WEMI outperform the other detectors for HM mines. In particular, the WEMI does a better job at detecting strong mines, but the EHD is better at rejecting the high metal clutter. Thus, a combination of these two algorithms can provide a higher probability of detection at a lower false alarm rate. Context 9 is another interesting one where the SPECT detector was assigned the higher weight. This is despite the fact that, globally, the performance of this detector is not even close to the other detectors. Context 7 is also interesting one where the mines come from all 4 types (refer to Table 7). In this Context, all individual algorithms

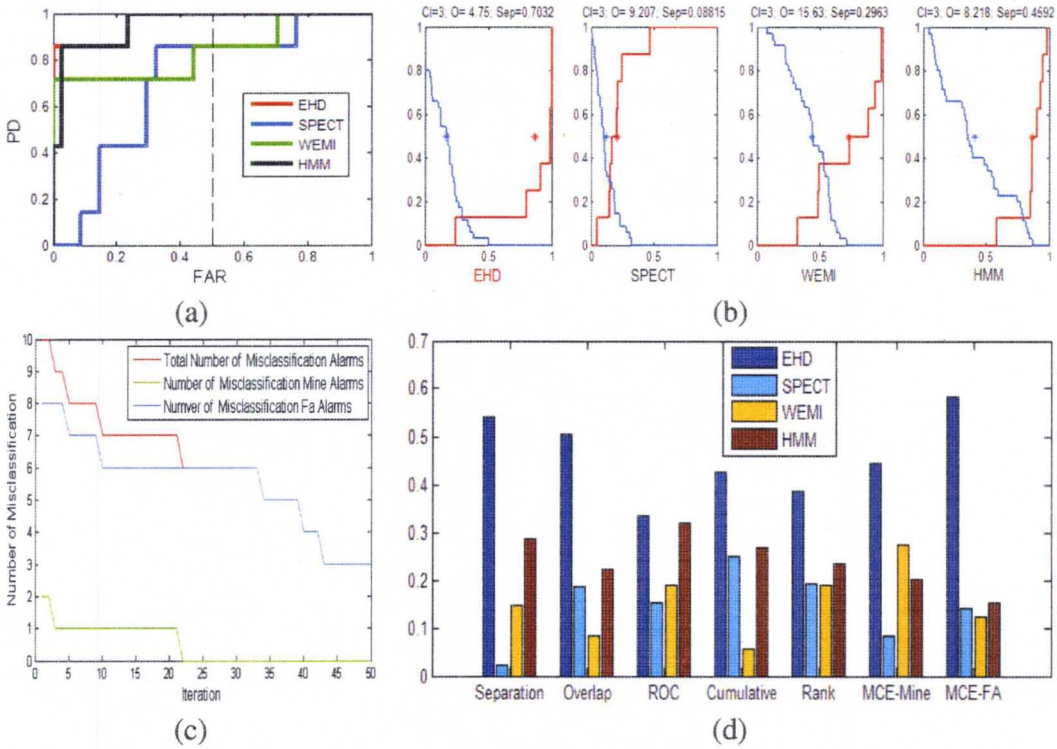


Figure 49. Context-Dependent Fusion performance in Cluster 3. (a) ROC, (b) Separation and overlap, (c) Misclassification in MCE, (d) Context-dependent weighs for all methods in Cluster 3

give a nearly average weights because the mixture mines. Figure 49 attempts to further zoom in to Cluster 3 to check the performance of worthiness of degree for all detection algorithms, (a) displays the ROC for each algorithms in this cluster; (b) shows the separation and overlap, from (a) and (b), we can find that, in this cluster, EHD is the best algorithm which consistent with the analysis of above; (c) displays the change for MCE/GPD misclassification error, as expected, the error should degrade after each update; (d) is the CDF weighs on different performance in this clusters, again, it shows all CDF methods under this cluster will assign a highest weight to EHD.

The above cluster-based fusion weights are intuitive and expected to be helpful as outlined in our motivation example in Chapter IV.A. However, here we want to emphasize that these weights are learned from the training data without user supervision.

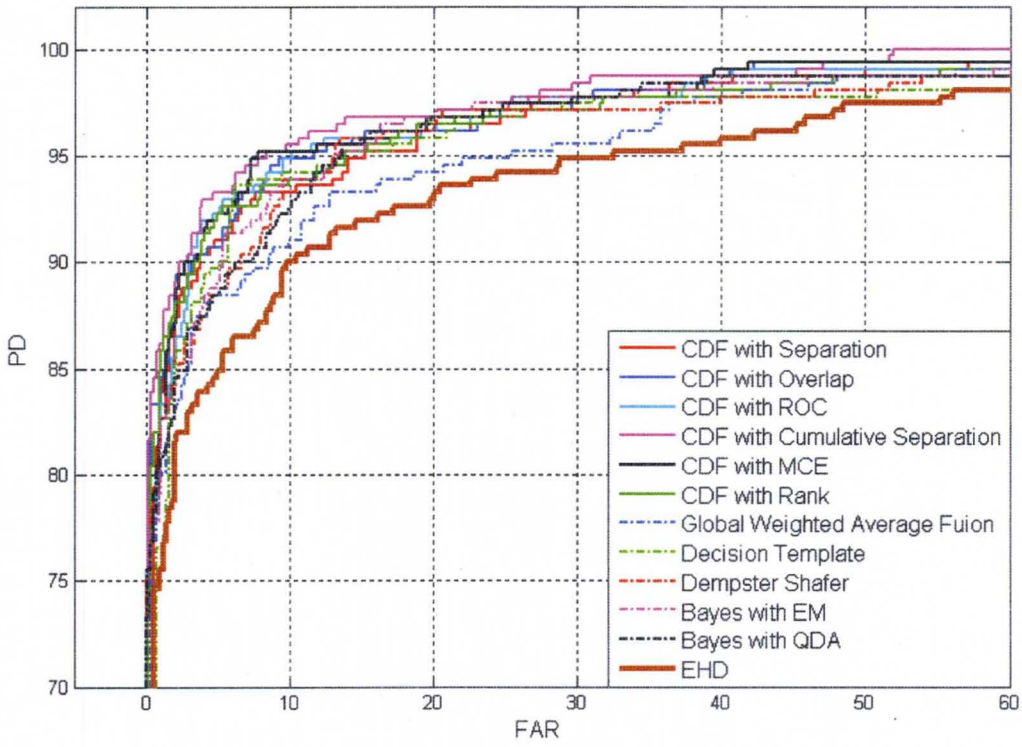


Figure 50. Performance of the individual detectors and the global and local fusion on the entire collection with 6 folds cross validation

5 Analysis of the Testing Phase

The performance (on the testing data) of the four individual detectors using cross-validation within the TUF system is shown in Figure 46. As it can be seen, the EHD has the best overall ROC, followed by the HMM, the WEMI, and then the SPECT. This is consistent with the performance on the training data and the global aggregation weights assigned to these algorithms shown in Figure 48. The ROC's resulting from the global fusion and the proposed context-dependent fusion are also shown in Figure 50. We also include the ROC of the EHD (best overall discrimination algorithm) as a reference. First, we note that even with a simple global fusion (dot dash blue curve in Figure 50), we obtain results that outperform all individual detectors. This is because these detectors operate on different sensor data, use different preprocessing, feature extraction, and classification algorithms. This diversity allows the fusion to take advantages of the strengths of the individual de-

tectors, overcome their weaknesses, and achieve a higher accuracy. Second, the proposed context dependent fusion outperforms all individual detectors and the global fusion significantly. Although the performance of CDF vary because the different worthiness of degree, the ROC's of all CDF fusion methods are clustered together, and thus all methods have comparable performances. For instance, for a 90% PD, the CDF method reduces the FAR by 63% when compared to the global fusion and by 70% when compared to the best individual detector. Similarly, at a 95% PD, the CDF method reduces the FAR by 57% when compared to the global fusion and by 69% when compared to the best individual detector. Third, all fusion methods improve the PD results over the best discrimination algorithm by an average of $10 - 20\%$ for PD around 90%. Additionally, Bayes based fusion results are not as good as the other methods. This is due mainly to the fact that one Gaussian components may not be suitable to model the distribution of the confidence values of the individual discriminators in the 4-dimensional confidence space.

