LANDMINE CLASSIFICATION USING POSSIBILISTIC K-NEAREST NEIGHBORS WITH WIDEBAND ELECTROMAGNETIC INDUCTION DATA

A Thesis

Presented to

The Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

JOSEPHINE DULA

Dr. Alina Zare, Thesis Supervisor

DEC 2012

The undersigned, appointed by the dean of the Graduate School,

have examined the thesis entitled

LANDMINE CLASSIFICATION USING POSSIBILISTIC K-NEAREST NEIGHBORS WITH WIDEBAND ELECTROMAGNETIC INDUCTION DATA

presented by Josephine Dula,

a candidate for the degree of master of science,

and hereby certify that, in their opinion, it is worthy of acceptance.

____

Alina Zare, Ph.D.

____

Dominic Ho, Ph.D.

____

Mihail Popescu, Ph.D.

To Mom and Dad

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Alina Zare for the continuous support through my Masters study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me throughout my research and compilation of this thesis. I could not have imagined having a better advisor and mentor for my Masters study. I would also like to thank my committee members, Dr. Dominic Ho and Dr. Mihail Popescu for all of their help and valuable suggestions. I would also like to thank my parents and sister for encouraging me for higher studies and being a constant source of motivation, and friends for their invaluable support and love. Lastly, I would like to thank my husband for his love and patience throughout my Masters journey.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii
LIST OF FIGURES v
LIST OF TABLES vii
LIST OF ABBREVIATIONS viii
ABSTRACT ix
1. PROBLEM DESCRIPTION AND LITERATURE REVIEW 1
1.1. Wide-band Electromagnetic Induction Sensors 1
1.2. Algorithm Literature Review 3
1.2.1. Literature Review of Methods for WEMI Data 4
1.2.2. Literature Review on Dictionary Matching Techniques 5
1.2.3. Literature Review on Methods Applied to Landmine Detection 9
2. POSSIBILISTIC DISSIMILARTY MEASURE AND K-NEAREST NEIGHBOR CLASSIFIER 12
2.1. Discrete Spectrum of Relaxation Frequencies 14
2.2. Joint Orthogonal Matching Pursuits 15
2.3. Possibilistic K-NN Classifier for WEMI Landmine Data 18
2.3.1. Training 22
2.3.2. Testing 24
2.3.3. Final Confidence Assignment 27
3. DICTIONARY ESTIMATION USING CLUSTERING 28
3.1. K-Means Clustering to Estimate Dictionary Elements 30
3.2. Algorithm implementation and Experimental Set-up 31
4. DICTIONARY ESTIMATED FROM LINEAR COMBINATION MODEL 33
4.1. Pseudo code for Model-based Estimation of Dictionary Elements 37
5. EXPERIMENTS AND RESULTS 38
5.1. Data Set Description 38
5.1.1. Dictionary counts on WEMI Data 39
5.2. Experiments and Results with the DSRF Dictionary using the Possibilistic K-NN classifier 41

5.2.1. Confusion matrix between 17 target types 44

5.2.2. Cross Validation on the Real Data Set 45

5.2.3. Comparison of Results between Possibilistic K-NN Classifier with DSRF Dictionary and the K-Nearest Neighbor Classifier 46

5.3. Experiments and Results on using the classifier with the Clustered Dictionary 48

5.4. Experiments and Results using Model-based Dictionary Estimation 51

5.5. Summary and Future work 55

6. CONCLUSION 57

7. REFERENCES 58

LIST OF FIGURES

Figure Page

1. Overview of Possibilisitc KNN Prescreener, Training and Testing 13
2. DSRF Dictionary along with the horizontal (blue line) soil model 15
3. JOMP confidence assignment for a point 17
4. (a)-(c) Argrand diagrams for the two taps and the corresponding dictionary element for target type I (d)-(f) Argrand diagrams for the two taps and the corresponding dictionary element for target type II (g)-(i) Argrand diagrams for the two taps and the corresponding dictionary element for target type III 19
5. Overview of Possibilistic KNN classifier with K-Means Clustering Dictionary Estimation 29
6. Dictionary map (Target type 1) and reconstruction with all the cluster prototypes obtained... 34
7. Overview of Possibilistic KNN classifier using Model-based Dictionary Estimation 36
8. Dictionary counts for different target types 40
9. Normalized Dictionary counts 40
10. The dictionary maps and corresponding Argand diagrams for Target type 5 41
11. ROC obtained using the proposed method for the test dataset. The blue curve shows the JOMP prescreener ROC and the red curve shows the proposed classifier results... 42
12. (a)Confusion Matrix showing the classification accuracy of 64.14% among 17 target type. (b) Weighted confusion Matrix showing the sum of the possibilistic weights 43
13. ROC for 10-fold cross-validation 45
14. ROC comparisons between JOMP Prescreener and K-Nearest Neighbor classifier 47
15. JOMP confidence map comparison between DSRF and clustered dictionary on a single alarm... 48
16. Clustered dictionary obtained by varying the number of clusters 49

17. ROC comparisons between JOMP and proposed classifier using clustered dictionary (K=30, 35, 40 …60), All ROC curves have the same x-axis scale for comparison. 50
18. ROC showing the comparison of JOMP prescreener to the proposed classifier using dictionary learnt using model based method. 52
19. Classifier performance with the model based dictionary when targets were divided by type and depth 53
20. ROC comparing JOMP with proposed classifier using model based dictionary for anti-tank type mines 54
21. ROC comparison between JOMP and proposed classifier using model based dictionary on anti-personal mine type 55

LIST OF TABLES

Page

Table

1. Target Distribution in WEMI data 39

2. Counts of different target types. Some targets types have zero count since those types are not found by JOMP prescreener algorithm... 43

3. Comparison between JOMP and proposed classifier at fixed FAR..................... 44

LIST OF ABBREVIATIONS

Page

ART
Adaptive Resonance Theory 11
ARTMAP
Adaptive Resonance Theory Mapping 11
BP
Basis Pursuits 8
DSRF
Discrete Spectrum of Relaxation Frequencies 13
EM
Expectation Maximization 5
EMI
Electromagnetic Induction 1
FAM
Fuzzy Adaptive Mapping 11
FAR
False Alarm Rate 41
GMP
Genetic Matching Pursuits 7
GPR
Ground Penetrating Radar 4
GRANMA
Gradient Angle Model Algorithm 10
JOMP
Joint Orthogonal Matching Pursuits 17
KNN
K Nearest Neighbor 12
MP
Matching Pursuits 7
OMP
Orthogonal Matching Pursuits 7
OWA
Order Weighted Averages 56
PD
Percentage of Detection 41
ROC
Receiver Operating Characteristics 41
RSS
Residual Sum of Squares 34
UXO
Unexploded ordnance 4
WEMI
Wide-band Electromagnetic Induction 1

ABSTRACT

In this thesis, a possibilistic K-nearest neighbor classifier is presented to distinguish between and classify mine and non-mine targets on data obtained from wideband electromagnetic induction sensors. The goal of this work is to develop methods for classifying wide-band electromagnetic induction data into one of several target classes or a non-target class. For some landmine detection systems, it could be necessary or helpful to discriminate between the several classes of targets, so that they can be analyzed and processed according to their specific properties. For example, it might be of importance to distinguish mines with high-metal content versus low-metal content or distinguish between anti-personnel versus anti-tank landmines. The proposed classifier achieves this goal using a method that is motivated by the observation that different buried object types often have consistent signatures depending on their metal content, size, shape, and depth. Given a sparse representation obtained using the joint orthogonal matching pursuits algorithm, particular target types consistently selected the same dictionary elements for their sparse representation. The proposed classifier distinguishes between particular target types using the frequency of dictionary elements selected by an alarm. Possibilistic weights are assigned for each alarm for sixteen landmine target classes as well as a false alarm class.

The proposed classifier accuracy is compared to several state of the art methods and it shows improvement in discrimination results.

1. PROBLEM DESCRIPTION AND LITERATURE REVIEW

For many years, extensive effort has been expended developing techniques for efficiently locating buried landmines. In order to achieve this, research has been done to improve the sensors and devices that are used to detect the landmines as well as research into the algorithms used to identify landmines in the collected data. This is made extremely challenging with the presence of clutter such as rocks, roots, cans, etc. These clutter objects often have very similar characteristics to target objects which makes target discrimination difficult. One type of sensor used for landmine detection is metal detector with electromagnetic sensors [1], [2].

In the following section, an overview of Wide-band Electromagnetic Induction (WEMI) sensors are provided as well as a summary of related work in landmine detection and dictionary matching techniques

1.1. Wide-band Electromagnetic Induction Sensors

Electromagnetic induction (EMI) sensors are capable of detecting many landmine types. However, they will also detect many metal clutter objects. Detection of clutter objects may result in an unacceptable false alarm rate which lowers the overall accuracy of detection. Advanced EMI sensors that measure the response over a broad range of frequencies or time and apply advanced signal processing have been developed. These have shown to be capable of discriminating between buried land mines and many types of buried metal clutter [3]-[6]. For these advanced EMI sensors to be effective, they must be able to measure the response of a buried target accurately, repeatedly, and quickly which is difficult. These requirements gave rise to development of Wideband EMI sensors that measure the response of the scattered field at a broad range of frequencies and are capable of discriminating between certain types of targets [1]. A typical sensor has a transmit coil that generates a primary electromagnetic field. It has one or more receive coils that measure the induced fields caused by the interactions of the primary field with the surrounding material.

WEMI sensors have been incorporated in many landmine detection systems [7],[8],[9]. As described in the following section much research has also been done to develop methods for detection and discrimination of landmines using data from WEMI sensors as input. In [10], three phenomenological models for wideband electromagnetic induction (EMI) response of buried conductors are presented. The models are based on analytic solutions for spheres, cylinders, and wire loops that can be extended to a variety of targets at all frequencies of interest including matches to theory in the low- and high-frequency limits. As described in [10], the response of a target, with its primary axis being aligned to the principal field of the sensor can be given as,

$$S(\omega) = A(X(\omega) + iY(\omega)) \qquad (1)$$

A three-parameter (amplitude q, a response time constant τ, and a factor s, that controls magnitude of asymptotes at low and high frequencies) model is designed to match targets with compact shape and can be expressed by the following equation,

$$(X+iY)_{p3} = q\left(s + \frac{(i\omega\tau)^{\frac{1}{2}} - 2}{(i\omega\tau)^{\frac{1}{2}} + 1}\right) \qquad (2)$$

This model is extended to non-compact targets with a new model parameter *c* (controlling the capability to fit different shapes) and is given by the following equation,

$$(X+iY)_{p4} = q\left(s + \frac{(i\omega\tau)^{c} - 2}{(i\omega\tau)^{c} + 1}\right) \qquad (3)$$

A five parameter model, specifically designed to match the driving band (soft metal rings) signal is provided as shown below,

$$(X+iY)_{p5} = q\left(s + \frac{(i\omega\tau)^{c} - 2}{(i\omega\tau)^{c} + 1} + b\,\frac{\left(i\omega\tau_{Loop}\right)^{cLoop} - 2}{\left(i\omega\tau_{Loop}\right)^{cLoop} + 1}\right) \qquad (4)$$

The second term in the above equation is the special loop-like signal customized to match response of driving bands by the weight parameter $b(\geq 0)$. The development of these models indicates that, given objects of particular size, shape, and materials, the response can be predicted and modeled.

1.2. Algorithm Literature Review

As mentioned above, WEMI sensors have been used in many landmine detection systems and, hence, research has been done to improve detection and discrimination algorithms using data from WEMI sensors as input. In the following sections a summary of some of the methods developed for WEMI data is presented as well as a summary of methods based on dictionary matching techniques and algorithms developed for landmine detection and discrimination.

1.2.1. Literature Review of Methods for WEMI Data

In [11] an automated anomaly detection method for unexploded ordinances (UXO) in broadband EMI data is presented. This algorithm uses in-phase and quadrature responses at multiple frequencies as features to detect all metal objects while suppressing false alarms. A detector function is defined that reduces the false alarms caused due to sensor height, geology or background noise and it able to detect these anomalies.

Another algorithm presented in [12] reviews the problem of subsurface discrimination using electromagnetic induction (EMI) sensors. This algorithm discriminates based on differences in the multi-axis magnetic polarizability between different objects [12].

EMI sensors are typically used to indicate the presence or the absence of metal targets below ground. However, many landmines are made of plastic and other types of non-metal. Ground Penetrating Radar (GPR) offers the promise of detecting landmines with little or no metal content and can detect discontinuities in the ground. Although GPR systems can achieve high detection rates and work well on non-metallic targets, they are unable to detect metallic targets with similar accuracy. Hence often GPR and EMI sensors are used together and results are fused to achieve a better detection rate while keeping the false alarm rate as low as possible. The relative performance of the EMI and GPR sensors can vary significantly depending on the mine type, geographical site, and soil and weather conditions, burial depth. All these factors can have significant effects on the received sensor data. Thus, multi-classifier, multi-algorithm, and multi-sensor fusion is a critical component in landmine detection specially to be incorporated as a part of an automated system. For example, in [13], an algorithm is presented which does context dependent multi-sensor fusion. The training part of this algorithm consists of context extraction and algorithm fusion. In the extraction phase, the features used by the different algorithms are combined which partition the feature space into groups based on similarity of context. The algorithm fusion component assigns an aggregation weight to each detector in each context based on its relative performance within the context. The confidence value of the individual algorithms and their degree of worthiness in the context are aggregated to compute the final confidence value. The experiments and results show that this fusion outperforms any of the individual detection technique by taking advantage of the strengths of the individual sensors.

Another algorithm presented in [14], uses data from WEMI and GPR in a Hierarchical Mixture of Experts model to increase the landmine detection rate. The EM algorithm (Expectation Maximization) is used to estimate the parameters of the hierarchical mixture. This method classifies the alarms in five classes based on the metal content that makes it possible to train the algorithm for low metal mines.

1.2.2. Literature Review on Dictionary Matching Techniques

Dictionary matching techniques and its variations have been used in many applications. A dictionary is a set of patterns and the goal of a dictionary matching technique is to represent the given information with these patterns in order to find the best fit. This can be of great advantage when applied on data that can be represented by some mathematical model. There are various algorithms that use the dictionary matching techniques in a wide range of applications. Matching pursuits (MP), first introduced by Mallat and Zhang is a well-known technique for sparse signal representation [15]. MP finds the linear approximations of signals by iteratively projecting them over a redundant, possibly non-orthogonal set of signals called a dictionary. The idea behind MP is that a signal can be decomposed into a linear combination of signals that can be represented by a redundant dictionary of functions.

Another similar method is the projection pursuit algorithms [16]. This method finds a model that when estimated from the data, represents the regression surface in a meaningful manner. The regression problem is then reduced to finding a set of parameters. A variation of matching pursuits algorithm is explained in [17], where in it is used in the analysis and synthesis of a phonocardiogram. A phonocardiogram signal is decomposed by projecting the signal iteratively over the function dictionary and by selecting the time- frequency functions which can best match the local structure or waveform of the signal.

Matching pursuits is a greedy algorithm, even though converges asymptotically, and thus the resulting approximations, after finite number of iterations can be sub-optimal. However, it is useful for approximations when it is hard to come up with optimal orthogonal approximations, as in the case of high-dimensional signals or images.

In [18], a particle filter matching pursuit algorithm is presented that adapts the dictionary to the waveform structure. This algorithm uses a sequential Monte Carlo approach, to estimate the dictionary suitable for the decomposition of a given waveform, and then uses the matching pursuit algorithm to decompose the waveform. This is done since matching pursuits can be computationally intensive as it is based on selecting elements from an over complete dictionary spanning the time-frequency plane of interest.

