2010

Applications of pattern classification to time-domain signals

Crystal Ann Bertoncini
College of William & Mary - Arts & Sciences

Recommended Citation
Bertoncini, Crystal Ann, "Applications of pattern classification to time-domain signals" (2010). *Dissertations, Theses, and Masters Projects.* Paper 1539623559.
https://dx.doi.org/doi:10.21220/s2-6b53-3790

Applications of Pattern Classification to Time-Domain Signals

Crystal Ann Bertoncini

Mechanicsburg, PA

Master of Science, The College of William and Mary, 2006
Bachelor of Arts, Vassar College, 2005

A Dissertation presented to the Graduate Faculty
of the College of William and Mary in Candidacy for the Degree of
Doctor of Philosophy

Department of Physics

The College of William and Mary
May, 2010

APPROVAL PAGE

This Dissertation is submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

Crystal Ann Bertoncini

Approved by the Committee, January 2010

Committee Chair
Professor Mark K. Hinders, Applied Science
The College of William and Mary

Professor Keith Griffioen, Physics
The College of William and Mary

VMEC Associate Professor Rosa A. Lukaszew, Applied Science and Physics
The College of William and Mary

Chancellor Professor Eugene Tracy, Physics
The College of William and Mary

Adjunct Professor Walter Silva, Applied Science
The College of William and Mary
NASA Langley

COMPLIANCE PAGE

Research approved by

The College of William and Mary Protection of Human Subjects Committee

Protocol number(s): PHSC-2007-03-28-4702-mkhind

Date(s) of approval: 16 April, 2007

ABSTRACT PAGE

Many different kinds of physics are used in sensors that produce time-domain signals, such as ultrasonics, acoustics, seismology, and electromagnetics. The waveforms generated by these sensors are used to measure events or detect flaws in applications ranging from industrial to medical and defense-related domains. Interpreting the signals is challenging because of the complicated physics of the interaction of the fields with the materials and structures under study. Often the method of interpreting the signal varies by the application, but automatic detection of events in signals is always useful in order to attain results quickly with less human error. One method of automatic interpretation of data is pattern classification, which is a statistical method that assigns predicted labels to raw data associated with known categories. In this work, we use pattern classification techniques to aid automatic detection of events in signals using features extracted by a particular application of the wavelet transform, the Dynamic Wavelet Fingerprint (DWFP), as well as features selected through physical interpretation of the individual applications. The wavelet feature extraction method is general for any time-domain signal, and the classification results can be improved by features drawn for the particular domain. The success of this technique is demonstrated through four applications: the development of an ultrasonographic periodontal probe, the identification of flaw type in Lamb wave tomographic scans of an aluminum pipe, prediction of roof falls in a limestone mine, and automatic identification of individual Radio Frequency Identification (RFID) tags regardless of its programmed code. The method has been shown to achieve high accuracy, sometimes as high as 98%.

Contents

List of Figures

List of Tables

This dissertation is dedicated to my parents, Daniel and Michele Bertoncini.

Acknowledgements

First, I would like to wholeheartedly thank my advisor, Dr. Mark Hinders, without whom this dissertation would never have happened.

I would also like to thank the Physics Department at the College of William and Mary, DentSply Int'l, the Virginia Space Grant Consortium, Naval Research Laboratory for funding. This work was performed [in part] using computational facilities at the College of William and Mary which were enabled by grants from Sun Microsystems, the National Science Foundation, and Virginia's Commonwealth Technology Research Fund.

Many people were involved in collecting and interpreting the data in this dissertation. For the periodontal project we must thank Gayle McCombs of ODU's Dental Hygiene Research Center as well as Jonathan Stevens. The periodontal simulations were written by Kevin Rudd, who very patiently helped explain how to operate and understand the simulations, and later assisted with the RFID data collection process. The data for the Lamb wave tomography classification was collected by Jill Bingham, who also helped me understand Helical Ultrasound Tomography and Lamb wave physics. The microseismic data was provided by National Institute of Occupational Safety and Hazards, where John Ellenberger was in particular an essential resource. I would also like to thank Martin Chapman of Virginia Polytechnic Institute for indispensable conversations on the subject of microseismic monitoring.

Technical assistance has been provided by Jonathan Stevens, who built the ultrasonographic periodontal probe prototype, and Chris Bording and Tom Crockett, who assisted with running data analysis on SciClone. Thank you for invaluable support.

Lastly, I would not be where I am today without my teachers. I cannot possibly list them all here but they were one of the most important influences of my life. To my friends - you made graduate school bearable. Most of all, I would like to thank my family, whose love and support nurtured me though my education.

Chapter 1

Introduction

Time-domain signals are produced by a wide variety of sensors. Ultrasonic sensors operate via piezoelectricity to transform mechanical energy to electricity and can record either bulk or surface waves with frequencies above 20 kHz. Geophones used in seismology also transform mechanical energy to electricity but use the movement of a magnet inside a coil to generate a current. Radio frequency electromagnetic sensors vary widely but often include a circuit to modulate an incoming or outgoing signal and an antenna for transmission or reception of the signal.