C Land Mine Detection with Airborne Hyperspectral Imagery Data

1 Data Statistics and Experimental Setup

The proposed local fusion CDF methods are also applied to Airborne Hyperspectral Imagery (AHI) data. AHI was flown over an arid site at various times in the years 2002, 2003 and 2005. Data was collected at altitudes of 300m and 600m with spatial resolution of 10 cm and 15 cm respective to altitude. Eight AHI images sets were created which covered approximately $145,000m^2$ of terrain. Each image contains 70 spectral bands after trimming and binning, ranging over long-wave infrared (LWIR) wavelengths 7.88μm-11.02μm. Each image contains millions of spatial pixels, where each pixel consists of 70 spectral signature. From each image, there are three different types of targets (buried mines) in the imagery [132].

Labeled data sets were constructed from the imagery. The well-known RX algo-

rithm [133] was run by Winter et al. [132] on the imagery as a pre-screener (anomaly detector) to reduce the size of imagery and collect Points of Interest (POIs). The pre-screener returned a various number of POIs per image. There are 4,490 POIs and 654 actual targets (buried mines) in the entire data set. Groups of samples surrounding each POI in a 15x15 pixel window were collected to form data sets.

Cross validation was performed on the image level. Each test set was compared to all labeled data sets except for the set that was constructed from the image from which the test set was constructed. This ensures that we do not include any of the test data in the training data during each experiment.

Three individual detection algorithms (RX, Whitening-Dewhitening (WD), and Mixture of Gaussians) were used for fusion. The RX algorithm is a prescreener and requires no training [133]. The WD transform is a classifier which whitens a test image with respect to the statistics of a training image [134]. The mixture of Gaussians is simply a mixture of Gaussians trained on target and background samples, and the confidence for each POI is the likelihood of the target class. The last two detection algorithms were trained/tested, using the same cross validation at the image level.

Features extracted from the POIs are used to partition the data into 10 clusters using $SCAD_c$ algorithm [117]. In each cluster/context, the CDF methods were used to assign worthiness to each algorithm.

To test an alarm, its features, extracted from the POIs, are used to assign it to the closed context. Then, the confidence value of the individual algorithms and their degree of worthiness in the cluster of interest are aggregated to compute the final confidence value.

2 Experimental Results and Analysis

After the context extraction, the performance of each detector is evaluated within each context based on the degree of worthiness proposed in Chapter V. Then, an aggre-

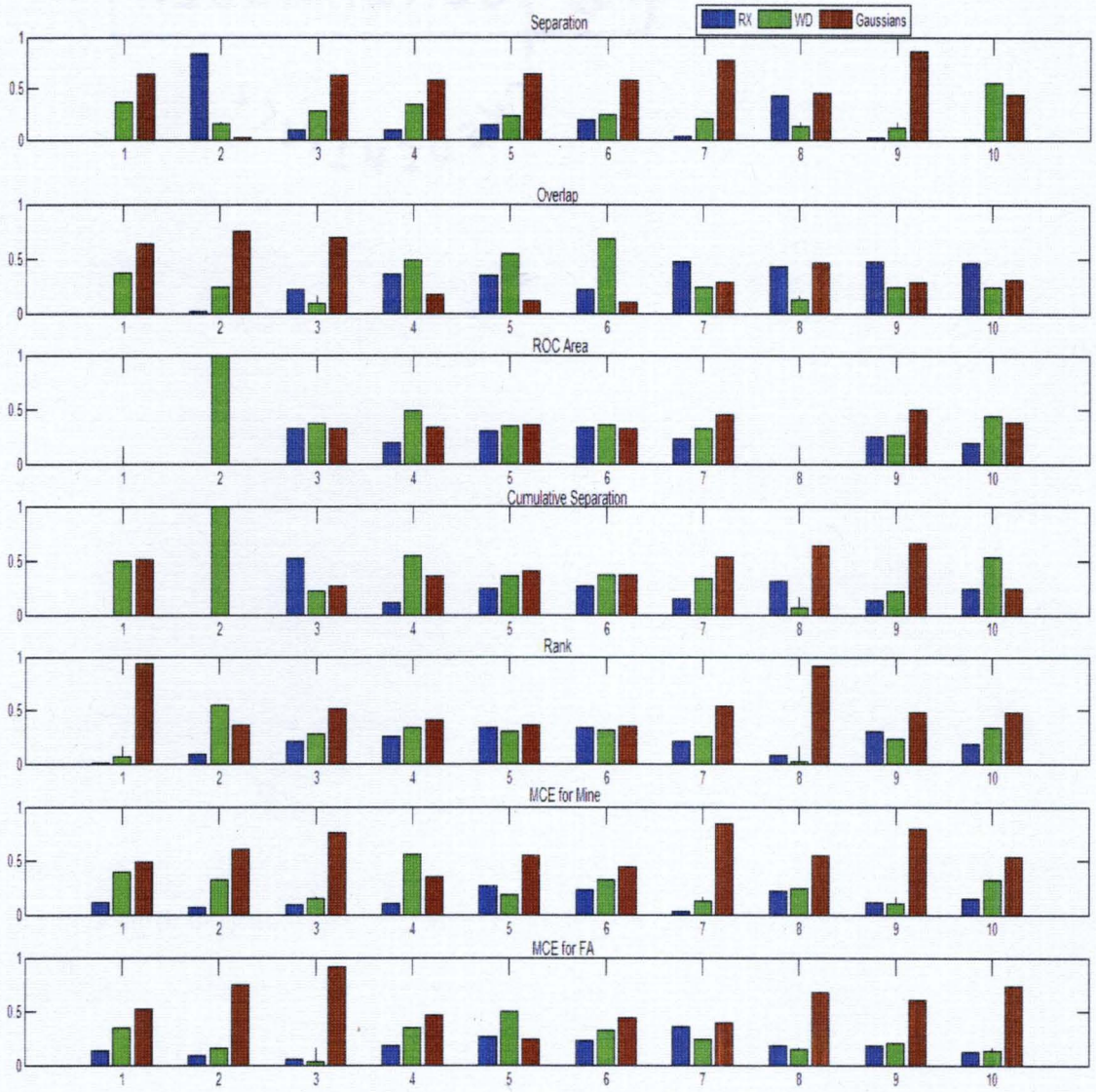


Figure 51. Context-Dependent Fusion weights assigned to three detections within 10 clusters in CV1 in the AHI data