In [19], Genetic Matching Pursuits (GMP) is presented. GMP uses a genetic algorithm [20] to find a suitable dictionary element every iteration. GMP finds a good match for the residue based on some acceptability criteria, instead of iterating over all the dictionary elements to find the best matching dictionary for the residue. This is useful in situations where the size of the dictionary is quite large and iterating over the whole dictionary may be computationally inefficient. One drawback of this algorithm is that it only guarantees the best match in the dictionary up to an acceptance level and, hence, might result in sub-optimal performance as compared to Matching Pursuits.

Since MP is a greedy algorithm, it does not guarantee that the signal representation is the best one possible given a dictionary. The algorithm presented in [21] claims that a tree-based approach can be used to find a better approximation than given by the greedy approaches of the MP and OMP algorithms. Each iteration of MP, instead of picking only the top-matching dictionary element, this algorithm chooses top $K \geq 1$ elements and uses them in parallel to approximate the current residue. This gives rise to a tree of depth j (for j iterations) with each node having K children. The subset corresponding to the smallest residue at leaf nodes is chosen as the representation of the input signal x.

MP has been shown to converge asymptotically [15]. However, since the MP is a greedy algorithm, the approximation of the signal will be suboptimal for finite j (number of iterations) which can results in using more iterations than necessary to reduce the residue below a given threshold. The Orthogonal Matching Pursuits (OMP) algorithm [22] reduces this drawback. OMP gives the optimal approximation with respect to the selected subset of dictionary elements each iteration by making the residue orthogonal to all of the chosen dictionary elements.

Basis Pursuits (BP) is a signal approximation technique that attempts to find a globally optimal representation of a signal over the given dictionary by convex optimization [23]. This algorithm chooses the dictionary elements and coefficients that minimize the l^1 norm between the signal and its approximation.

The performance of the above dictionary algorithms is closely associated with the choice of dictionary for the process. For many applications of MP, parametric dictionaries are preferred. Parametric dictionaries refer to the signals that can be expressed by a well-defined parametric form and, given this form an entire family of signals can be generated by varying the values of its parameters. Parametric dictionaries have an advantage in terms of storage as the parametric equation and the parameters only need to be stored and the entire dictionary can be generated with these. However, in some cases parametric dictionaries may not give accurate and sparse representations of the data. Another problem with using parametric dictionaries is choosing the model parameters [24]. In these cases, sometimes it is more useful to learn the dictionary given training data rather than using a general parameterized form of a dictionary. Learning the dictionary from the data produces sparser solutions with better approximations, while keeping the size of the dictionary manageable. Their only drawback is that they need more storage space than the parametric dictionaries. Some of the methods used to learn the dictionaries are clustering based methods, optimization methods and others [25]-[26]. The size of the dictionary is a factor that determines the performance of the algorithm, however an overcomplete dictionary can be more robust to noisy or degraded signals. On the other side, an over-complete dictionary might have elements too similar in the parametric space and do not provide much distinct information. Hence, in order to reduce repeated elements, clustering methods can be used to filter the unwanted or similar elements.

1.2.3. Literature Review on Methods Applied to Landmine Detection.

As discussed earlier, MP is used for sparse signal representation. MP has been used as a feature extractor to build classifiers that can discriminate between the different types of mines. The dictionary and coefficient information produced by MP approximation of signals is used to build feature vectors that are used to train the classifiers. The general approach to MP based feature extraction is to first find the matching pursuit approximation of each training pattern and then the ordered counts of dictionary elements chosen (the projection sequence) and the corresponding indices used to build the feature vectors at each MP iteration. One advantage of using the MP algorithm for feature extraction is that it not only gives an accurate description of the patterns in terms of its features, but it also provides an explicit control over the dimensionality of the feature vector.

In [27], a variation of MP is used to build a classifier for detection of buried unexploded ordnance (UXO) targets. A subset of the training data that is most informative in characterizing the distribution of the target test site is used to build an information matrix of the training data. The classifier is then trained using another subset of training data whose class labels are the most informative.

The matching pursuit algorithm and its variations have also been applied to landmine detection problems involving wideband EMI data [1]. In [28], the authors have presented a Bayesian approach to the classification problem by building a classifier that assigns confidence probabilistically. Here, classification is reduced to a two class problem that is mine and none- mine and it uses K closest clusters to assign a final confidence to a test signature. However, this approach will assign a confidence to either of the two classes even if a test signature does not belong to either of the classes. In [29], the MP algorithm is used to find approximations of training signals by projecting them onto a dictionary that is made by segmenting the training signals. These approximations are used as prototypes to find the class members by calculating the similarity to all the other prototypes and assigning membership to the most similar one.

A multimodal MP dissimilarity measure is presented in [30], which uses the MP algorithm along with competitive agglomeration clustering to model to compare signals from multiple sensors. In [31], a method called GRANMA (Gradient Angle Model Algorithm), used on wideband EMI data in landmine detection is presented. This method applies gradient angle approach to the parametric models and uses a lookup table to find parameters as opposed to nonlinear optimization. GRANMA algorithm reduces a four parameter model to a two parameter model thereby making the algorithm computationally faster. This method performed statistically similar to other similar methods such as by Fails et al [32] and Xiang et al [33], while presenting a noteworthy improvement in the speed.

In [34], a methodology for target discrimination using wavelet-based and morphological feature extraction is presented that uses metal detector array data as input. The main advantage of this method over certain modeling techniques is that it can adapt to variations in target material and its surrounding environmental conditions. The alarms are detected by using a gradient-based peak recognition method as discussed in [35].

The classification is done using a neural network architecture known as fuzzy Adaptive Resonance Theory Mapping (ARTMAP), also known as FAM (Fuzzy Adaptive Mapping). The FAM neural network is a supervised variant of the ART network that realizes a fusion of ART and fuzzy set theory [36]-[37]. This method does not associate a confidence value during the classification stage that might be a good indication for expressing the confidence level of detection. Since this method mainly focuses on discriminating on bases of target type and burial depth, the number of misclassifications is high, when the metal content and size of the targets is similar.

As discussed above, the main challenge in building an effective landmine system is to have a higher detection rate and at the same time a lower false alarm rate. Some of the clutter present in the earth may have similar characteristics as that of mine making, a 0% false alarm rate is impossible to achieve. However, a balance can be maintained between the detection and the false alarm rate. It may be helpful in maintaining this balance by being able to distinguish between the different types mines based on their properties. This can be achieved through a robust classifier that classifies each mine type and can be used to assign a confidence based on the mine properties.

2. POSSIBILISTIC DISSIMILARTY MEASURE AND K-NEAREST NEIGHBOR CLASSIFIER

The K-nearest neighbor (KNN) classifier classifies an object based on the closest training examples in the feature space. However, the K-NN classifier generally does not take into account the distance of the test object to its nearest neighbor and hence cannot discriminate between neighbors that are equally close and neighbors that are equally far away. For example the crisp K-NN, weighted K-NN, and fuzzy K-NN classifiers as discussed in [38]-[39] do not identify when a point is very far from all training samples and will place test points that are far from training samples with as much weight as those that are close to training samples. This drawback to K-NN classifiers was addressed in [40] where in a possibilistic K-NN classifier is presented for landmine detection that identifies a few prototypes that can capture the variations of the signatures within each class. This classifier uses the edge distribution within the signatures as features, which are then clustered to assign fuzzy memberships to indicate their memberships to either of the mine or false alarm class. A possibilistic K-NN based classifier is used to assign a final confidence value in order to distinguish mines from false alarms. However, this classifier does not use the variations in the signature to discriminate between the different mine types.

In this section, a possibilistic K-NN classifier for WEMI landmine detection is presented. The proposed possibilistic classifier is developed by defining a dissimilarity measure that can be used to identify alarms that do not belong to any of the training classes. By using this approach, the possibilistic K-NN has the ability to assign a weight of zero to all target and non-target classes and, thus, identify points that are unlike any of the previously seen training data. The proposed K-NN classifier makes use of features obtained from the application of the Joint Orthogonal Matching Pursuits (JOMP) algorithm using a dictionary defined from the Discrete Spectrum of Relaxation Frequencies (DSRF) [41] model. In this section, the DSRF model for EMI data and the JOMP algorithm are discussed. Then, the proposed Possibilistic K-NN classifier is described. An overview of the proposed methodology is as follows.

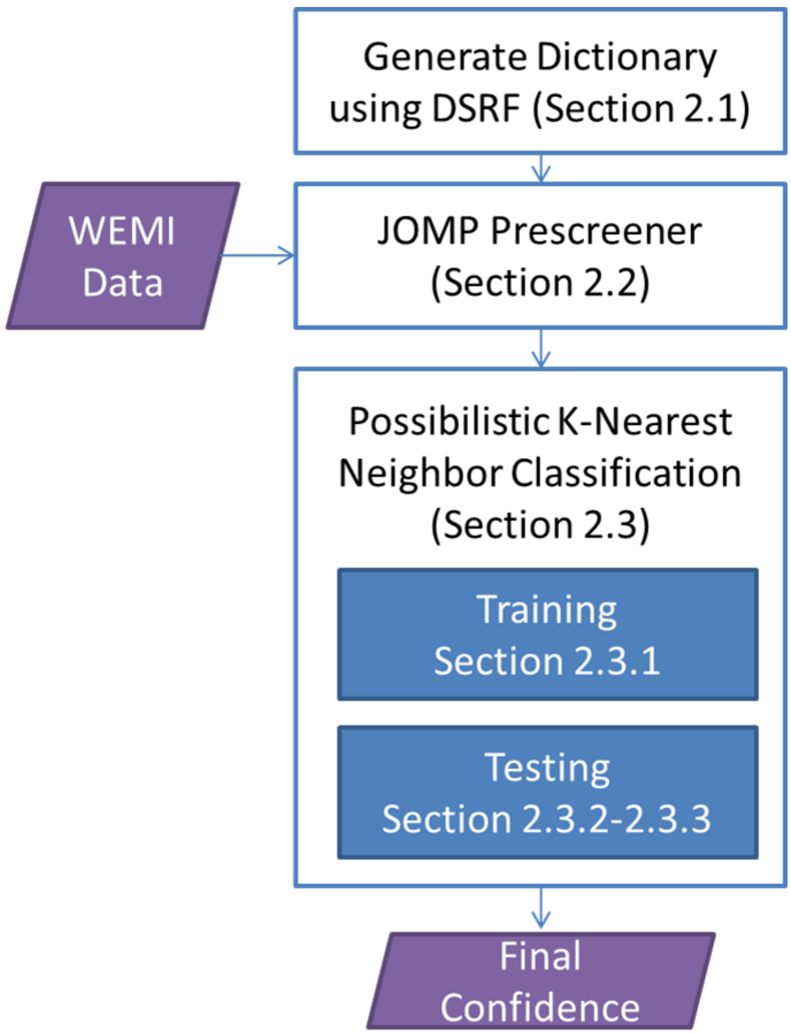

Figure 1: Overview of Possibilisitc KNN Prescreener, Training and Testing

2.1. Discrete Spectrum of Relaxation Frequencies

Dictionary matching techniques have been applied to the response obtained from the electromagnetic induction sensors in order to detect buried targets. The dielectric response of materials has similar properties as compare to the EMI response and, hence, the models developed for dielectric materials can be applied to EMI response [41]. In the proposed method, the dictionary elements used for EMI data are generated using the Discrete Spectrum of Relaxation Frequencies (DSRF) model which can be used to characterize materials from an EMI response [41][1]. The DSRF model, shown in equation *(5)*, is used to produce dictionary elements by varying a large set of possible relaxation frequencies ζ, and using fixed values of ω that correspond to the frequencies measured by the EMI sensor.

$$H(\omega) = c_0 + \sum_{k=1}^{K} \frac{c_k}{1 + \frac{j\omega}{\zeta_k}} \tag{5}$$

To relate the DSRF model to the MP framework, each dictionary element will correspond to a vector d_i as shown in equation *(6)* and the estimated weights will be correspond to the c_k values in the DSRF.

$$d_i = \left[\frac{1}{1 + \frac{j\omega_1}{\zeta_i}}, \frac{1}{1 + \frac{j\omega_2}{\zeta_i}} \cdots \frac{1}{1 + \frac{j\omega_N}{\zeta_i}} \right]^T \tag{6}$$

In addition to the dictionary elements described above, a background dictionary element is generated using the following soil model,

$$R_m(\omega) = p_1 + p_2\left[ln\left(\frac{\omega}{\omega_0}\right) + j\frac{\pi}{2}\right] \qquad (7)$$

The figure below shows the DSRF dictionary generated using the model in equation *(5)*. The horizontal blue line, shows the soil response as modelled by equation *(7)*

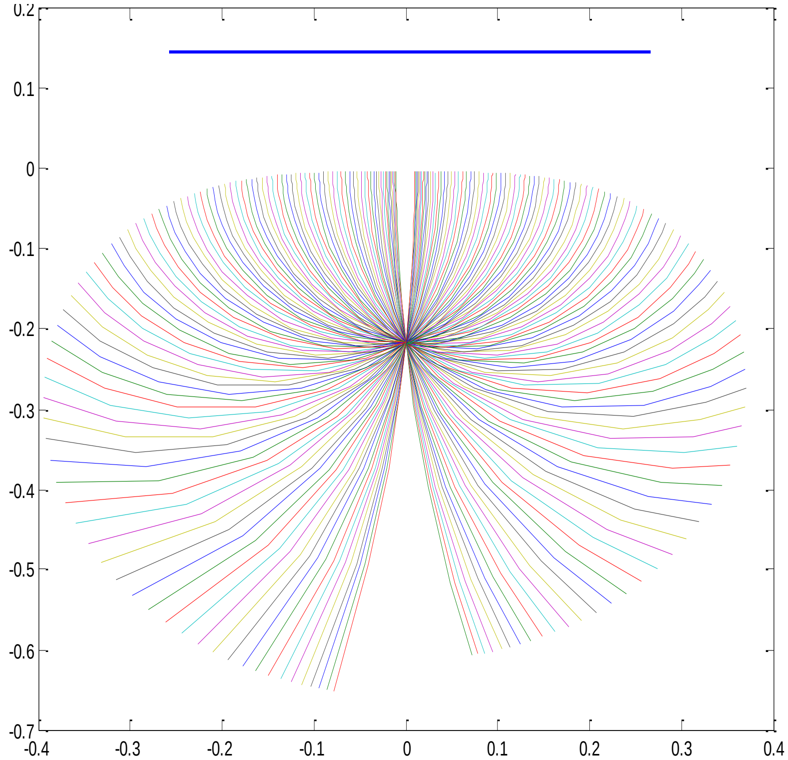

Figure 2: DSRF Dictionary along with the horizontal (blue line) soil model

2.2. Joint Orthogonal Matching Pursuits

As shown in Figure 1 the Joint Orthogonal Matching Pursuits [42] algorithm (JOMP) is used as a prescreener for the proposed possibilistic K-NN classifier. JOMP is an extension of MP technique for sparse signal representation. In MP, a signal, x_j is represented with dictionary elements, $D = \{d_i\}_{i=1}^{M}$ using a sparse, linear combination,

$x_j = \sum_{k=1}^{m} w_{kj} d_k$ where $m << M$. An iterative, greedy approach is used to estimate the weights, w_{ij} in the sparse representation.

The pseudo-code below summarizes the MP algorithm:

Matching Pursuits (MP) Algorithm

Inputs: x_j, D

Parameters: Stopping Threshold, t

Set R = x_j, $r = \|R\|_2$, $m = 1$

while ($r > t$)

    for $i = 1,\ldots,M$

        Compute $s_i = \langle R, d_i \rangle$

    end

    Set $k_m = \underset{i}{\operatorname{argmax}}\, s_i$ *// Identify the dictionary element with the largest projection value*

    Set $w_m = s_{k_m}$

    Set $R = x_j - \sum_m w_m d_{k_m}$ *// Update the residual using the first m weighted dictionary elements*

    Set $r = \|R\|_2$

    Set $m = m+1$

end

At every step, the dictionary element that is the most similar to the residue is chosen and subtracted from the current residue. If the angle of projection at each iteration is small, then it will require fewer iterations to drive the residue to zero. Conversely, if the angle of projection between the residue and the chosen element is large, it will take more iterations and dictionary elements to reduce the residue significantly and below a certain acceptable level. In addition, if the dictionary is large, then the computation time of the iterations will be large. Hence, the proper choice of dictionary is essential. Since MP is a greedy algorithm, the chosen coefficients should get smaller as the iteration index, increases. Hence, the maximum information about the signal is contained in the first few coefficients.