The most common method of analyzing the signals produced by these kinds of sensors involves windowing around the portion of the signal with the event or reflection, which usually has a higher amplitude than the surrounding noise, and measuring the amplitude and time that the event occurs in the signal. However, in many real-world materials and structures, which are often irregular or anisotropic, the amplitude of the event is small enough that it is difficult to resolve it from the surrounding noise merely by a simple threshold. This is often the result of a complicated interaction of the physical fields with the materials. Modeling these complex interactions carefully enough to reproduce the environment is often impossible so interpreting the signal cannot usually be performed by theory alone. Instead, sophisticated processing is required to resolve the event from the surrounding noise.

Artificial intelligence techniques can assist the signal analysis as well as provide other advantages, such as possibly increasing the speed of analysis and reducing human error. Artificial intelligence is a broad term but is used here to imply automatic decisions made by a computer to maximize success. Pattern classification is a subset of artificial intelligence that uses multivariate statistics to assign predicted labels to raw data. In supervised pattern classification, the raw data has been assigned known categories, but unsupervised classification techniques look for natural divisions of categories in the raw data. This dissertation applies pattern classification techniques to aid the automatic detection capability of sensors. The success of the technique is demonstrated through four applications using a variety of sensors and physical fields.

1.1 Pattern Classification

Pattern classification is a process of grouping raw data into categories. Also known as pattern recognition, it is a subset of machine learning and artificial intelligence, in which computers and machines are meant to learn tasks through experience. Pattern classification is often a necessary stage of scientific research in a variety of fields, including medicine and biology, biometrics, sensing, signal analysis, psychology, financial forecasting, and image analysis. There are many fine text books on the subject of pattern classification [1–8]. Of these, Duda and Hart [1] have written one of the most widely used and accessible textbooks on the subject, while an in-depth statistical analysis is found in [2] and [4]. Some authors handle pattern classification, in which an object is described to a class, parameter estimation, which attempts to find a model for an event, and state estimation, which is similar to pattern classification but ordered in time, in the same text [6]. Our discussion will focus on pattern classification. The pattern classification process often contains the following steps [9]:

1. Sensing
2. Segmentation (optional)
3. Pre-processing/dimensionality reduction (as needed)
4. Feature Extraction
5. Feature Selection (optional)
6. Classification
   (a) Training
   (b) Testing
7. Classifier Combination (optional)
8. Evaluation

The first step, sensing, gathers the raw data that is to be classified, which sometimes needs to be segmented afterwards, as in the case of speech or character recognition. In simple cases, the raw data is scalar, but higher-dimensional data is more common, as in image processing or signal analysis. Therefore, dimensionality reduction or preprocessing is meant to reduce the dimensionality to a manageable size either before or after feature extraction. The preprocessing stage also includes other techniques such as filtering or ordering of the raw data.

For each object to be classified there may be at most a 1-dimensional array of features, known as a feature vector, to be used in classification. This explains why dimensionality reduction may be required before or after feature extraction. Feature extraction is arguably the most critical step of the pattern classification process. If the features are poorly chosen for classification, then raising the mathematical complexity by choosing more complicated classifiers will not lead to successfully finding patterns in the data. The selection of features for classification varies by application, and feature extraction often requires transforms into other mathematical spaces, such as wavelet or Fourier transforms, principal component analysis, and discriminant analysis. In image analysis, for example, the procedures to extract features are a necessary precursor to template matching [10].

In selecting the feature vector, one must be careful to avoid the Curse of Dimensionality, in which the number of objects to be classified is small relative to the size of the feature vectors. Depending on the classifier, as a rule of thumb, there ought to be ten times as many samples in each class as there are features in the feature vector [11]. However, up until that point, increasing the number of features often increases the performance of a classifier. Furthermore, the Ugly Duckling Theorem states that "in the absence of assumptions there is no privileged or 'best' feature representation" [1] since the choice of features rests on making assumptions about what 'best' means. Therefore, there is no *a priori* best feature vector for a particular application. Lastly, there exist several known procedures for selecting the best features that have been extracted [12, 13]. These methods often involve first testing the success of the classifier on these features. Feature selection often takes place offline, and may only need to be optimized once.

There are many different types of statistical pattern classification schemes. Bayes decision theory can be employed if the samples are drawn from a known distribution function or if the distribution can be modeled from the training data. Pattern classification can be supervised, so that the object, its feature vector, and its class are all known, but unsupervised pattern classification, in which the class is unspecified, also occurs [14]. Both supervised and unsupervised learning can be broken down into further categories. Parametric classifiers assume that the data belongs to a particular probability distribution function. Nonparametric classifiers measure the distribution from the data, as in the well-known k-nearest-neighbor classifier. The special case of invariant patterns classifiers, which are invariant to rotation and translation, is yet more selective [15].