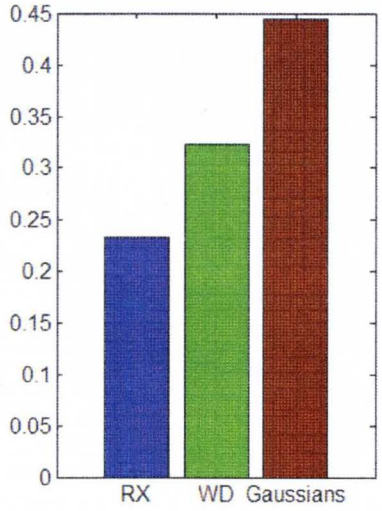


Figure 52. Global weighted average weights of CV1 on the AHI data

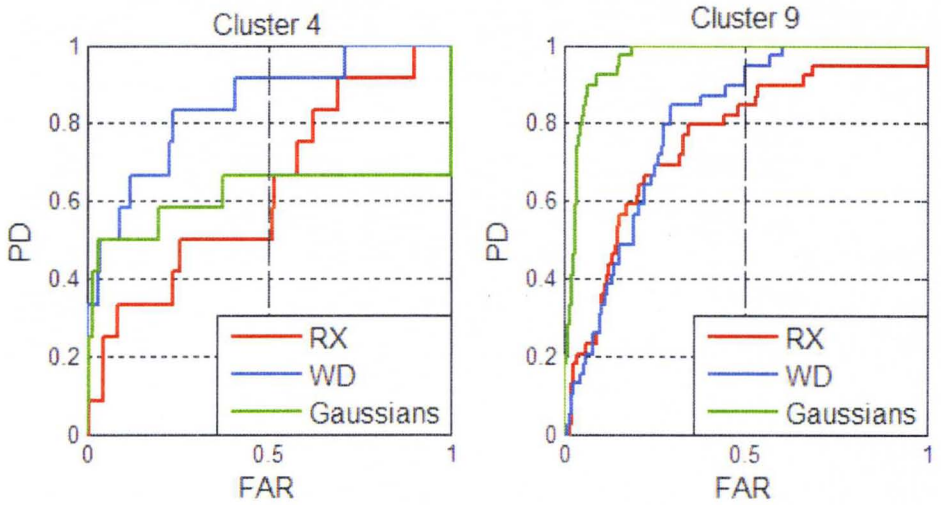


Figure 53. Performance of the 3 individual algorithms in two different clusters

gation weight is assigned to each detector in each context. These weights are shown in Figure 51. For comparison purposes, we also assign a global weight using the entire training collection, i.e. we treat all data as one cluster, and weighted the algorithms according to the ROC area. These weights are shown in Figure 52. As it can be seen, overall, the Gaussian has the best performance followed by the WD and then the RX. However, the performance of the different algorithms can vary significantly from one context to another.

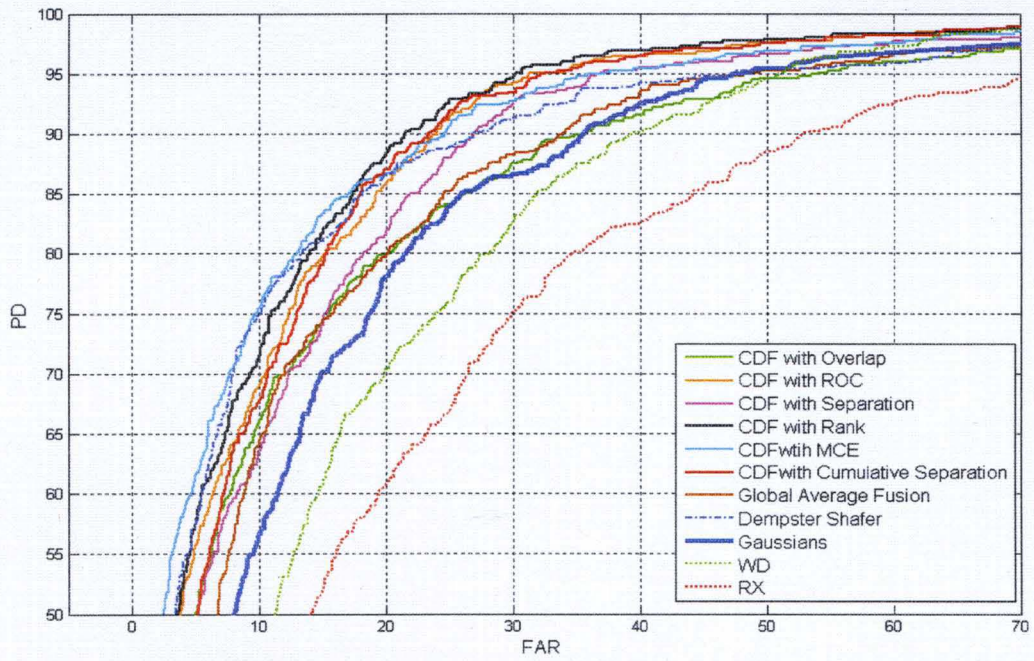


Figure 54. Comparison of the ROCs obtained with the Context-Dependent Fusion and the global fusion

For instance, the Gaussian detector has the second weight in context 4 for ROC-based, Cumulative Separation-based and MCE-based methods. On the other hand, context 9 always assign a highest worthiness to the Gaussian.

The ROC curves of the three algorithms on two typical clusters are shown in Figure 53. These results show that the performance of the algorithms can vary significantly from one cluster to another. For instance, the WD algorithm was the best performer for cluster 4, and the Gaussian is the second one in the PD range [0, 0.5], but in cluster 9, Gaussian is the best and WD and RX are very similar within this context.

The overall performance of the CDF with different local weighting methods on the entire data collection averaged over all cross validations is shown in Figure 54. We also explore the Dempster Shafer fusion [33] and average global fusion on this data for comparison. As it can be seen, all of the CDF methods outperform all of the individual algorithms significantly. Moreover, most of the ROC with this local fusion are better than the ROC obtained with global fusion where the weights are learned in the same way, and also better

than Dempster Shafer fusion. We notice that not all CDF with local weights perform well on this data, for instance, CDF with Overlap and CDF with Separation are slight better than Gaussian detection algorithm. Investigation of this problem has revealed that these two fusion methods generate confidence values that have a distribution close to binary. Due to these nearly binary distributions, weak mines will be assigned confidence values close to zero, and this would explain the lower PD at low FAR. Also, strong false alarms will be assigned confidence values close to 1, and this would explain the relatively lower PD at higher FAR.

CHAPTER VII

CONCLUSIONS

We have proposed a novel Context Dependent Fusion (CDF) method that fuses multiple classification algorithms for decision making. The proposed CDF is a local, dynamic, and feature dependent method that adapts the fusion to different regions of the feature spaces. It has three main components: *context extraction*, *algorithm fusion*, and *decision making*.

The *context extraction* component explores the training data in the feature space. It combines the features extracted by the different algorithms from the different sensors and partitions the aggregate feature space into clusters or contexts. The feature combination step can also be regarded as feature-level fusion. For this step, we have experimented with simple raw features combination and combination after feature mapping and reduction using Principal Component Analysis (PCA). The latter case proved to be more effective. This is because the features extracted by the different algorithm can vary significantly in dimension, range, and type. The PCA provides an effective way to normalize and transform the number of possibly correlated variables into a smaller number of uncorrelated variables for each algorithm. This transformation can also be used to make the dimensionality of the different feature subsets comparable and avoid any bias that may be induced by the algorithms that extract a larger number of features. Another benefit of using the PCA is the reduction in time complexity for both the off-line training and on-line testing phases.