JOMP extends the MP method by considering multiple signals simultaneously. JOMP uses the same subset of dictionary elements to estimate the sparse representation of a set of signals [42]. Therefore, the same sequence of dictionary elements is used to represent a set of data points. For example, JOMP can be used to estimate a sparse representation of several data points around a central point of interest as shown in Figure 3. Figure 3, shows the central point x_i for which JOMP confidence is assigned by looking at two points $r_1 = x_i - q$ and $r_2 = x_i + q$ that is, q scans behind and q scans forward.

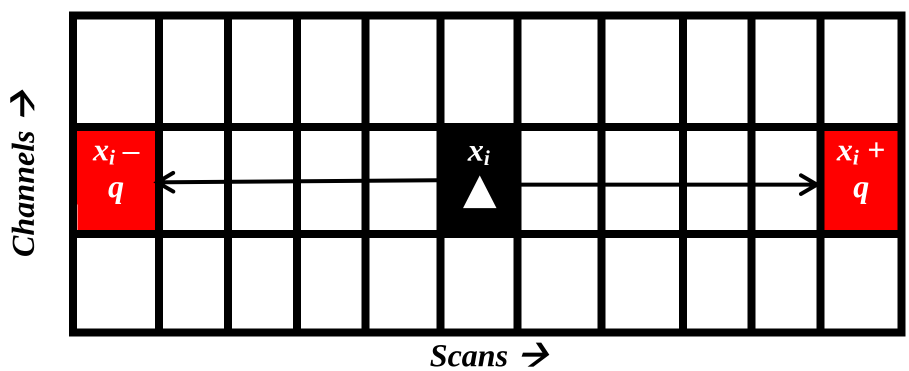

Figure 3: JOMP confidence assignment for a point

In the JOMP algorithm, a confidence value can be assigned based on the residual errors of the points being jointly considered. The confidence value is computed to the central point using the following equation,

$$c_x = \frac{1}{1 + \frac{1}{n}\sum_j r_j} \qquad (8)$$

where r_j is the residual error for the j^{th} point being considered simultaneously. This confidence provides a measure of how well the dictionary can represent the points under consideration.

2.3. Possibilistic K-NN Classifier for WEMI Landmine Data

Different buried objects have specific signatures depending on their metal content, size, shape, depth and material properties and, hence, they are often consistent in choosing dictionary elements. The frequency response of targets is presented on an argand diagram (in the complex plane) where in, the imaginary part is plotted on the vertical axis and the real part on the horizontal axis, with frequency as the parameter.

Figure 4 shows the signatures for different buried objects and consistency in selecting dictionary elements on argand diagram. Each row of the Figure 4, is a particular target type that has a similarity in the EMI response. In the proposed approach, this fact is used in order to discriminate the different types of buried objects.

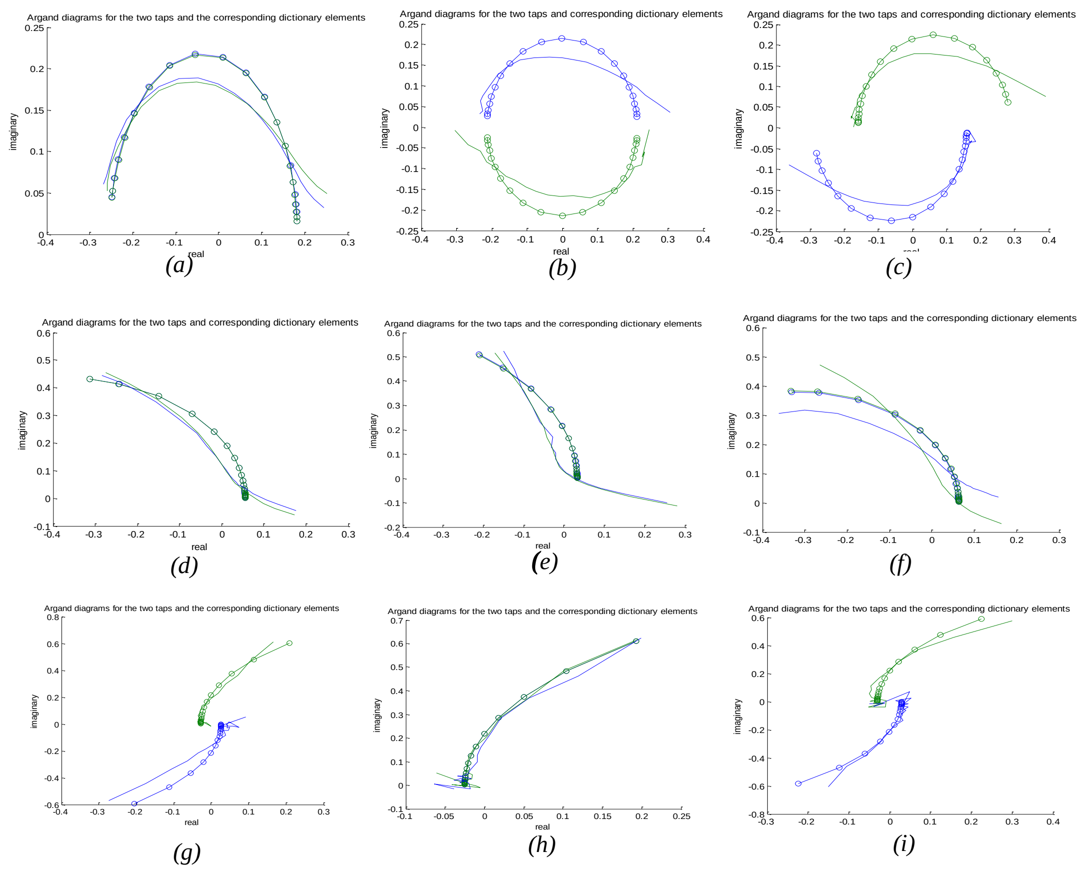

Figure 4: (a)-(c) Argrand diagrams for the two taps and the corresponding dictionary element for target type I (d)-(f) Argrand diagrams for the two taps and the corresponding dictionary element for target type II (g)-(i) Argrand diagrams for the two taps and the corresponding dictionary element for target type III

In spite of most targets having distinguishable properties, clutter objects in the data may be identified increasing the false alarm count. Research has been conducted in using the difference in the properties of targets to discriminate between different target types as well as against clutter in order to reduce the false alarm rate.

Some of these techniques involve modeling the target responses using different models such as body of revolution with two principal coordinates [43], in-phase and quadrature measurements for selected sampled frequencies [44], and decay rate, energy, and axis decay rate symmetry measures [45]. Many techniques involve characterizing the target responses according to a particular property and comparing them to a library of known target responses.

As discussed above, for a landmine detection system, it can be necessary or helpful to discriminate between the several sub-classes of targets [34]. Many discrimination-based classifiers are unable to accurately classify outliers in test data. Discrimination-based classifiers learn the discrimination rule by estimating a decision boundary between training examples of the two or more classes. Therefore, these classifiers are able to accurately classify the test points that belong to somewhat similar distribution as the training data. However, for the landmine detection problem, the inability to detect outliers, leading to mislabeling can prove to be quite costly.

In order to provide discrimination to a multi-class problem and to include the possibility of a test alarm belonging to none of the classes, a K-NN classifier using a possibilistic dissimilarity measure is presented. The proposed classifier takes advantage of the consistency in selecting a dictionary element by a particular target type as shown in Figure 4. A training data is collected which is used to collect frequency of dictionary elements selected per target type. These counts further validate that they could be useful features in order to distinguish the alarms according to their target types. The normalized dictionary counts are calculated from the training data. The accuracy of the classifier data depends on the training data and it increases as the number of training examples increases. However, higher number of training data increases the computational complexity too. Usually, there are more examples of clutter class as opposed to actual mine types. This can create a bias in the output, since the training examples are not balanced in numbers.

In the proposed method, detection has been broken down into four phases: prescreening, feature extraction, confidence assignment, and decision making [46]. Prescreening involves a quick run on the data to generate some alarms, which are points of interest. For the data used in this paper, the pre-screener creates alarms using Joint Orthogonal Matching Pursuits [42] as discussed in the previous section. The dictionary counts of these alarms are the features that are extracted during the feature extraction phase. The proposed algorithm is then run on JOMP alarms in order to assign confidences and make a final decision as to whether the alarm is mine or non-mine.

To sum up, the algorithm uses JOMP as a prescreener and dictionary matching as a feature extractor in order to discriminate different types of targets. The ultimate goal would be to use this algorithm as a part of an automated decision system capable of discriminating different types of targets as well as non-targets such as clutter and possibly infer information related to the detected target with reduced false alarm rates.

Possibilistic clustering assigns typicality for a test point to one or more clusters. The main advantage of using a possibilistic clustering algorithm as opposed to hard K means or fuzzy clustering is that, possibilistic methods can assign typicalities for a point to a class without considering its membership or distance to any other class. In case of landmine detection, some mines may have features that can give it equal membership to two classes or during the testing, if the classifier encounters an alarm that has not been seen in the training or that looks different from all the existing classes, then the classifier can identify that mine and give it low typicality to all the classes. In this way, alarms are not forced to belong to at least one of the classes and can identify outlier points.

The proposed classifier uses a possibilistic measure to assign typicality to each class. In order to have a measure of the closeness of a test point to a mine class, we use a unique dissimilarity measure that makes the weight assignment possibilistic. For each test point in the alarm blob, this measure uses the reciprocal of the difference in the residual error between a test point and the best dictionary element that projects on to it. The dictionary counts as shown in Figure 9 show some correlation within the neighbor indexes since these elements are chosen in a logarithmic progression. For example, mine type 1 chooses dictionary indexes in the range of 60-65. Hence for every point in the alarm blob, the K-most nearest neighbors are considered. The training, testing and final confidence assignment is explained in more detail in the subsequent paragraphs.

2.3.1. Training

Training data for the proposed method is comprised of alarms obtained from the JOMP_2TAP prescreener along with their labels whether belonging to mine or non-mine classes. This prescreener also gives us a dictionary index map, which indicates the dictionary index chosen by every point in the blob. We use this dictionary index map in order to get the overall dictionary counts for the entire data set. The mine signatures obtained from the JOMP_2TAP prescreener have different EMI characteristics and hence can belong to multiple subclasses according to their size, shape, depth and the way they are buried. Similarly many buried objects such as rocks, metal objects, plastics can correspond to a variety of clutter characteristics. Hence we use a count of the dictionary element selected by each mine type in order to classify the buried objects into several subtypes. In order to get the dictionary counts, for each alarm type, dictionary element chosen at each pixel in alarm blob is stored in a matrix and added over all the alarms. Thus at the end, a matrix is obtained which has the counts of the dictionary elements per target type. In order to solve the bias problem, these counts are normalized. The normalized dictionary counts, D, are found by computing $n_{rj,}$ the number of pixel locations from the r^{th} target type represented with j^{th} dictionary element in the training data, and normalizing across dictionary elements,

$$d_{rj} = \frac{n_{rj}}{\sum_k n_{rk}} \tag{9}$$

Let this matrix be represented by D,

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1k} \\ \vdots & \ddots & \vdots \\ d_{r1} & \cdots & d_{rk} \end{bmatrix} \tag{10}$$

such that,

$$\sum_k d_{rk} = 1 \tag{11}$$

These dictionary counts are used as features that give an idea of the distribution of the data and the discriminative power of WEMI sensors. These features are used to train the classifier.

The pseudo code for the training phase is as shown below

Training- Dictionary Counts

<u>Inputs</u>: *JOMP alarms* $A=\{a_1, a_2, a_3 \ldots a_n\}$

<u>Set</u> *n = Number of alarms*

for *i = 1…n*

Compute

$\boldsymbol{n_{rj}}$ *// number of pixel locations from the* r^{th} *target type represented with* j^{th} *dictionary element*

end

Normalize the count using equation *(9)*

<u>Output</u>: *D* - Counts of the dictionary elements as shown in equation *(10)*

2.3.2. Testing

Let $A=\{a_1, a_2, a_3 \ldots a_n\}$ represent the set of alarms created by the pre-screener and let *r* denote the different target classes and *k* denote different dictionary elements. For each test alarm a_i, the dictionary indexes selected by each pixel in the alarm blob are determined. The *K* most frequently selected dictionary elements for the alarm are identified. Next, the residual error of every point in the alarm blob is calculated when projected on those *K* dictionary elements. The residual error is used as a possibilistic distance measure between the dictionary element and each pixel in the alarm. This distance measure quantifies the decision criterion: if a pattern is near to any of the training patterns then it should get a high weight for belonging to that class and if a pattern is far away from the training patterns in the feature space then it is unlikely to be a part of any of the training classes. The mean of these residuals errors are then averaged and used, in conjunction with the normalized dictionary counts learned in training, to assign a confidence value for the alarm in each target type. Each of the K neighbors are assigned a possibilistic weight as shown below,

$$W_{ik} = \frac{1}{1 + \left[max\left(0, \left(\frac{R_{ik} - \eta_1}{\eta_2}\right)\right)\right]^{\frac{2}{m-1}}} \qquad (12)$$

Where W_{ik} is the possibilistic weight of the alarm to the k^{th} neighbor, R_{ik} = mean residual of the alarm a_i to the k^{th} dictionary element, η_1, η_2 and m are parameter values. In the current implementation, η_1 is set to mean of all the residuals, η_2 set to thrice the standard deviation of the residuals and m is a fuzziness parameter. In the current implementation, this parameter is set to 2.

This method of weight assignment is motivated from the weight assignment done in [40]. However in our implementation it is significantly different from the way it is applied in [40], since in there the weight is used to calculate the distance to each of the training prototypes and in our method this distance is used to find the residue of the i^{th} pixel in the test alarm to the K^{th} dictionary element. This method of weight assignment can allow non-typical test samples to get low weights in all the classes.

The confidence assignment for an alarm in a particular class is done as follows,

$$P_r(a_i) = \frac{1}{K}\sum_{k=1}^{K} d_{rk}W_{ik} \tag{13}$$

where $P_r(a_i)$ is the possibilistic confidence of the i^{th} alarm in the r^{th} target class.

The pseud-code for testing is as shown below:

Testing- Possibilistic K-NN Classifier Algorithm

<u>Inputs</u>: *JOMP alarms A=*$\{a_1, a_2, a_3 \ldots a_n\}$*; Normalized dictionary counts D*

<u>Set Parameters</u>: *η_1, η2, m, K, n = Number of alarms*

for *i = 1,…,n*

Pick K most frequently selected dictionary elements

for *k = 1,…,K*

Compute $\boldsymbol{R_{ik}}$ *// mean residual of the alarm a_i to the k^{th} dictionary element,*

Compute $\boldsymbol{W_{ik}}$ *// assign a weight to each neighbor using (12)*

end

Compute $\boldsymbol{P_r}(\boldsymbol{a_i})$ *// possibilistic confidence of the i^{th} alarm in the r^{th} target class using (13)*

Compute $\boldsymbol{F}(\boldsymbol{a_i})$ *// final confidence value for test alarm using (14)*

end

<u>Output</u>: Confidence Value for an alarm specifying it to be either a mine or false alarm.