After feature fusion, a clustering algorithm is used to partition the feature space into contexts. We have experimented with three different algorithms to perform this task,

namely the Fuzzy C-Means (FCM), The Self-Organizing Map (SOM), and the Simultaneous Clustering and Attribute Discrimination (SCADc). The FCM has the advantages of being simple and computationally efficient. However, it is not effective in clustering high-dimensional data, cannot discriminate between the different feature sets, and is sensitive to the specified number of clusters. The SOM has the advantage of generating a map that preserves the spatial information. This map could be explored for visualization or to reduce the number of contexts. However, like the FCM, the SOM cannot discriminate between the different feature subsets. Moreover, because the SOM generates a crisp partition, it may not be possible to learn useful information from the small clusters. The SCADc algorithm has the advantages of generating a fuzzy partition, learning the relevant features for each context, and finding the optimal number of contexts. The fuzzy memberships and feature relevance weights are explored in the subsequent steps of the CDF.

The second component of the CDF, *algorithm fusion*, explores the training data in the confidence space. In particular, the confidence values assigned by each algorithm are used to assign aggregation weights to each algorithm within each context and identify "local experts". For this component, we have proposed, implemented, and tested six local weighting methods. Some of these methods are based on the performance of the individual algorithms. Here, performance can be measured by the degree of separation or overlap between the distributions of the confidence values in the different classes. It can also be measured by the area under the Receiver Operating Characteristic (ROC) curve. Another weighting method that we have proposed is based on discriminative learning. We have formulated the problem and derived the necessary conditions to learn weights that minimize the classification error within each context. Extensive experiments were conducted to analyze and compare the performance of the different weighting methods. The results have indicated that all of the above methods outperform the global fusion, however, the relative performance of the proposed local methods can vary from one data to another, and there is

no clear winner.

The third component of the proposed CDF, *decision making*, utilizes the context-dependent weights assigned to each algorithm to perform the final decision-making process. In this step, we explore the fuzzy membership functions learned by SCADc during clustering to assign a fuzzy membership degrees to the test sample into multiple contexts. This multiple context assignment reduces the randomness of the test pattern assignment (when some contexts are similar), and makes the algorithm more robust and consistent.

Another contribution of this thesis is the application of the proposed fusion method to the problem of landmine detection. For this application, it has been established that the performance of different detection algorithms and sensors is strongly dependent upon a variety of factors that are not well understood. It is typically the case that one algorithm (or sensor) may perform well in one setting and not so well in another. Thus, in order to achieve a reliable and robust detection system, several distinct detection algorithms need to be developed and fused. We have applied our CDF method to this problem. In particular, we have applied it to multiple data collected by Vehicle Mounted Ground Penetrating Radar (VMGPR), Autonomous Mine Detection Multi Sensors System with Ground Penetrating Radar (GPR) and Wideband Electromagnetic Induction (WEMI), and Airborne Hyper-spectral Imagery (AHI) systems. Our extensive research and testing in this application have revealed that the CDF can identify meaningful and coherent contexts consistently. Typically, the different contexts include signatures that have similar properties. Examples include alarms of the same type, alarms of targets buried at the same depth, and alarms collected from the same geographical site. Moreover, for each context, CDF identifies the most reliable algorithm/sensor. For instances, the GPR sensor can be more reliable for a context that includes mainly mines with low-metal content, while the WEMI sensor can be more reliable for anti-personnel mines. Similarly, a detector that uses frequency-domain features may be more reliable for one context, while an edge-based detector may be more

reliable for another context.

Our extensive experiments in this application have shown that the CDF outperforms the global fusion and several state-of-the-art fusion techniques. Moreover, experimentation with various data sizes, multiple sensors and algorithms, clustering parameters, and weighting methods have indicated that our proposed fusion is stable and consistent. More importantly, the CDF produces contexts and results that can be interpreted.

The proposed context-dependent fusion approach is a generic approach that partitions the feature space into local contexts and identifies the optimal fusion within each context. In this work, we have developed only simple linear fusion methods. However, our approach can integrate any other fusion method and future research may include investigating fusion methods such as Bayesian, Dempster-Shafer, and fuzzy integral within the CDF framework. Another interesting future work may include the use of semi-supervised clustering to partition the feature space into contexts. In several applications, partial supervision information may be available and may be explored in partitioning the high-dimensional feature space to obtain semantically meaningful contexts. Moreover, the quality of the obtained clusters may need to be assessed and used in the fusion. For instance, a context with good validity measure should be more reliable than a context with worse validity.

REFERENCES

[1] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," in *Computing and Information Systems*, M. Crowe, Ed., vol. 7, pp. 1–10. University of Paisley, 2000.

[2] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, and A. Gelzinis, "Soft combination of neural classifiers: A comparative study," *Pattern Recognit. Lett*, vol. 20, pp. 429–444, 1999.

[3] R. A. Jacobs, "Methods for combining experts probability assessments," *Neural Computation*, vol. 7, no. 5, pp. 867–888, 1995.

[4] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 2, pp. 5–32, 2001.

[5] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, pp. 80–91, 1999.

[6] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 66–75, Jan. 1994.

[7] L. I. Kuncheva, "Switching between selection and fusion in combining classifiers: An experiment," *IEEE Trans. Systems, Man, and CYbernetics-Part B*, vol. 32, no. 2, pp. 146–156, Apr. 2002.

[8] L. A. Rastrigin and R. H. Erensterin, *Method of Collective Recognition*, Moscow, Russian: Energoizdat, 1981.

[9] L. I. Kuncheva, "Change-glasses approach in pattern recognition," *Pattern Recog. Lett.*, vol. 14, pp. 619–623, 1993.

[10] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405–410, 1997.

[11] J. A. MacDonald, *Alternatives for Landmine Detection*, RAND Corporation, 2003.

[12] J. N. Wilson and P. D. Gader, "Use of the borda count for landmine discriminator fusion," in *Detection and Remediation Technologies for Mines and Minelike Targets XII*, Russell S. Harmon, J. Thomas Broach, and Jr. Holloway, John H., Eds., vol. 6553. Proc. of the SPIE, May 2007.

[13] "Hidden killers: The global landmine crisis," *U.S. Dept. State Rep.,Publ. 10575, Washington, DC*, Sep. 1998.

[14] R. K. Sharma, M. Pavel, and T. K. Leen, "Multistream video fusion using local principal components analysis," in *Proc. SPIE on Infrared Technology and Applications XXIV*, B. F. Andresen and M. Strojnik Scholl, Eds., Oct 1998, vol. 3436, pp. 717–725.

[15] H. Schwenk and M. Milgram, "Transformation invariant autoassociation with application to handwritten character recognition," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7, pp. 992–998. The MIT Press, 1995.

[16] W. Pedrycz, J. C. Bezdek, R. J. Hathaway, and G. W. Rogers, "Two nonparametric models for fusing heterogenous fuzzy data," *IEEE Transactions on Fuzzy Systems*, vol. 6, pp. 411–425, 1998.