2.3.3. Final Confidence Assignment

In the previous section, a possibilistic weight is computed for each alarm for every target type to allow for multi-class classification. For comparison with the JOMP prescreener results and to evaluate overall landmine detection performance, a final confidence is assigned to each alarm and it is labeled as either a mine or false alarm. In our implementation, the final confidence assignment also uses the JOMP_2TAP pre-screener confidence. Confidence is assigned as follows,

$$F(a_i) = \sum_{r \epsilon M} P_r(a_i) J(a_i) \qquad (14)$$

where $F(a_i)$ is final confidence value for test alarm, M is group of mine types and $J(a_i)$ are the confidences assigned to the test alarm during the prescreening stage to each test object.

The sum of possibilistic weights for a mine will high when the alarm is actually a mine and low when it is a false alarm. This assignment makes sure that a mine gets high confidence value, whereas a non-mine gets low confidence due to the low additive weight. The JOMP prescreener confidence is also used so that the results of JOMP which is inversely proportional to the residue

3. DICTIONARY ESTIMATION USING CLUSTERING

In recent years, attention has been focused on adaptively choosing a dictionary so that a given class of signals has sparse coordinates. As discussed in the previous section, the dictionary elements used for EMI data are generated using the Discrete Spectrum of Relaxation Frequencies model as shown below,

$$H(\omega) = c_0 + \sum_{k=1}^{K} \frac{c_k}{1 + \frac{j\omega}{\zeta_k}}$$

This is a parametric dictionary since the entire set of dictionary signals can be generated by varying the parameters. Similarly, a few other parametric dictionaries have been proposed in [47], [48]. These dictionaries have been specifically designed to approximate the wave-forms scattered from the targets of interest. The biggest advantage of using a parametric dictionary is it requires very little storage space.

Although parametric dictionaries may have a storage advantage, it is hard to come up with a well-suited model that accurately represents the data. Instead of using a general parameterized model to express the dictionary, learning the dictionary from the training data may be able to give better results. The frequency distribution of the dictionary elements generated from the DSRF dictionary in Figure 9 show that many of the dictionary elements are not selected by any alarm. The size of the dictionary is an important factor that governs that computational efficiency of the MP algorithm. Therefore, the MP dictionary may be estimated from the training data to have a concise dictionary that is better representative of the data.

In the following section, we discuss learning the dictionary using K-means clustering. This dictionary is used inside the classifier to discriminate between classes. An overview of the way the clustered dictionary is used inside the classifier is as follows

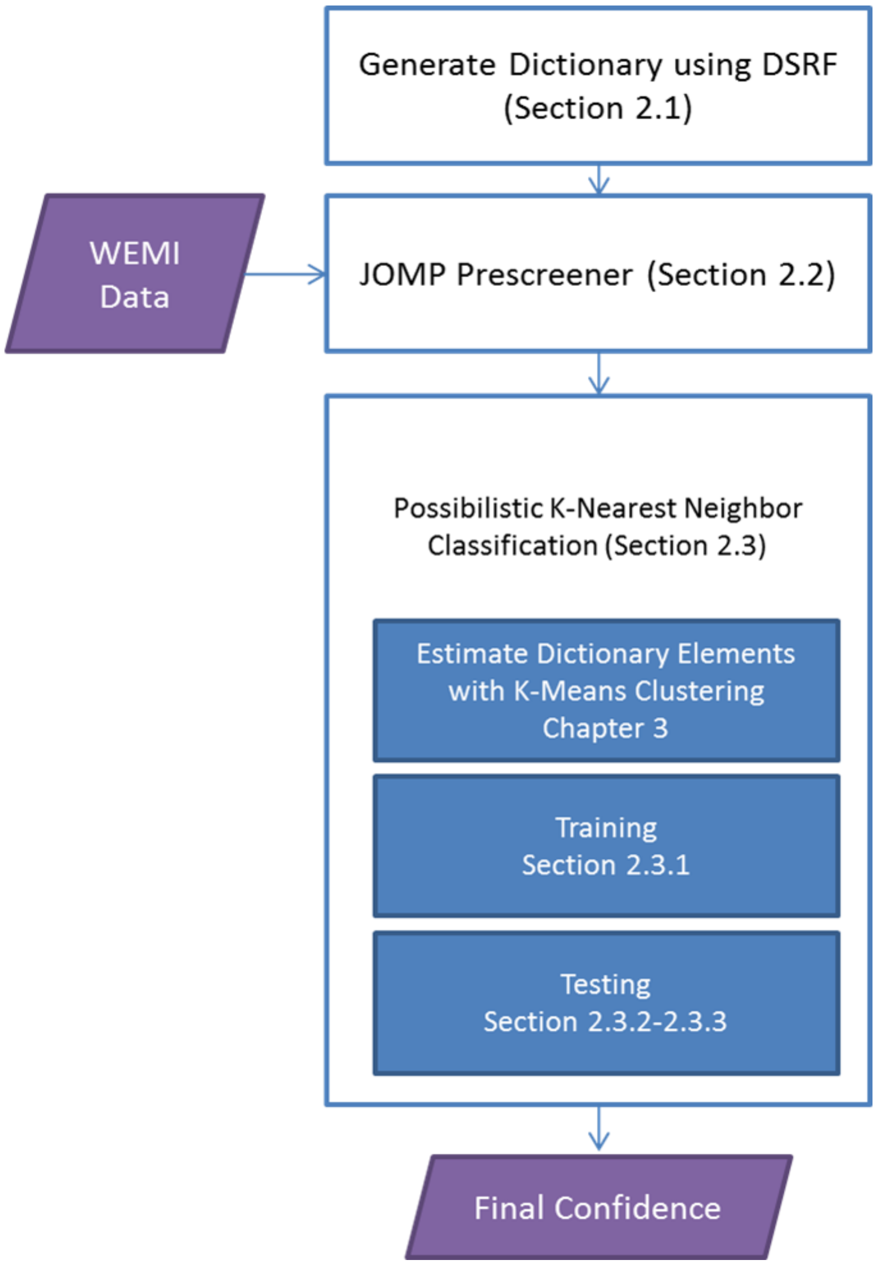

Figure 5: Overview of Possibilistic KNN classifier with K-Means Clustering Dictionary Estimation

3.1. K-Means Clustering to Estimate Dictionary Elements

Several clustering based approaches have been developed over the past few years with the increasing interest in having a dictionary tailored to the data. In [29] the dictionary is built from training data by segmenting signals to extract shape information. The dictionary obtained in this manner may have many elements of similar shape with different displacements. Therefore, these duplicate dictionary elements need to be removed. A two-threshold sequential clustering algorithmic scheme (TTSAS) [49] is used to remove the redundant dictionary elements by clustering similar elements together. The cluster centers obtained are used as the elements building the dictionary.

The K-SVD algorithm provided in [50], generalizes the K- means algorithm by relaxing the constraint of having a unit coefficient for each dictionary atom. Another novel dictionary training algorithm, named method of optimal directions (MOD), is presented in [51],[52]. This method follows more closely the K-means outline, with a sparse coding stage, followed by a dictionary update.

For most applications, while learning dictionary merely finding the cluster centers that best fit the data as in K- means is not suffice. In [54], another algorithm that uses clustering along with sparse modeling and dictionary learning is introduced. The data is modeled as distributions around discrete points and a set of dictionaries are optimized, one for each cluster, for which the sparse representations of the signals are best reconstructed.

3.2. Algorithm implementation and Experimental Set-up

In order to find the dictionaries using clustering, the training data was divided according to the class types. A simple K- means clustering was done on the data to find prototypes that matches most target types and that can be used to sparsely represent the targets in an accurate way.

In order to find the optimal number of clusters, the number of clusters was varied around 30 to 60 clusters, after which the number was varied by small increments. As seen from the dictionary counts obtained using the DSRF dictionary about 35- 40 dictionary indices are never selected. Hence the number of clusters was selected at 60. Even though with this many number of clusters, there are some redundant element, the results obtained show a fair performance improvement than using fewer number of dictionary elements.

The dictionary obtained was then used to perform the JOMP algorithm and dictionary maps for each alarm were created. Using the dictionary maps, counts of the dictionary indices chosen were determined which were used as features for training the classifier.

The pseudo-code below summarizes the implementation.

Dictionary Estimation and Training Approach using K-Means Clustering

Inputs: *JOMP alarms A=*$\{a_1, a_2, a_3 \ldots a_n\}$

Set *n = Number of alarms*

Divide the alarms A into groups as per their class type

Cluster the training data using K-means and obtain the centers to be used as dictionary.

Perform JOMP with the clustered dictionary

for *i = 1…n*

Compute

n_{rj} *// number of pixel locations from the* r^{th} *target type represented with* j^{th} *clustered dictionary element*

end

Normalize the count using equation *(9)* and use them as features for the classifier.

4. DICTIONARY ESTIMATED FROM LINEAR COMBINATION MODEL

As discussed in the earlier chapter, for data that is difficult to represent using parametric model, dictionaries estimated from the data give better approximation of the data. There are several methods that can be used for learning the dictionary such as clustering, convex optimization etc. As mentioned in the previous chapter, simple clustering approach like the K-means [53] can be used to cluster the data to find prototypes corresponding to the signatures in the data. However K-means needs to be provided the optimal number of clusters in order to cluster correctly. Knowing the optimal number of dictionary elements beforehand is a challenging process. Also using the K means algorithm merely clusters the data to find the optimal cluster centers. When K-means clustering was performed on WEMI data, it was seen in the dictionary maps that the choice of dictionary elements within an alarm blob was very much inconsistent. This is due to the fact that clustering the data results in cluster centers quite close to each other.

Figure 6 illustrates this by reconstruction a signal with all the dictionary prototypes obtained from K- means. As can been seen, more than one dictionary element seems to provide good reconstruction.

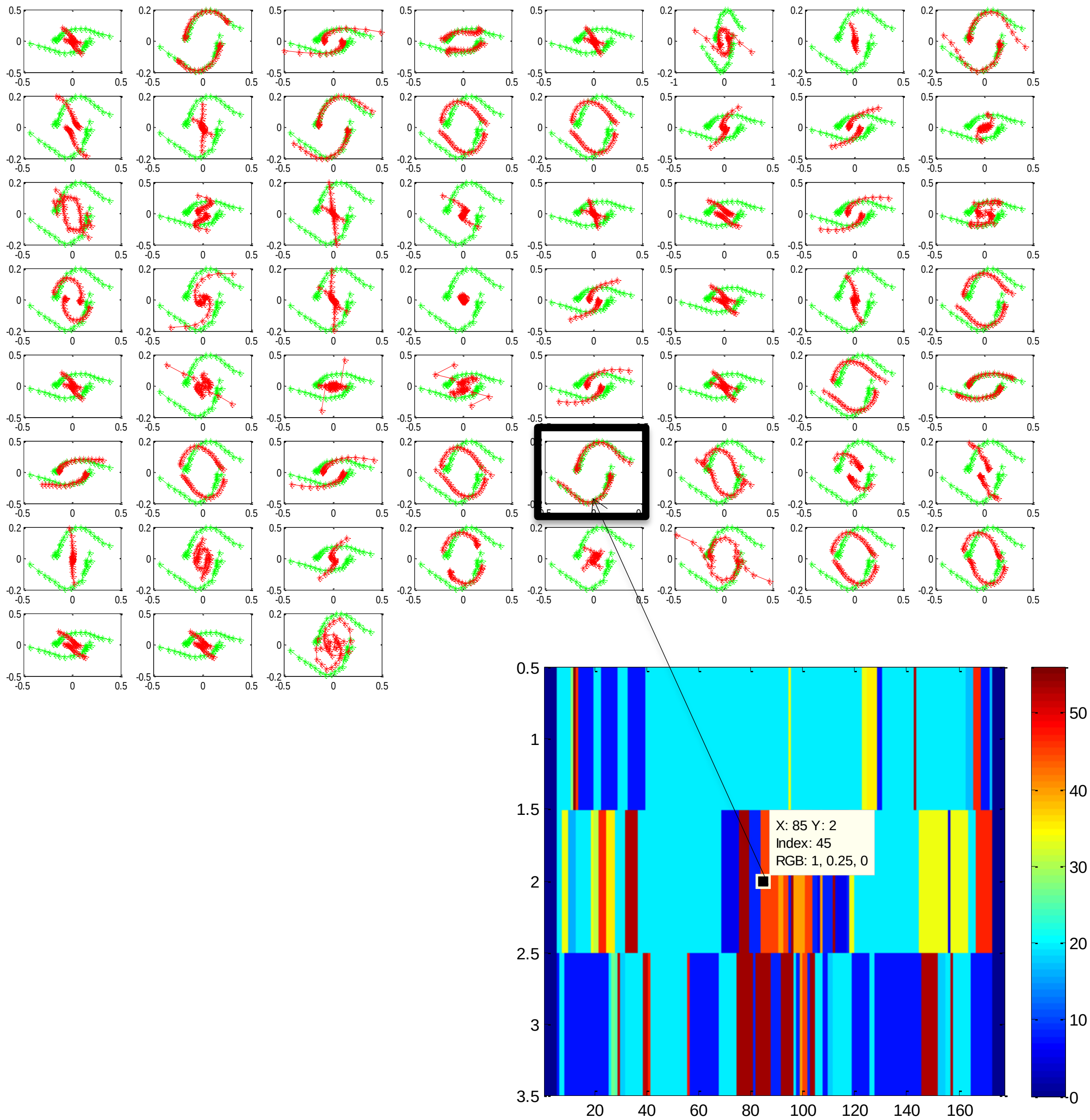

Figure 6: Dictionary map (Target type 1) and reconstruction with all the cluster prototypes obtained

In order to obtain dictionary elements that better represent the training data, the dictionary is estimated by minimizing the error using the linear combination model from the JOMP algorithm. This is done by performing a least squares minimization of the residual sum of squares (RSS) objective function.

The equation for the RSS objective function is as shown below,

$$RSS = \sum_{i=1}^{N} \left(X_i - \sum_{k=1}^{M} p_{ik} E_k \right)^T \left(X_i - \sum_{k=1}^{M} p_{ik} E_k \right) \qquad (15)$$

where, M is the number of dictionary elements to be learnt, p_{ik} is weight of the dictionary element k, in i^{th} data point, E_k is the k^{th} dictionary element. In our implementation we learn one dictionary element per target type. The above equations can be used to find the optimal dictionary by dropping the summation across multiple dictionary elements and minimizing the error between the data and the dictionary. The RSS equation for finding the optimal dictionary is thus given as follows,

$$RSS_{dict} = \sum_{i=1}^{N} (X_i - p_i E)^T \, (X_i - p_i E) \qquad (16)$$

Given dictionary estimates, the weights for each data are estimated. For the first iteration, dictionary estimates are made using K-means. Then the weights are updated using equation *(17)* followed by dictionary element update using *(18).* This iterative procedure is continued until the value of RSS_{dict} is smaller than a tolerance value.

$$p_i = \{E^T E\}^{-1} E^T X \qquad (17)$$

$$E = \{p^T p\}^{-1} p^T X \tag{18}$$

The following section 4.1 shows the pseudo code and section 5.4 discusses the experiments and results performed using this dictionary learning method. The algorithm flow chart of incorporating the model-based dictionary estimation with the Possibilistic K-NN is shown below:

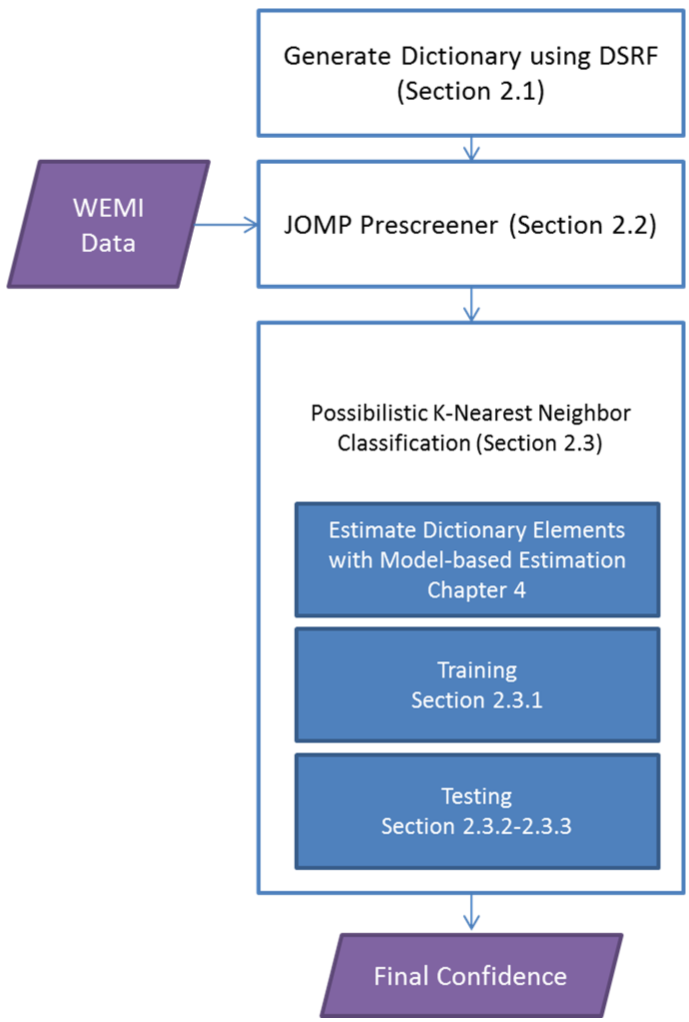

Figure 7: Overview of Possibilistic KNN classifier using Model-based Dictionary Estimation

4.1. Pseudo code for Model-based Estimation of Dictionary Elements

The pseudo code summarizing the model-based estimation approach for dictionary learning is as shown below:

Model-based Dictionary Learning and Training Algorithm

<u>Inputs</u>: *JOMP alarms A=*$\{a_1, a_2, a_3 \ldots a_n\}$

<u>Set</u> *n = Number of alarms*

Divide the alarms A into groups as per their class type.

Initialize dictionary elements using K-Means clustering

Minimize equation (16) iteratively to find the dictionary elements that fit the data.

Perform JOMP with the estimated dictionary

for *i = 1…n*

Compute

$\boldsymbol{n_{rj}}$ *// number of pixel locations from the r*th *target type represented with j*th *clustered dictionary element*

end

Normalize the count using equation *(9)* and use them as features for the classifier.

5. EXPERIMENTS AND RESULTS

This section describes the various experiments conducted to verify the robustness of the algorithm as well as to compare the proposed method with some of state-of-art algorithms. The results on using the proposed classifier with clustered dictionary as well as model based dictionary are presented and the sensitivity of these methods to the number of clusters and different initializations is provided as well.

5.1. Data Set Description

For all the experiments in the subsequent sections, WEMI data obtained from Wideband Electromagnetic Induction Sensors as discussed in section 1.1 has been used. The sensor measures the responses 21 frequencies that are approximately logarithmically distributed over the frequency range of 300 Hz-90 kHz. Hence ω_{min} and ω_{max} are 2073.4 and 565675.1 respectively which corresponds to the above frequency range. The range of ζ for estimation is chosen such that $\zeta_{min} \approx \omega_{min}$ and $\zeta_{max} \approx \omega_{max}$. Hence, $\log(\zeta_{min})$ and $\log(\zeta_{max})$ are set to 2.4470 and 6.6223 respectively, corresponding to a frequency range of 45 Hz–670 kHz, which is larger than the measured frequency range [35].

This dataset has 16 types of target out of which some are metal clutter. For each type of landmine, the WEMI response is collected from several targets buried at different depths and location. A distribution of the target types is as shown in Table 1. This table also includes the classification based on metal content such as non -metal, low-metal and high- metal. The purpose field classifies the targets as either anti-tank (AT) or anti-personal (AP) type. Each of these targets are encountered multiple times in the dataset and hence the total and unique encounters are also noted in the table.

Name	**Content**	**Purpose**	**Total Encounters**	**Unique Encounters**
Target Type 1	Low	AP	29	8
Target Type 2	Low	AP	30	8
Target Type 3	Low	AT	30	8
Target Type 4	Low	AT	28	7
Target Type 5	Low	AT	30	8
Target Type 6	Non	AT	30	8
Target Type 7	Non	AT	34	8
Target Type 8	Metal	AT	9	3
Target Type 9	Low	AT	32	8
Target Type 10	Low	AT	40	11
Target Type 11	Non	AT	41	11
Target Type 12	Non	AT	28	8
Target Type 13	Metal	AT	3	1
Target Type 14	Metal	AT	10	3
Target Type 15	Metal	AT	6	1
Target Type 16	Metal	AT	12	3
Total			**392**	**104**

Table 1: Target Distribution in WEMI data

5.1.1. Dictionary counts on WEMI Data

The training stage of the proposed algorithm is comprised of obtaining the counts of the dictionary elements chosen per target type. The total counts for the data set used are shown in Figure 3. These counts show the consistency in selecting dictionary elements by targets. However, the false alarms seem to choose many of the available dictionary elements. This consistency shows the motivation behind using this classifier. The dictionary counts are normalized across all the elements to get the normalized counts as shown in Figure 4 and are used in the classifier.

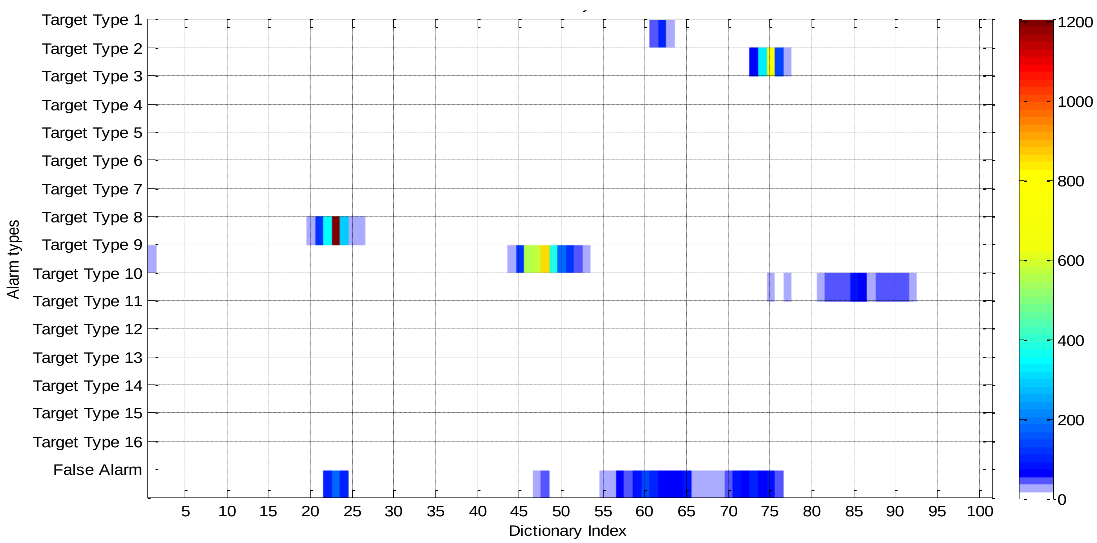

Figure 8: Dictionary counts for different target types

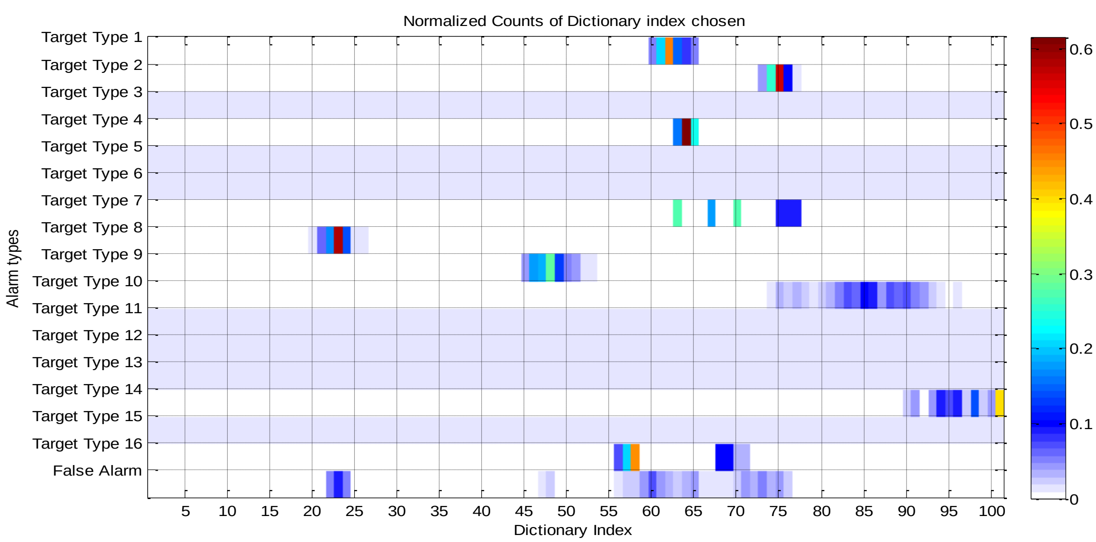

Figure 9: Normalized Dictionary counts

The dictionary maps and the corresponding argand diagrams shown below for certain target types also show that the blob around a target chooses dictionary elements consistently. These maps further validate the use of dictionary counts as a good feature for discriminating the target types

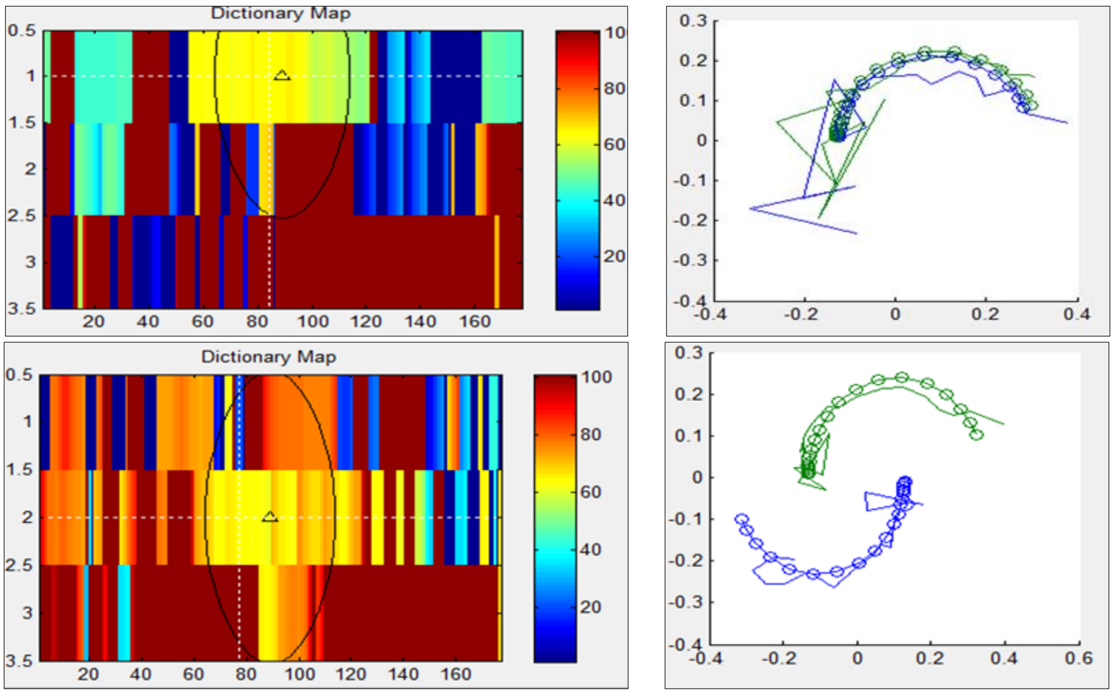

Figure 10: The dictionary maps and corresponding Argand diagrams for Target type 5

5.2. Experiments and Results with the DSRF Dictionary using the Possibilistic K-NN classifier

In order to validate our results, we ran the classifier on the WEMI dataset described above. The proposed method was applied to alarms generated using the JOMP algorithm on the WEMI data. To report and compare results a Receiver Operating Characteristics Curve (ROC) is used to plot the false positives (x – axis) versus the true positives (y – axis). The x-axis shows the false alarm rate (FAR) and the y-axis shows the percentage of detection (PD). In Figure 11, the blue curve shows the JOMP algorithm performance and the red curve shows the classifier performance. As can be seen from the ROCs, our method shows a significant improvement in the classifier performance at the smaller false alarm rates and goes straight up until a PD of 21% as opposed to JOMP which classifies some of the false alarms as mines, giving them a very high confidence. The area under the red curve is significantly larger than the area under the blue curve, thus showing a better detection.

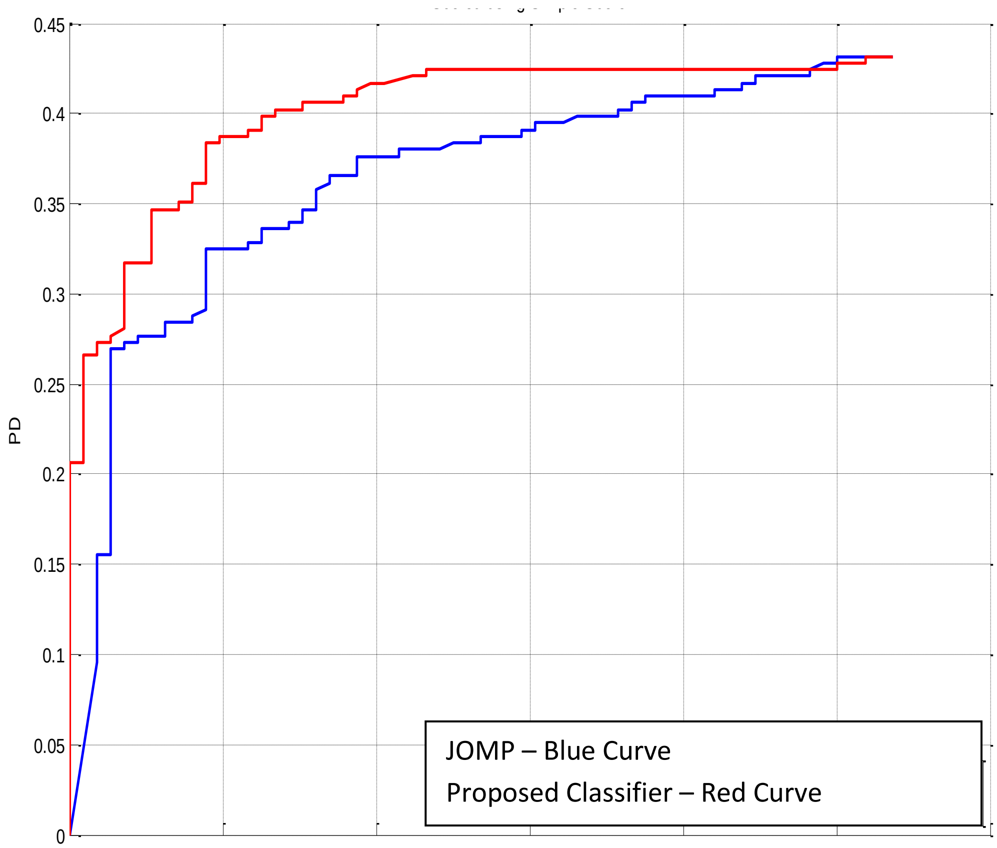

Figure 11: ROC obtained using the proposed method for the test dataset. The blue curve shows the JOMP prescreener ROC and the red curve shows the proposed classifier results.