[17] J. Bezdek, *Fuzzy models and algorithms for pattern recognition and image processing*, Kluwer Academic, 1999.

[18] C. Ji and S. Ma, "Combined weak classifiers," in *Advances in Neural Information Processing Systems 9*, M.C. Mozer, M.I. Jordan, and T. Petsche, Eds., pp. 494–500. MIT Press, Cambridge, 1997.

[19] P. W. Munro and B. Parmanto, "Combining neural network regresion estimates with regularized linear weights," in *Advances in Neural Information Processing Systems 9*, M.C. Mozer, M.I. Jordan, and T. Petsche, Eds., pp. 592–598. MIT Press, Cambridge, 1997.

[20] S. Hashem, "Optimal linear combinations of neural networks," *Neural Networks*, vol. 10, no. 4, pp. 599–614, 1997.

[21] L. Lam and C. Y. Suen, "Optimal combination of pattern classifiers," *Pattern Recognition Letters*, vol. 16, pp. 945–954, 1995.

[22] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. on Pattern Analalysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[23] F. Kimura and M. Shridar, "Handwritten numerical recognition based on multiple algorithms," *Pattern Recognition*, vol. 24, no. 10, pp. 969–983, 1991.

[24] M. Ceccarelli and A. Petrosino, "Multi-feature adaptive classifiers for sar image segmentation," *Neurocomputing*, vol. 14, pp. 345–363, 1977.

[25] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural Computation*, vol. 6, pp. 181–214, 1994.

[26] H. Tahani and J. M. Keller, "Information fusion in computer vision uusing the fuzzy integral," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 20, no. 3, pp. 733–741, 1990.

[27] P. D. Gade, M. A. Mohamed, and J. M. Keller, "Fusion of handwritten work classifiers," *Pattern Recognition Letters*, vol. 17, pp. 577–584, 1996.

[28] S. Le Hegarat-Mascle, I. Bloch, and D. Vidal-Madjar, "Introduction of neighborhood information in evidence theory and application to data fusion of radar and optical images with partial cloud cover," *Pattern Recognition*, vol. 31, no. 11, pp. 1811–1823, 1998.

[29] E. Mandler and J. Schurmann, "Combining the classification results of independent classifiers based on the dempster-shafer theory of evidence," *Pattern Recognition and Artificial Intelligence*, pp. 381–393, 1988.

[30] M. Stone, "The opinion pool," *The Annals of Statistics*, vol. 32, pp. 1339–1342, 1961.

[31] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*, Ellis Horwood, New-York, London, 1994.

[32] L. A. Klein, *Sensor and Data Fusion Concepts and Applications*, SPIE, 1993.

[33] S. Challa and D. Koks, "Bayesian and Dempster-Shafer fusion," *Sadhana*, vol. 29, no. 2, pp. 145–174, 2004.

[34] D. M. Buede and P. Girardi, "Information fusion in computer vision uusing the fuzzy integral," *IEEE Transactions on Systems, Man and Cybernetics-Part A*, vol. 27, no. 5, pp. 569–577, 1999.

[35] D. Fasbender, J. Radoux, and P. Bogaert, "Bayesian data fusion for adaptable image pansharpening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1847–1857, 2008.

[36] F. Cremer, E. Breejen, and K. Schutte, "Sensor data fusion for anti-personnel landmine detection," in *Proc. of the Int'l Conference on Data Fusion (EuroFusion98)*, 1998, pp. 55–60.

[37] E. Breejen, K. Schutte, and F. Cremer, "Sensor fusion for anti personnel landmine detection: a case study," in *Proc. of the SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets IV*, 1999, pp. 1235–1245.

[38] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley, New York, 1973.

[39] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[40] G. Shafer, *A Mathematical Theory of Evidence*, Princeton, NJ, Princeton University Press, 1996.

[41] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *The Annals of Statistics*, , no. 28, pp. 325–339, 1967.

[42] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods for combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[43] M. Beynon, D. Cosker, and A. D. Marshall, "An expert system for multi-criteria decision making using dempster shafer theory," *Expert Systems with Applications*, vol. 20, no. 4, pp. 357–367, 2001.

[44] Y. A. Aslandogan and C. T. Yu, "Evaluating strategies and systems for content based indexing of person images on the web," in *Proc. of the ACM Int'l Multimedia Conference and Exhibition*, 2000, pp. 313–321.

[45] N. Milisavljevic and I. Bloch, "Sensor fusion in anti-personnel mine detection using a two-level belief function model," *IEEE SMC, PArt C: Applications and Reviews*, vol. 33, pp. 269–283, 2003.

[46] K. Sentz, "Combination of evidence in Dempster-Shafer theory, technical report, sand 2002-0835," .

[47] R. Yager, "On the Dempster-Shafer framework and new combination rules," *Information Sciences*, vol. 41, pp. 93–137, 1987.

[48] L. A. Zadeh, "A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination," *The AI Magazine*, vol. 7, pp. 85–90, 1987.

[49] C. Lee, "A comparison of two evidential reasoning schemes," *Artifical Intelligence*, vol. 35, no. 1, pp. 127–134, 1988.

[50] P. L. Bolger, "Shafer-Dempster reasoning with applications to multisensor target identification systems," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 22, no. 6, pp. 968–977, 1987.

[51] G. J. Klir and M. J. Wierman, *Uncertainty-Based Information:Elements of Generalized Information Theory*, Heidelberg, Physica-Verlag, 1998.

[52] J. C. de Borda, *Mémoire sur les élections au scrutin*, Histoire de l'Académie Royale des Sciences, Paris, 1781.

[53] M. J. A. N. de Caritat Condorcet and Marquis de, *Essai sur lapplication de lanalyse á la probabilité des décisions rendues ála pluralité des voix*, Paris, Imprimerie Royale, 1785.

[54] D. Black, *The Theory of Committees and Elections, 2Ed*, Kluwer Academic Publishers, Boston, 1998.

[55] K.J. Arrow, *Social Choice and Individual Values, 2 Ed*, Yale University Press, New Haven, CT, 1963.

[56] K. van Erp and L. Schomaker, "Variants of the borda count method for combining ranked classifier hypotheses," in *Proc. of the Seventh International Workshop on Frontiers in Handwriting Recognition*, vol. 11-13, pp. 443–452. Sep. 2000.

[57] I. McLean, "E. j. nanson, social choice, and electoral reform," *Australian Journal of Political Science*, vol. 31, pp. 369–385, Nov. 1996.

[58] J. T. Miller, H. B. Thomas, and J. Soukup, "Simple phenomenological models for wideband frequency-domain electromagnetic induction," *IEEE Trans. Geoscience and Remote Sensing*, vol. 39, pp. 1294–1298, Jun. 2001.

[59] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognition*, vol. 34, no. 2, pp. 299–314, 2001.

[60] Y. S. Huang and Suen C. Y., "A method of combining multiple experts for the recognition of unconstratined handwritten numerals," *IEEE Trans. Pattern Analysis and Mechine Intelligence*, vol. 17, no. 1, pp. 90–94, 1995.

[61] H. J. Kang, K. Kim, and J. H. Kim, "Optimal approximation of discrete probability distribution with kth-order dependency and its application to combining multiple classifiers," *Pattern Recognition Letters*, vol. 18, pp. 515–523, 1997.