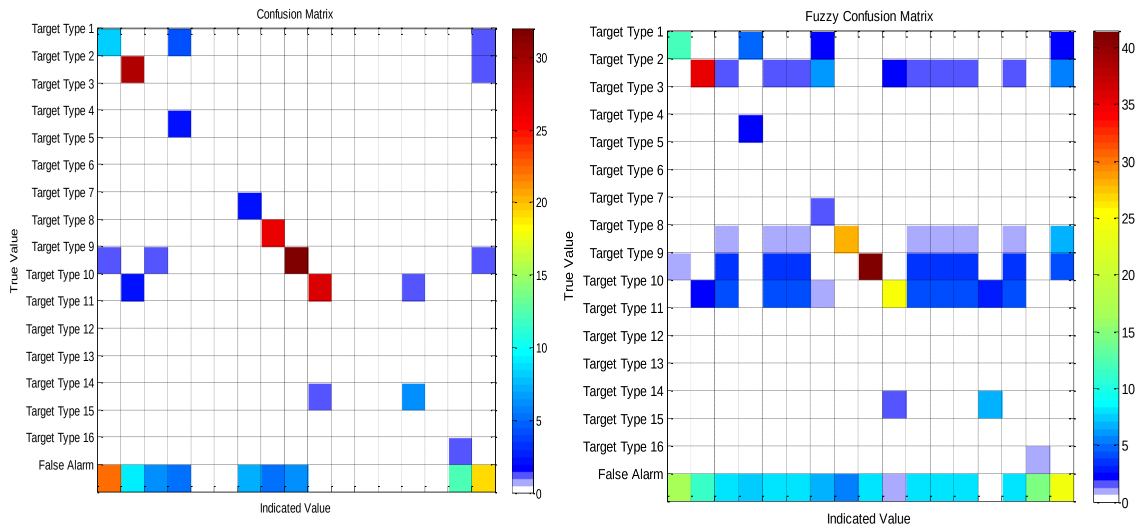

Figure 12(a): Confusion Matrix showing the classification accuracy of 64.14% among 17 target type. (b) Weighted confusion Matrix showing the sum of the possibilistic weights

Table 3 shows the comparison at a fixed false alarm rate (FAR) between JOMP and our classifier. There is a 76% bump in the PD at this FAR. These are test on train results which consisted of 237 total alarms. The first sixteen types are mines and the 17th type is false alarm. These are distributed as follows:

Target Type	Count
1	13
2	30
3	0
4	2
5	0
6	0
7	2
8	26
9	35
10	30
11	0
12	0
13	0
14	7
15	0
16	1
17	91

Table 2: Counts of different target types. Some targets types have zero count since those types are not found by JOMP prescreener algorithm

Method	PD at a fixed FAR value
JOMP_2TAP	0.1550
Proposed Classifier	0.2730

Table 3: Comparison between JOMP and proposed classifier at fixed FAR

5.2.1. Confusion matrix between 17 target types

The confusion matrix in Figure 12(a) shows the actual labels on the *y*-axis and the indicated value on the *x*-axis. For a total of 237 targets, the accuracy of classification is 64.14% within the 17 types. This confusion matrix is created by selecting the class that gets the maximum possibilistic weight as the indicated mine class. This confusion matrix is colored such that the blue shades represent low values and red shades show higher values. The confusion matrix shows dark shades of red along the diagonal meaning most alarms were classified as belonging to the correct class. The other parts of the matrix above and below the diagonal show lighter shades of blue meaning lesser alarms were misclassified. A weighted version of the confusion matrix is shown in Figure 12(b) which sums the possibilistic weights assigned to each target type. The weighted confusion matrix has a smaller possibilistic weight assigned to the false alarms classified as target type one. This shows that even though many false alarms were classified to be targets of type 1, they had lower weights. The diagonal elements have much higher weights showing more confidence in the classification.

5.2.2. Cross Validation on the Real Data Set

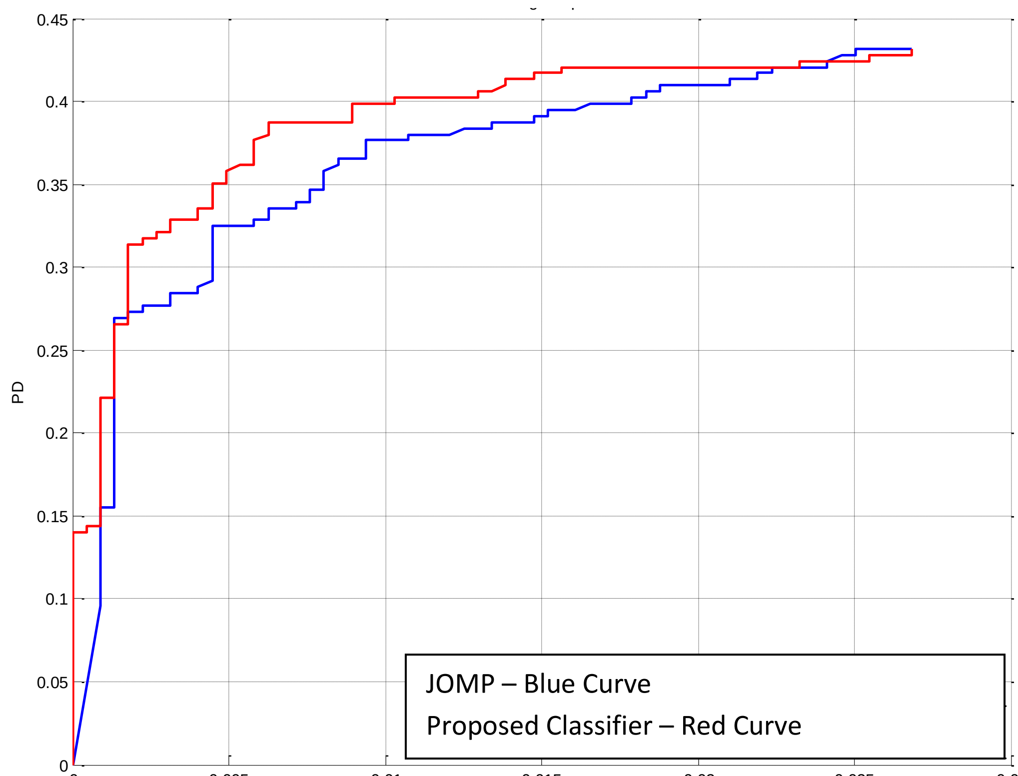

Figure 13: ROC for 10-fold cross-validation

Cross Validation is done in order to show the robustness of the algorithm. The training data is divided in 10 folds, with each fold having similar target type distribution. The ROC for 10 fold cross-validation is shown in Figure 13. The red curve shows the classifier performance on the JOMP alarms (blue curve). The classifier out performs the JOMP agorithm and stays over the blue curve at almost all the places, which indicates the bump in the classification. Since this classification is done on the JOMP alarms, the scope for the bump is limited by the alarms that JOMP finds. Even then the classifier is able to give significant boost to the ROC.

5.2.3. Comparison of Results between Possibilistic K-NN Classifier with DSRF Dictionary and the K-Nearest Neighbor Classifier

The possibilistic classifier is also compared to the standard *K*-nearest neighbor classifier. In order to make the comparison fair, the *K* most frequently chosen dictionary elements are used to determine the confidence value. The *K* used in the proposed classifier is 2 and hence the same *K* value was used with the *K*-nearest neighbor classifier. The residual error is calculated for each pixel in the blob with respect to each of these *K* dictionary indices. The vector of residual errors multiplied by the normalized counts at each dictionary index is chosen as the feature vector. The distance between the feature vectors of the alarm to all other alarms is measured. The *K* closest distances are taken into consideration. The final class is assigned by majority voting between the *K* neighbors. If all the neighbors have different class labels, the label of the closest class is assigned to the alarm. The class assignment for an alarm a_i can be summarized by the following equation,

$$C(a_i) = \begin{cases} \underset{k\epsilon K}{mode}[\, C(a_k)], & if\ all\ neighbors\ do\ not\ have\ unique\ labels \\ C(a_{\underset{k\epsilon K}{min}\, D_{ik}}), & otherwise \end{cases} \qquad (19)$$

where D_{ik} is the distance between the alarm a_i and a_k. K is the set of the nearest neighbors and $C(a_i)$ is the class label assigned to the alarm a_i

In order to make a ROC curve, confidence is assigned to the test alarm, proportional to the mean JOMP confidence of the *K* nearest alarms and inversely proportional to the mean distances of the *K* neighbors as shown in equation below,

$$Conf(a_i) = \frac{\underset{k \epsilon K}{mean}\, J(a_k)}{\underset{k \epsilon K}{mean}\, D_{ik}} \tag{20}$$

where $J(a_k)$ is the confidence assigned to the alarm a_k during the JOMP prescreening stage.

In order to make the comparison fair, the results are scored in the same manner as for the proposed classifier. A 10- fold alarm based cross-validation is done, similar to that used to score the results for the classifier presented in this paper. The results are as shown in Figure 14, with the blue curve representing JOMP results and red curve showing the *K*-nearest neighbor algorithm results. These results show that that *K*-nearest neighbor algorithm by itself does worse than the existing JOMP prescreener.

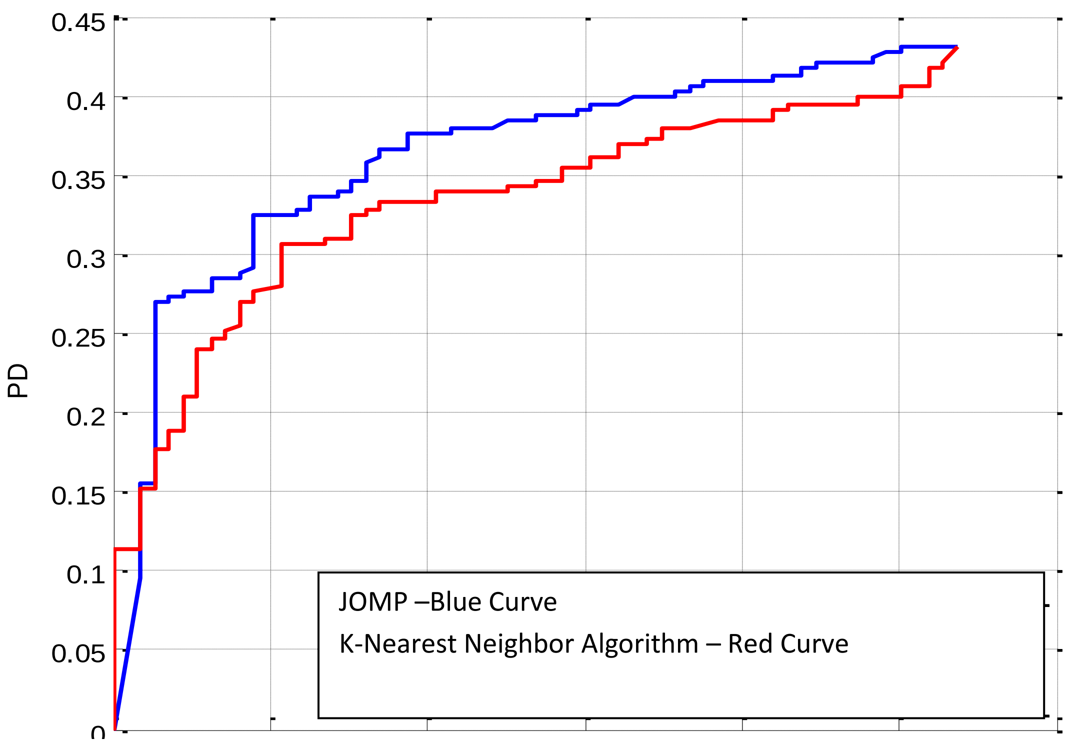

Figure 14: ROC comparisons between JOMP Prescreener and K-Nearest Neighbor classifier

5.3. Experiments and Results on using the classifier with the Clustered Dictionary

Figure 15 shows the confidence maps generated using both the dictionaries on a single alarm. As can be seen, the confidence map with the clustered dictionary has higher values even around the alarm blob whereas the DSRF dictionary gives higher confidence to distinct points contained in the alarm. The confidence values are inversely proportional to the residuals produced while performing JOMP.

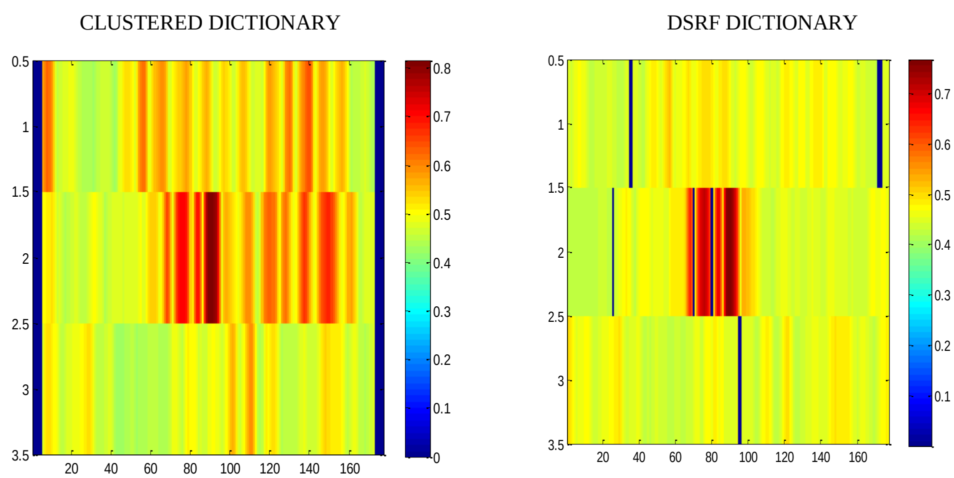

Figure 15: JOMP confidence map comparison between DSRF and clustered dictionary on a single alarm

With the K-means clustering, it was also difficult to obtain a single soil model to represent the background points. Since some of the training alarms also contained false alarms, using K-means clustering on the false alarms, gave inappropriate elements.

To get an idea of the number of clusters, the K-means algorithm was run with different values of *K*. On looking at the dictionary obtained, the right number of clusters seems to be between 30 and 60. As the number of clusters was increased, there were more prototypes that looked similar. The figure below shows the dictionary obtained each time the number of cluster was changed. Since increasing the number of clusters, essentially only increases, redundant elements, and much effect is not seen with regards to the dictionary matching and residues obtained and, hence, the classifier results improve to some extent on increasing the number of dictionary elements.

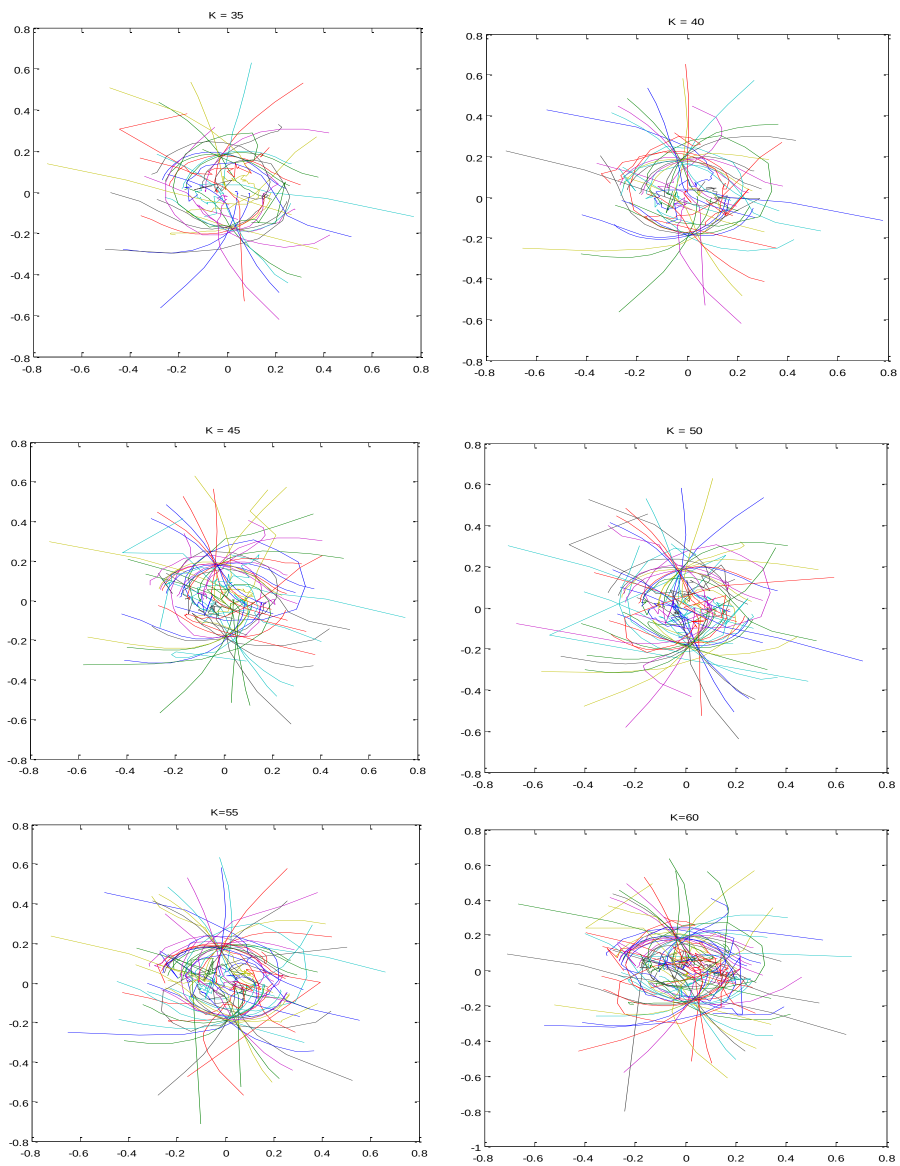

Figure 16: Clustered dictionary obtained by varying the number of clusters

As can be seen from Figure 16, many dictionary elements look exactly the same, which means that the cluster prototypes are very close to each other. The *K*-means algorithm was used with a random data point being chosen as initial cluster centers each time. In order to measure the sensitivity of the classifier to the number of clusters, the classifier was run using dictionary with number of dictionary elements ranging from 30, 35,40 … 60. The results obtained are as follows.

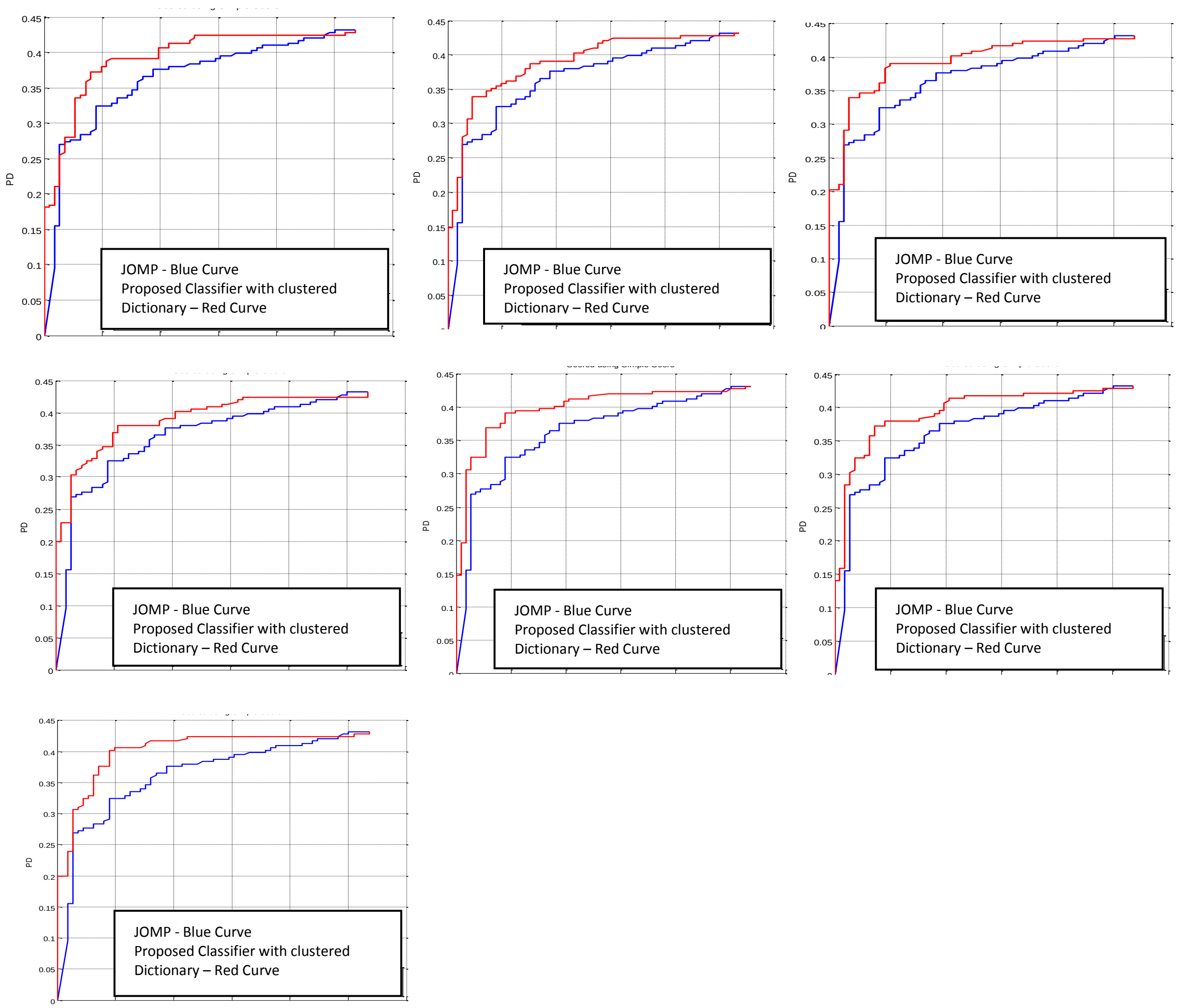

Figure 17: ROC comparisons between JOMP and proposed classifier using clustered dictionary (K=30, 35, 40 …60), All ROC curves have the same x-axis scale for comparison.

These results show that 30 dictionary elements are too less to represent all the alarms and the ROC does not show a significant improvement over JOMP. When fewer clusters were used (K = 30) to perform clustering, the classifier performance is almost same as JOMP prescreener. Finally the number of clusters was fixed to 60, since this is also the number of dictionary elements chosen by alarms when using DSRF dictionary.

The two main concerns with using K-means clustering to finding the dictionary elements is that it is difficult to estimate the number of clusters and clustering the data may lead to centers that are very close (similar in shape) to each other. Also varying the number of clusters did not give exact same results each time, but these were statistically the same given a 95% confidence interval.

5.4. Experiments and Results using Model-based Dictionary Estimation

The model-based dictionary estimation described above requires the number of dictionary elements needed in advance. The number of dictionary elements was kept to ten relating to one dictionary element per target type. In order to tailor the dictionary to the training data, the data was divided by class type. These dictionary elements were then used with the proposed classifier to find the frequency of the element picked by each target. The classifier results when compared to JOMP prescreener are as shown,

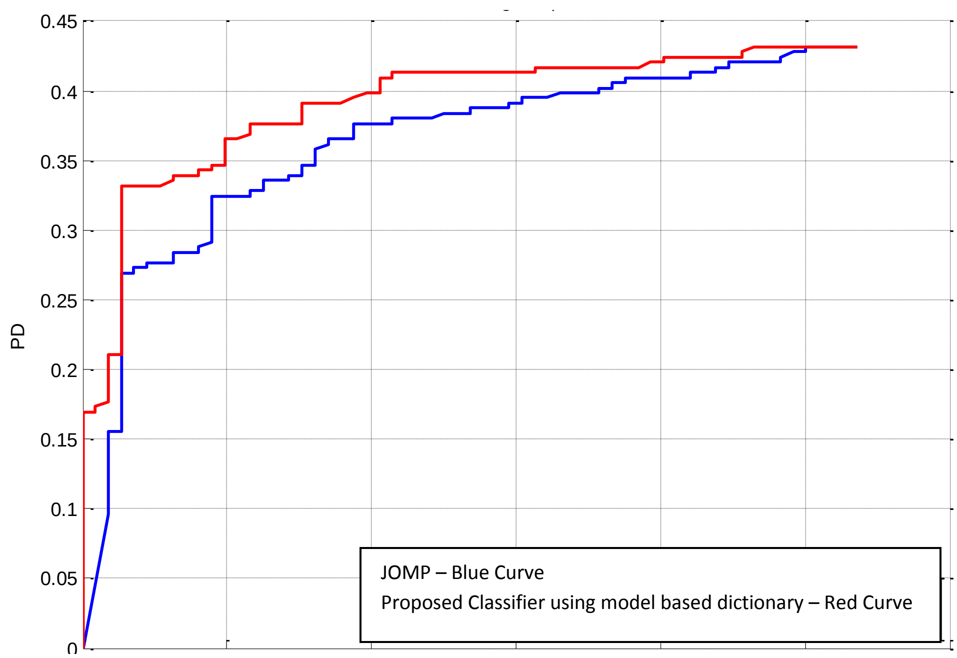

Figure 18: ROC showing the comparison of JOMP prescreener to the proposed classifier using dictionary learnt using model based method.

The classifier results are improved when compared to the JOMP prescreener; however this dictionary does not out-perform the parametric DSRF dictionary. However, the results being similar, this approach can provide the added advantage in terms of computational speed since now for each test point the MP computations can be reduced. In order to learn the dictionary, the training data was divided by type. When the data was divided according to type and depth, the dictionary obtained gave slightly better results. This means that the depth information contained in the buried object can be used as an important feature for discrimination. This also seems reasonable since the response collected by the metal detector would be much stronger for targets buried at a lesser depth as opposed to targets buried deep beneath the earth surface. These results are as shown below,

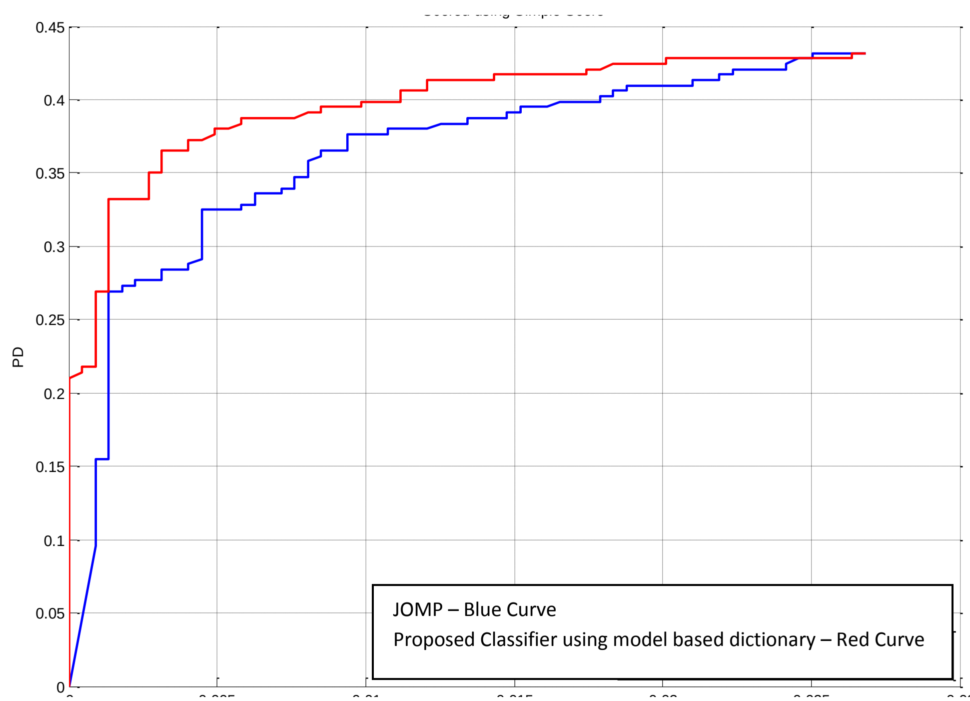

Figure 19: Classifier performance with the model based dictionary when targets were divided by type and depth

Some other experiments using this dictionary included looking at ROC of individual target types to see if certain target types were benefitted on learning the dictionary from training data as opposed to using a parametric dictionary. The ROC below compared the JOMP and proposed classifier performance on anti-tank type mines. This ROC indicates that learning the dictionary shows an improvement in ROC at an early false alarm rate.

The proposed method was also tested on the anti-tank and anti-personal type landmines individually. This was done by performing the training and cross validation only on the anti-tank type or the anti-personal mine type and ignoring the rest.

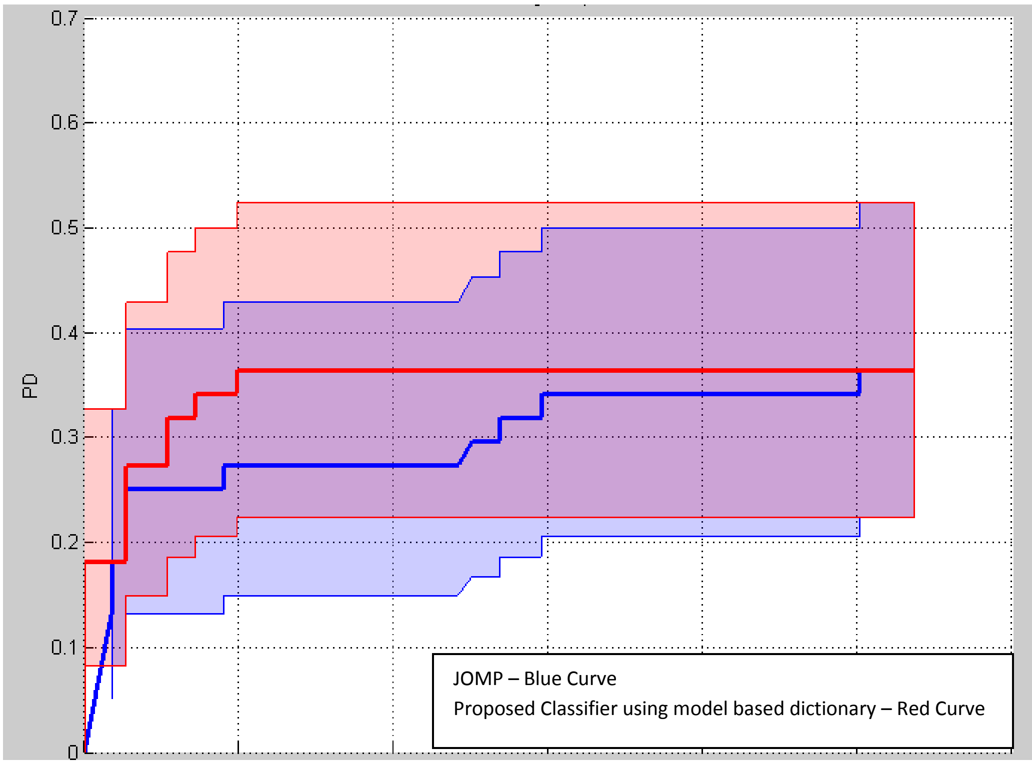

Figure 20: ROC comparing JOMP with proposed classifier using model based dictionary for anti-tank type mines

In case of anti- personal type mines, as shown in Figure 21, dictionary estimated from the data does not make any improvement in the ROC. Hence for those mine types it may be better to use parametric form of dictionary.