[62] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. on Neural Networks*, vol. 10, pp. 988–999, 1999.

[63] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *In Proc. of the Second European Conference on Computational Learning Theory*, London, UK, 1995, pp. 23–37, Springer-Verlag.

[64] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *In Proc. of the Thirteenth International Conference on Machine Learning*. 1996, pp. 148–156, Morgan Kaufmann.

[65] Y. Freund and R. E. Schapire, "A brief introduction to boosting," in *In Proc. of the Sixteenth International Joint Conference on Artificial Intelligence*. 1999, pp. 1401–1406, Morgan Kaufmann.

[66] *Landmines, Mine Action News from the United Nations*, vol. 3.2, 4th Qtr., 1998.

[67] T. R. Witten, "Present state of the art in ground-penetrating radars for mine detection," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets III*, pp. 576–586. Orlando, Florida, 1998.

[68] P. D. Gader, H. Frigui, B. Nelson, G. Vaillette, and J. M. Keller, "New results in fuzzy set based detection of landmines with gpr," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets IV*, p. 10751084. Orlando, Florida, 1999.

[69] *http://diwww.epfl.ch/w3lami/detec/susdemsurvey.html.*

[70] *http://www.niitek.com.*

[71] R. Wu, A. Clement, J. Li, and E. Larsson, "Adaptive ground bounce removal," *Electron. Lett.*, , no. 37, pp. 1250–1252, 2001.

[72] K. Gu, J. Li, M. Bradley, J. Habersat, and G. Maksymonko, "Adaptive ground bounce removal," in *Proc. of SPIE. Int. Soc. Opt. Eng.*, vol. 4742, p. 719727. Orlando, Florida, Apr. 2002.

[73] L. Carin, N. Geng, and M. McClure, "Ultra-wide-band syntheticaperture radar for mine-field detection," *IEEE Antennas Propag. Mag.*, vol. 41, pp. 1833, 1999.

[74] G. D. Sower and S. P. Cave, "Detection and identification of mines from natural magnetic and electromagnetic resonances," in *Proc. of SPIE.*, vol. 2496, pp. 1015–1024. Orlando, Florida, Apr. 1995.

[75] A. H. Trang, P. V. Czipott, and D. A. Waldron, "Characterization of small metallic objects and nonmetallic anit-personnel mines," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets I.* Orlando, Florida, Apr. 1997.

[76] K. Tsipis, "Report on the landmine brainstorming workshop of aug. 25-30, nov. 96," in *Report No.27, Program in Science and Technology for International Security.* MIT, Cambridge, MA, USA. Web: http://mcnutt.mit.edu/PSTIS/minereport/minereport.html, 1996.

[77] D. Mills, *Improvements to Mine Detectors*, Number PO1408. Australian patent application, 1996.

[78] L. Carin, H. Yu, Y. Dalichaouch, and C. Baum, "On the wideband emi response of a rotationally symmetric permeable and conducting target," *IEEE Trans. Geosci. Remote Sensing*, vol. 39, pp. 1113–1206, June 2001.

[79] I. J. Won, D. A. Keiswetter, and D. R. Hansen, "Gem-3: a monostatic broadband electromagnetic induction sensor," *J. Environ. Eng. Geophys*, vol. 2, pp. 53–64, 1997.

[80] J. R. Simard, "Improved landmine detection capability (ILDC): Systematic approach to the detection of buried mines using ir imaging," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets II*, vol. 3079. Orlando, Florida, Apr. 1997.

[81] K. L. Russell, J. E. McFee, and W. Sirovyak, "Remote performance prediction for infrared imaging of buried mines," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets II*, pp. 762–769. Orlando, Florida, Apr. 1997.

[82] T. Gozani, "Inspection techniques based on neutron interrogation," in *Proc. of SPIE. Physics-Based Technologies for the Detection of Contraband*, number 2936, pp. 9–20. Nov. 1996.

[83] R. B. Moler, "Nuclear techniques for mine detection research," in *Technical Report.* Lake Luzerne, NY, Sponsored by BRDEC, Ft. Belvoir, VA, U.S.A., July 1985.

[84] R. B. Moler, "Nuclear and atomic methods of mine detection," in *Workshop Report.* Contract DAAK70-89-C-0002, Dep. of the Army BRDEC, Ft. Belvoir, VA, U.S.A., Nov. 1991.

[85] J. E. McFee and Y. Das, "Advances in the location and identification of hidden explosive munitions," *Defence Research Establishment Suffield, Report*, , no. 548, pp. 83, Feb. 1991.

[86] P. Bach, "Neutron activation and analysis," in *EUREL Int'l Conf. On The Detection of Abandoned Land Mines*, pp. 58–61. Edinburgh, UK, Oct. 1996.

[87] D. Carevic, "Clutter reduction and target detection in ground penetrating radar data using wavelets," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets IV*, p. 973978. Orlando, Florida, 1999.

[88] D. Carevic, "Kalman filter-based approach to target detection and target-background separation in ground-penetrating radar data," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets IV*, p. 12841288. Orlando, Florida, 1999.

[89] A. Gunatilaka and B. A. Baertlein, "Subspace decomposition technique to improve gpr imaging of anti-personnel mines," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets V*, pp. 1008–1018. Orlando, Florida, 2000.

[90] S. Yu, R. K. Mehra, and T. R. Witten, "Automatic mine detection based on ground penetrating radar," in *Proc. of SPIE. Detection and Remediation Technologies for Mines and Minelike Targets IV*, p. 961972. Orlando, Florida, 1999.

[91] H. Frigui, K. Satyanarayana, and P. Gader, "Detection of land mines using fuzzy and possibilistic membership functions," in *proc. of the IEEE Conference on Fuzzy Systems*, vol. 2, pp. 834–839. Saint Louis, Missouri, 2003.

[92] P. Gader, M. Mystkowski, and Y. Zhao, "Landmine detection with ground penetrating radar using hidden markov models," *IEEE Trans. Geoscience and Remote Sensing*, vol. 39, pp. 12311244, 2001.

[93] H. Frigui, K. C. Ho, and P. Gader, "Real-time land mine detection with ground penetrating radar using discriminative and adaptive hidden markov models," *EURASIP Journal on Applied Signal Processing*, vol. 12, pp. 1867–1885, 2005.

[94] P. Gader, B. Nelson, H. Frigui, G. Vaillette, and J. Keller, "Fuzzy logic detection of landmines with ground penetrating radar," *Signal Processing, special issue on fuzzy logic in signal processing*, vol. 80, pp. 1069 – 1084, 2000.

[95] P. Gader, W. H. Lee, and J. N. Wilson, "Detecting landmines with ground penetrating radar using feature-based rules, order statistics, and adaptive whitening," *IEEE Trans. Geoscience and Remote Sensing*, vol. 42, no. 11, pp. 2522–2534, 2004.

[96] R. Grandhi, "Integration of ordered weighted averaging operators with feed-forward neural networks for optimal feature subset selection and pattern classification," M.S. thesis, UNIVERSITY OF FLORIDA, 2003.

[97] H. Frigui and P. D. Gader, "Detection and discrimination of land mines based on edge histogram descriptors and fuzzy k-nearest neighbors," in *Proc. of the IEEE Int'l Conf. on Fuzzy Systems*, Vancouver, BC, Canada, July 2006.