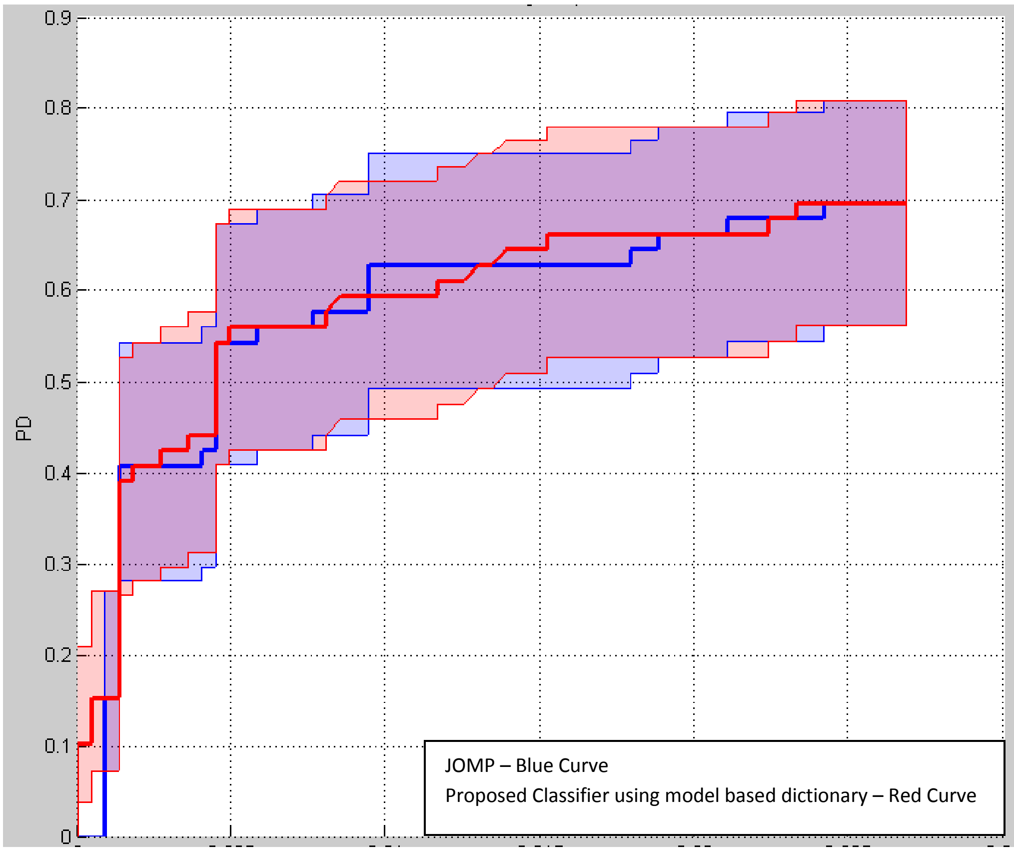

Figure 21: ROC comparison between JOMP and proposed classifier using model based dictionary on anti-personal mine type

5.5. Summary and Future work

The classifier presented in this paper is able to discriminate between the different mine types in WEMI data using a possibilistic K-NN classifier. The proposed method is shown to outperform the prescreener JOMP algorithm as well as the SVM and the standard *K*-Nearest Neighbor Algorithm for landmine classification.

Some of the drawbacks of the proposed classifier are that the confidence assignment depends on the pre-screener confidence. This will essentially add to the error of detection, if there was an error at the prescreening stage. Hence it might be useful to remove this dependence by using OWA operators or some other method to come up with a final confidence value. Since the buried targets can be of various types some of which have overlapping responses, it might be advantageous to use more features besides the counts to get better classification.

As discussed earlier the choice of the dictionary plays a very important role in detection. A dictionary combining model based estimation with clustering might be able to represent the data better. Some dictionary pruning methods can also be applied to remove redundant elements and only keep the informative elements. Incorporating this in the prescreener stage itself might provide a better classification and also find more targets in the original JOMP pre-screener that are currently missed (target types 5,11,12 etc).

6. CONCLUSION

The experiments and results conducted on the WEMI data suggest that the proposed classifier offers promising results by not only improving the ROC over the JOMP prescreener but also leveraging the discriminative power of JOMP to distinguish different mine types. These results are better than some of the state-of-art algorithms such as SVM and *K*- nearest neighbor's algorithm.

Experiments on learning the dictionary also gave interesting results. The *K* means clustering of the WEMI data showed that it is difficult to come up with a good background model by clustering, that can be used as a prototype for background points. The model based dictionary when used with the proposed classifier gave better results than the JOMP prescreener. When compared with the DSRF dictionary the model based dictionary performs better at lower false alarm rates. These results show that investing further on model based dictionary learning algorithms might provide better results on WEMI data.

The above results indicate that while a parametric dictionary can have storage advantages, learning the dictionary from the data can have its own benefits. The choice of using either is highly dependent on the data. In some cases, it may also be worthy to use a combination of parametric and tailored dictionary to get better results. Using the model-based form approach to obtain the dictionary shows improvement in the classifier results as opposed to using the dictionary clustered using K-means.

7. REFERENCES

[1] W. Scott, “Broadband array of Electromagnetic Induction Sensors for Detecting Buried Landmines” *in Proc, IGARSS, pages: 375-378, Jul. 2008*

[2] J. Igel, K. Takahashi, and H. Preetz “Electromagnetic soil properties and performance of GPR for landmine detection: How to measure, how to analyze and how to classify”, *Advanced Ground Penetrating Radar (IWAGPR), pages: 1-6, June 2011*

[3] P. Gao, L. Collins, P. Garber, N. Geng, and L. Carin, “Classification of Landmine-Like Metal Targets Using Wideband Electromagnetic Induction,” *IEEE Transactions on Geoscience and Remote Sensing, Vol. 38, No. 3, May 2000.*

[4] L. Collins, P. Gao, and L. Carin, “An Improved Bayesian Decision Theoretic Approach for Land Mine Detection,” *IEEE Transactions on Geoscience and Remote Sensing, Vol. 37, No. 2, March 1999.*

[5] G. D. Sower and S. P. Cave, “Detection and identification of mines from natural magnetic and electromagnetic resonances,” *in Proc. SPIE, 1995.*

[6] C. Baum, “Low Frequency Near-Field Magnetic Scattering from Highly, but Not Perfectly Conducting Bodies,” *Phillips Laboratory, Interaction Note 499, Nov. 1993*

[7] I. Won, D. Keiswetter and D. Hansen, “GEM-3: A Monostatic Broadband Electromagnetic Induction Sensor” *Journal of Environmental and Engineering Geophysics, pages: 53-64, Aug. 1997*

[8] L. Collins, P. Gao, L. Makowsky, J. Moulton, D. Reidy and D. Weaver, “Improving detection of low-metallic content landmines using EMI data” *IEEE transactions, Geoscience and Remote Sensing Symposium, 2000. IGARSS, vol.4, pages 1631 - 1633*

[9] P. Gao, L. Collins, N. Geng and L. Carin “Classification of buried metal objects using wideband frequency-domain electromagnetic induction responses: a comparison of optimal and sub-optimal processors” *IEEE Transactions, Geoscience and Remote Sensing Symposium, IGARSS Proceedings. International, vol. 3, pages: 1819 – 1822*

[10] J. Miller, T. Bell, J. Soukup, and D. Keiswetter, “Simple phenomenological models for wideband frequency-domain electromagnetic induction,” *IEEE Tran. on Geoscience and Remote Sensing, vol. 39, no.6, pages: 1294–1298, 2001.*

[11] H. Huang and I. Won, “Automated anomaly picking from broadband electromagnetic data in an unexploded ordinance (uxo) survey,” *Geophysics, vol. 68, no. 6, pages: 1870–1976, 2003.*

[12] T. Bell, B. Barrow and J. Miller, “Subsurface Discrimination Using Electromagnetic Induction Sensors” *IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 6, June 2001*

[13] H. Frigui, L. Zhan and P. Gader,” Context-Dependent Multi-Sensor Fusion For Landmine Detection” *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. vol. 2, pages: 371-374*

[14] S. Yuksel, G. Ramachandran, P. Gader, J. Wilson, K. Ho and G. He,” Hierarchical Methods For Landmine Detection With Wideband Electro-Magnetic Induction And Ground Penetrating Radar Multi-sensor Systems” *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International, vol. 2, pages 177-18, July 2008*

[15] S. G. Mallat and Z. Zhang, “Matching Pursuits With Time-Frequency Dictionaries,” *IEEE Trans. Signal Processing, vol. 41, no. 12, Dec.1993*

[16] J. Friedman and W. Stuetzle, “ Projection Pursuit Regression”, *Journal of the American Statistical Association, vol 76, Issue 376, 1981*

[17] X. Zhang, L. Durand, L. Senhadji, H. Lee, J. Coatrieux, “Analysis-synthesis of the phonocardiogram based on the matching pursuit method” *Biomedical Engineering, IEEE Transactions, vol. 45, Issue: 8, 1998*

[18] I. Kyriakides, A. Papandreou and D. Morrell, “Adapting Matching Pursuit Dictionaries to Waveform Structure Using Particle Filtering” *Sensor Array and Multichannel Processing, 2006. Fourth IEEE Workshop, pages: 561-565, July 2006,*

[19] D. Stefanoiu and F. lonescu, “A genetic matching pursuit algorithm," *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on, vol. 1, pages: 577-580, 1-4 July 2003.*

[20] D. Whitley, “A genetic algorithm tutorial," *Statistics and Computing, vol. 4, pages: 65-85, 1994*

[21] S.F. Cotter and B. Rao, “Application of tree-based searches to matching pursuit," *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on, vol. 6, pages: 3933- 3936.*

[22] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *Signals, Systems and Computers, vol. 1, pages: 40-44, Nov 1993.*

[23] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review, vol. 43, no. 1, pages: 129-159, 2001.*

[24] M. Yaghoobi, L. Daudet and M. Davies, "Parametric Dictionary Design for Sparse Coding" *IEEE Transaction on Signal Processing, Vol.57 , pages: 4800-4810, Dec 2009*

[25] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, L. Carin, "Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images" *IEEE Transactions on Image Processing, vol 21, issue 1, pages:130-144, Jan 2012*

[26] L. Fang and S. Li "An Efficient Dictionary Learning Algorithm for Sparse Representation" *Pattern Recognition (CCPR), Chinese Conference pages:1-5, Oct 2012*

[27] Y. Zhang, X. Liao, and L. Carin, "Detection of buried targets via active selection of labeled data: application to sensing subsurface uxo," *Geoscience and Remote Sensing, IEEE Transactions on, vol. 42, no. 11, pages: 2535-2543, Nov. 2004.*

[28] R. Mazhar, P. Gader, and J. Wilson, "A Matching Pursuit Based Similarity Measure for Fuzzy Clustering and Classification of Signals" *IEEE Trans. Fuzzy Syst., pages: 1950–1955, 2008.*

[29] R. Mazhar, J. Wilson, P. Gader, "Use of an Application Specific Dictionary for Matching Pursuits Discrimination of Landmines and Clutter" *presented at IEEE International Geoscience and Remote Sensing Symposium, 2007.*

[30] T. Glenn, J. Wilson and K. Ho "A multimodal Matching Pursuits Dissimilarity Measure applied to landmine/clutter discrimination", *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*

[31] G. Ramachandran, P. Gader, and J. Wilson, "GRANMA: Gradient Angle Model Algorithm on Wideband EMI Data for Land-Mine Detection," *IEEE Geosci. and Remote Sens., Vol. 7, No. 3, July 2010*

[32] 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 Proceedings of the SPIE: 2007 Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Vol. 6553, May 2007*

[33] J. Xiang, N. B. Jones, D. Cheng, and F. S. Schlindwein, “Direct inversion of the apparent complex-resistivity spectrum,” *Geophysics, vol. 66, no. 5, pages: 1399–1404, Sep./Oct. 2001.*

[34] M. Tran, C. Abeynayake and L. Jain, "A Target Discrimination Methodology Utilizing Wavelet-Based and Morphological Feature Extraction with Metal Detector Array Data", Geoscience *and Remote Sensing, IEEE Transactions on, vol 50, issue 1, pages: 119 - 129, Jan. 2012*

[35] M. Tran, C. Abeynayake, L. Jain and C. Lim, “An automated decision system for landmine detection and classification using metal detector signals,” in *Innovations in Defence Support Systems—1, vol. 304, 2010, pages: 175–200.*

[36] M. Tran, C. Lim, C. Abeynayake and L. Jain, “Feature extraction and classification of metal detector signals using the wavelet transform and the fuzzy ARTMAP neural network,” *Intell. Fuzzy Syst., vol. 21, no. 1, pages: 89–99, Apr. 2010.*

[37] G. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds and D. Rosen, “Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps,” *IEEE Trans. Neural Network., vol. 3, no. 5, pages: 698–713, Sep. 1992.*

[38] S. A. Dudani, “The distance weighted k-nearest neighbor rule” *IEEE Trans, Syst., Man, Cybern., vol. SMC-6, no. 4, pages: 325-327, Apr. 1976*

[39] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy k-nearest neighbor algorithm” *IEEE Trans. Syst., Man, Cybern., vol. SMC-15, no. 4, pages: 580–585, Jul./Aug. 1985*

[40] H. Frigui and P. Gader, “Detection and Discrimination of Land Mines in Ground-Penetrating Radar Based on Edge Histogram Descriptors and a Possibilistic K-Nearest Neighbor Classifier” *IEEE Trans. Fuzzy Systems, Vol. 17, No. 1, Feb. 2009*

[41] M. Wei, W. Scott and J. McClellan, “Robust Estimation of the Discrete Spectrum of Relaxations for Electromagnetic Induction Responses” *IEEE Trans. Geosci. Remote Sens, Vol. 48, No. 3, March 2010*

[42] S. Goldberg, T. Glenn, J. Wilson, P. Gader, “Landmine Detection using Two-tapped Joint Orthogonal Matching Pursuits,” *Proceedings of the SPIE, vol. 8357, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII, April 2012.*

[43] W. Hu, S. Tantum, and L. Collins, "EMI-based classification of multiple closely spaced subsurface objects via independent component analysis," *IEEE Trans. Geoscience Remote Sens., vol. 42, no. 11, pages 2544– 2554, Nov. 2004.*

[44] B. Nelson, D. Schofield, and L. Collins, "A comparison of neural networks and subspace detectors for the discrimination of low-metal content landmines," *in Proc. Detection Remediation Technol. Mines Minelike Targets VIII, 2003, pages: 1046–1053.*

[45] L. Lowe, L. Riggs, S. Cash and T. Bell, "Improvements to the U.S. Army's AN/PSS-12 metal detector to enhance landmine discrimination capabilities," *in Proc. Detection Remediation Technol. Mines Minelike Targets VI, 2001, pages 85–96.*

[46] D. Sundman, S. Chatterjee and M. Skoglund, "Greedy pursuits for compressed sensing of jointly sparse signals" in *Proc. EUSIPCO 2011.*

[47] M. McClure and L. Carin, "Matching pursuits with a wave-based dictionary," *Signal Processing, IEEE Transactions [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on], vol. 45, no. 12, pages: 2912-2927, Dec 1997.*

[48] P. Bharadwaj, P. Runkle, and L. Carin, "Target identification with wave-based matched pursuits and hidden markov models," *Antennas and Propagation, IEEE Transactions, vol. 47, no. 10, pages: 1543-1554, Oct 1999.*

[49] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Second Edition, Elsevier Academic Press, 2003.*

[50] M. Aharon, M. Elad and A. Bruckstein," K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation" *IEEE Transactions on Signal Processing, vol. 54, No. 11, pages: 4311-4322, Nov 2006*

[51] K. Engan, S. Aase, and J. Husoy, "Multi-frame compression: Theory and design," *EURASIP Signal Process., vol. 80, no. 10, pages: 2121–2140, 2000*

[52] K. Engan, S. Aase, and J. Hakon-Husoy, "Method of optimal directions for frame design," *in IEEE Int. Conf. Acoust., Speech, Signal Process., 1999, vol. 5, pages: 2443–2446.*

[53] S. Loyd, "Least squares quantization in PCM", *IEEE Trans. on Information Theory, vol 28, pages: 129-137, 1982.*

[54] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features" *IEEE*

Computer Vision and Pattern Recognition (CVPR), pages: 3501 – 3508, June 2010