[98] K. J. Hintz, "Snr improvements in NIITEK ground penetrating radar," in *Proc. of the SPIE Conf. on Detection and Remediation Technologies for Mines and Minelike Targets IX*, Orlando, FL, USA, April 2004.

[99] G. D. Forney, "The viterbi algorithm," *Proc. of the IEEE*, vol. 61, pp. 268–278, 1973.

[100] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG 7: Multimedia Content Description Language*, John Wiley, 2002.

[101] T. Kohonen, *Self-Organization and Associative Memory*, Springer Verlag, 1989.

[102] W. H. Lee, P. D. Gader, and J. N. Wilson, "Optimizing the area under a receiver operating characteristic curve with application to landmine detection," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 45, no. 2, pp. 389–397, 2007.

[103] K. C. Ho, L. Carin, P. D. Gader, and J. N. Wilson, "On using the spectral features from ground penetrating radar for landmine-clutter discrimination," submitted to IEEE Trans. Geoscience and Remote Sensing.

[104] W. R. Scott, "Broadband array of electromagnetic induction sensors for detecting buried landmines," in *Proc. of the IEEE Geoscience and Remote Sensing Symposium (IGARSS 2008)*, Boston, MA, July 2008, vol. 2, pp. 375–378.

[105] E. Fails, P. Torrione, W. Scott, and L. Collins, "Performance of a four parameter model for landmine signatures in frequency domain wideband electromagnetic induction detection systems," in *Proc. of the SPIE Conf. on Detection and Remediation Technologies for Mines and Minelike Targets XII*, April 2007, vol. 6553, pp. 1–8.

[106] P. D. Gader, J. N. Wilson, D. Ho, S. E. Yuksel, G. Ramachandran, and G. Heo, "Hierarchical methods for landmine detection with wideband electromagnetic induction and ground penetrating radar multi-sensor systems," in *Proc. of the IEEE Geoscience and Remote Sensing Symposium (IGARSS 2008)*, Boston, MA, July 2008.

[107] H. Huang and I. J. Won, "Automated anomaly picking from broadband electromagnetic data in an unexploded ordance (uxo) survey," *Geophysics*, vol. 68, no. 6, pp. 1870–1876, 2003.

[108] H. Frigui, L. Zhang, and P. D. Gader, "Context-dependent fusion for landmine detection with ground-penetrating radar," in *Proc. of the SPIE Conf. on Detection and Remediation Technologies for Mines and Minelike Targets XII*, FL, USA, Apr. 2007.

[109] H. Frigui, L. Zhang, P. D. Gader, J. Wilson, K. C. Ho, and A. Mendez-Vasquez, "A large-scale evaluation of several fusion algorithms for landmine detection and discrimination," *IEEE Int'l Journal on Information Fusion*, Jan. 2008.

[110] H. Frigui, L. Zhang, and P. D. Gader, "Context-dependent multi-sensor fusion for landmine detection," in *IEEE Int'l on Geoscience and Remote Sensing Symposium*, Boston, USA, 2008, vol. 2, pp. 371–374.

[111] L. Zhang, H. Frigui, and P.D. Gader, "Context-dependent fusion for mine detection using airborne hyperspectral imagery," in *The First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS2009)*, Grenoble, France, 2009.

[112] L. Zhang, H. Frigui, and P.D. Gader, "Contex-dependent fusion of multiple algotithms with minimum classification error learning," in *The Eighth International Conference on Machine Learning and Applications (ICMLA2009)*, Miami, USA, 2009.

[113] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers and Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[114] H. Almuallim and T. G. Dietterich, "Learning with many irrelevant features," in *Ninth National Conference on Artificial Intelligence*, 1991, pp. 547–552.

[115] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Tenth National Conference on Artificial Intelligence*, 1992, pp. 129–134.

[116] L. A. Rendell and K. Kira, "A practical approach to feature selection," in *Int'l Conference on Machine Learning*, 1992, pp. 249–256.

[117] H. Frigui and S. Salem, "Fuzzy clustering and subset feature weighting," in *IEEE Int'l Conference on Fuzzy Systems*, 2003.

[118] K. Tumer and J. Ghosh, *Linear and Order Statistics Combiners for Reliable Pattern Classification*, Ph.D. thesis, The Univ. of Texas, Austin, 1996.

[119] H. A. David, *Order Statistics*, Wiley, New York, 1970.

[120] B. H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. on Signal Processing*, vol. 40, no. 12, pp. 3043–3054, 1992.

[121] H. Shimodaira, J. Rokui, and M. Nakai, "Modified minimum classification error learning and its application to neural networks," in *Proc. of the Joint IAPR Int'l Workshops on Advances in Pattern Recognition*, London, UK, 1998, pp. 785–794, Springer-Verlag.

[122] *http://www.dice.ucl.ac.be/mlg/?page=Elena.*

[123] P. Alinat, "Periodic progress report 4, rpars project esprit ii," Tech. Rep. 5516, Thomson report TS, ASM 93/S/EGS/NC/079, Feb. 1993.

[124] P. A. Torrione, C. S. Throckmorton, and L. M. Collins, "Performance of an adaptive feature-based processor for a wideband ground penetrating radar system," *IEEE Trans. Aerospace and Electronic Systems*, vol. 42, no. 2, pp. 644–658, 2006.

[125] L. Ayers and E. Rosen, "MIDAS: Mine detection assessment and scoring user's manual v1.1," Institute for Defense Analysis, Technical Report, 2004.

[126] P. Torrione and L. M. Collins, "Texture features for antitank landmine detection using ground penetrating radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 7, pp. 2374–2382, 2007.

[127] P. Torrione, "personal communication," .

[128] P. A. Torrione and L. Collins, "Application of Markov random fields to landmine detection in ground penetrating radar data," in *Proceedings of the SPIE Conference on Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIII*, 2008, vol. 6953, pp. 69531B–69531B–12.

[129] M. H. Horng, "Texture feature coding method for texture classification," *Optical Engineering*, vol. 42, no. 1, pp. 228–238, 2003.

[130] J. T. Miller, T. H. Bell, J. Soukup, and D. Keiswetter, "Simple phenomenological models for wideband frequence-domain electromagnetic induction," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 39, no. 6, pp. 1294–1298, 2001.

[131] H. Huang and I. J. Won, "Automated anomaly picking from broadband electromagnetic data in an unexploded ordance (uxo) survey," *Geophysics*, vol. 68, no. 6, pp. 1870–1976, 2003.

[132] J. Bolton and P. D. Gader, "Application of context-based classifier to hyperspectral imagery for mine detection," in *SPIE on Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIII*, R.S. Harmon, J.H. Holloway, Jr., and J.T. Broach, Eds., vol. 6953. Orlando, FL, USA, 2008.

[133] I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern withunknown spectral distribution," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1760–1770, 1990.

[134] R. Mayer, F. Bucholtz, and D. Scribner, "Object detection by using" whitening/dewhitening" to transform target signatures in multitemporal hyperspectral and multispectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 41, no. 5, pp. 1136–1142, 2003.

APPENDIX A

LIST OF ABBREVIATIONS

AHI	airborne hyperspectral imagery
AP	anti-person
APHM	anti-person high metal
APLM	anti-person low metal
AT	anti-tank
ATHM	anti-tank high metal
ATLM	anti-tank low metal
AUC	area under receiver operating characteristic (ROC) curve
BBA	basic belief assignment
Bel	belief function
CDF	context dependent fusion
CV	cross validation
DST	Dempster Shafer theory
DT	decision template
DP	decision profile
EDS	energy density spectrum
EHD	edge histogram descriptor
EM	expectation maximization
FAR	false alarm rate
FCM	fuzzy c-means

FOWA	feed-forward order-weighted-average
FPR	false positive rate
GEOM	geometric feature
GFIT	Gaussian fit descriptor
GMRF	Gaussian Markov random field descriptor
GPD	generalized probabilistic descent
GPR	ground penetrating radar
HM	high metal
HMC	high metal clutter
HMM	hidden Markov model
LDA	linear discriminant analysis
LM	low metal
LMC	low metal clutter
LMS	least mean squares
MCE	minimum classification error
MD	metal detectors
ML	maximum likelihood
MLP	multi-layer perceptron
MPP	mapped posterior probability
MWM	meandering winding magnetometer
PCA	principal component analysis
PD	probability of detection
PFA	probability of false alarm
Pl	plausibility function
POI	points of interest
QDA	quadratic discriminant analysis

ROC	receiver operation characteristic
SAR	synthetic aperture radar
SCADc	coarse simultaneous clustering and attribute discrimination
SOM	self-organizing map
SPECT	spectral feature
TFCM	texture feature classification method detector
TPR	true positive rate
TUF	testing/training unified framework
VMMDS	vehicle mounted mine detection system
WD	whitening dewhitening
WEMI	wideband electromagnetic induction

CURRICULUM VITAE

NAME: Lijun Zhang

ADDRESS: Department of Computer Engineering and Computer Science
University of Louisville
Louisville, KY 40292

RESEARCH INTERESTS: Classification, Clustering, Data Mining, Machine Learning
Pattern Recognition, Multimedia & Image Processing,
Fusion Theory, Content Based Image Retrieval

EDUCATION:

Ph.D., Computer Engineering and Computer Science, August 2009
University of Louisville, Louisville, KY, USA
- **Dissertation Topic:** "Context Dependent Fusion with Application to Landmine Detection"
- **Advisor:** Prof. Hichem Frigui, Ph.D

M.Sc., Electrical Engineering, March 2005
Shanghai Jiaotong University, Shanghai, China
- **Thesis Topic:** "3D Facial Reconstruction System Based on Turn-Table"
- **Advisor:** Prof. Xin Yang, Ph.D

B.Sc., Electrical Engineering, July 1999
Harbin Institute of Technology, Harbin, China

RESEARCH EXPERIENCE:

- **Research Assistant**, 8/2005 - 8/2009
  UNIVERSITY OF LOUISVILLE, LOUISVILLE, KY
  - ✓ ***Classification and Clustering Methods:***
    Implemented classification/clustering algorithms, e.g. KNN ANN, Support Vector Machines (SVM), EM, Decision Tree, Logistic Regression, Bayes, Simultaneous Clustering and Attribute Discrimination.
  - ✓ ***Content-Based Image Retrieval:***
    Initiated and implemented Content-Based Image Retrieval

on medical image and general image.

✓ ***Context-Based Fusion:***
This fusion frame has three main components, i.e., Context Extraction, Context Fusion and Test Pattern Decision. The three components were integrated into a complete cross validation system which was validated by utilizing six large real data sets (different times and locations) for accuracy, and improvement over basic detection algorithms.

✓ ***Algorithms Embedded into Fusion Frame:***
Several parameters aggregation methods (Separation-Based, Overlap-Based, ROC Area-Based, Rank-Based, and MCE/GPD Based methods) were investigated, derived and embedded into our fusion frame. Also several clustering, classification algorithms (SCAD, FCM, SOM, Bayesian, EM, etc.) were implemented and applied into this frame.

✓ ***Global Fusion Methods:***
EM, Borda Count, Bayesian, Decision Temple, and Dempster Shafer Algorithms were explored, and employed to the real world data sets.

- **R&D Software Engineer**, 2/2005 - 8/2005
  SYNTEST TECHNOLOGIES, INC. SHANGHAI, CHINA
  ✓ Developed and debugged software for the Automatic Test Pattern Generation (ATPG) and Design For Test (DFT) on chip with Tcl/Tk under Linux.

- **Research Assistant**, 9/2002 - 3/2005
  SHANGHAI JIAO TONG UNIVERSITY, CHINA
  ✓Initiated and created 3D Facial Reconstruction System
  ✓Thesis: 3D Facial Reconstruction System Based on Turn Table

- **R&D Intern**, 12/2003 - 5/2004
  WIRELESS COMMUNICATION GROUP
  PHILIPS RESEARCH EAST ASIA, CHINA
  ✓Performed baseband signal algorithm based on TD-SCDMA standard on the platform called General Smart Antenna Testbed (GSAT). Coding in C and assemble language on DSP (TMS320C6416) hardware to implement the algorithm.

- **R&D Engineer**, 7/1999 - 8/2002
  QINGDAO AUTOMATION RESEARCH INSTITUTE, CHINA

✓Project Manager (8/2001 - 8/2002):
Presided over the R&D of Leap Imaging Logging Computer System (Leap system) which utilized VME bus and AD DSP to process image signals;
✓R&D Engineer (5/2000 - 8/2001):
Provided technology support for 521 computerized logging system (521 System).

PUBLICATIONS:

JOURNALS:

1. H. Frigui, **L. Zhang**, and P. D. Gader, "Context Dependent Multi-Sensor Fusion and its application to Land mine Detection," to appear in *IEEE Trans. Geoscience and Remote Sensing*, 2009.
2. H. Frigui, **L. Zhang**, J. Wilson, K. Ho, and A. Mendez-Vazquez, "A Evaluation of Several Fusion Algorithms for Anti-tank Landmine Detection and Discrimination," in *IEEE Int'l Journal on Information Fusion*, Oct, 2008.

CONFERENCES:

3. **L. Zhang**, H. Frigui, P. D. Gader, "Context-Dependent Fusion of Multiple Algotithms with Minimum Classification Error Learning," in *8th Int'l Conf. on Machine Learning and Applications (ICMLA 2009)*, Miami, USA, Dec. 2009.
4. **L. Zhang**, H. Frigui P. D. Gader, and J. Bolton, "Context Dependent Fusion for Mine Detection Using Airborne Hyperspectral Imagery," in *IEEE on the 1st Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, Grenoble, France, Aug. 2009.
5. H. Frigui, **L. Zhang**, P. D. Gader,, "Context-Dependent Multi-Sensor Fusion for Landmine Detection," in *IEEE Int'l Geoscience & Remote Sensing Symposium*, p.371-374, Boston, USA, July, 2008.
6. H. Frigui, **L. Zhang**, P. D. Gader,, "Comparison of Different Fusion Methods for Landmine Detection with Ground Penetrating Radar," in *TUXO Countermine/Range Forum 2007*, Orlando, Aug. 2007.
7. H. Frigui, **L. Zhang**, P. D. Gader,, "Context-dependent fusion for landmine detection with ground-penetrating radar," in *proc. of the SPIE Conf. on Detection and Remediation Technologies for Mines and Minelike Targets XII*, p.1-10, FL, USA, Apr. 2007.