In the best practice of pattern classification, the classifier is first trained and optimized then tested on a different set of samples as those used to train the classifier. However, for practical reasons, the training and testing set of data may not always be separate, due to the difficulty or cost of obtaining more data. In these cases, hold-out or leave-one-out techniques can be employed, as well as bootstrapping and jackknifing, in which some subset of the data is withheld for testing, possibly with resubstitution or repetition. The mathematical rigor required to properly explain these classifiers can be found in [1–9]. However, as far as choice of classifier is concerned, we must refer to the No Free Lunch Theorem, which states that "there are no context-independent or usage-independent reasons to favor one learning or classification method over another" [1]. Even if one classifier outperforms another in a particular application, we cannot therefore conclude that the successful classifier is *a priori* better than any other for different applications.

Recently, the advantages of training and combining multiple classifiers have been explored [16–21]. Combining several classifiers can often reduce error rates [16] but also increases the complexity of the problem because combining classifiers requires fusing decision-making schemes [17]. Samples can be labeled either before or after combination of classifiers occur, which further increases the complexity [18]. Combining classifiers achieves the most success over single classification schemes whenever the classifiers are both accurate and diverse [19]. References [20, 21] review available methods for combining classifiers. It should be noted that when only labels are available for combining classifier results, majority vote is often the method of combination [17]. Other methods like weighted mean voting require more information, such as the probability distribution or expected error rate. One should be careful to note that combining classifiers cannot substitute for proper individual classifications [21] in the same way that complicated mathematics cannot improve poor feature selection. However, gains from combining classifiers can be small. As an example, applying classifier combination techniques to fingerprint recognition showed that combining two major fingerprint recognition paradigms, texture-based and minutiae-based methods, or from multiple fingers resulted in small performance increases of around $1 - 5\%$ [22, 23].

1.1.1 Wavelets for Pattern Classification

Previous researchers have made use of the wavelet transform for pattern classification applications [24, 25]. One option is to integrate wavelets directly by capitalizing on the orthogonal property of wavelets to estimate the class density functions [26]. However, most applications of wavelets to pattern recognition focus on feature extraction techniques [27]. One common method involves finding the wavelet transform of a continuous variable (sometimes a signal) and computing the spectral density, or energy, which is the square of the coefficients [28, 29]. Peaks of the spectral density or the sum of the density can be used as features and have been applied to flank wear estimation in turning processes and classifying diseased lung sounds [29] as well as to evaluating simulated chirp signals and the equine gait [28]. This technique is similar to finding the cross-correlation but is only one application of wavelets to signal analysis. One alternate method is to deconstruct the signal into an orthogonal basis, such as Laguerre polynomials [30]. Another technique is the adaptive wavelet method [31–34] which stems from multiresolution analysis [35]. Multiresolution analysis applies a wavelet transform using an orthogonal basis resulting in filter coefficients in a pyramidal computation scheme, while adaptive wavelet analysis uses a generalized M-band wavelet transform to similarly achieve decomposition into coefficients and inserts those coefficients into matrix form for optimization. Adaptive wavelets result in efficient compression and have the advantage of being widely applicable.

Wavelet methods are also often applied for feature extraction in images, such as for shape characterization and to find boundaries [36, 37]. Wavelets are particularly useful for detecting singularities, and in 2D data spaces this results in an ability to identify corners and boundaries. Shape characterization itself is a precursor to template matching in pattern classification, in which outlines of objects are extracted from an image and matched to known shapes from a library. Other techniques are similar to those described above, including multiresolution analysis, which is also similar to the image processing technique of matched filters [38–42]. Either libraries of known wavelets or wavelets constructed from the original signal are used to match the signal of interest [38]. Pattern recognition then proceeds in a variety of ways from the deconstructed wavelet coefficients. The coefficients with minimal cost can be used as features [39] or the results from each sequential step can be correlated individually [40].

To reduce dimensionality, sometimes projection transforms are precursors to decomposition [41, 42]. Some authors have constructed a rotationally-invariant projection that deconstructs an image into sub-images and transforms the mathematical space from 2D to 1D [26, 43, 44]. Also, constructing a set of orthonormal bases, just as in the case of adaptive wavelets above, remains useful for images [45]. Because of the dimensionality, computing the square of the energy is cumbersome and so a library of codebook vectors is necessary for classification [46]. There have been many successful applications of pattern recognition in ultrasound, some of which include wavelet feature extraction methods. In an industrial field, Tansel *et al.* [47, 48] selected coefficients from wavelet decomposition for feature vectors and were able to use pattern recognition techniques to detect tool failure in drills. Learned and Wilsky [49] constructed a wavelet packet approach, in which energy values are calculated from a full wavelet decomposition of a signal, to detect sonar echoes for submarines. Wu and Du [50] also used a wavelet packet description but found that feature selection required knowledge of the physical space, in this case, drill failure. Case and Waag [51] used Fourier coefficients instead of wavelet coefficients for features that successfully identified flaws in pipes. Comparing several techniques, including features selected from wavelet, time, and spectral domains, Drai *et al.* [52] identified welding defects using a neural network classifier. Buonsanti *et al.* [53] compared ultrasonic pulse-echo and eddy current techniques to detect flaws in plates using a fuzzy logic classifier and wavelet features relevant to the physical domain.

In the medical field, an early example of applying classification techniques is evidenced in Momenan *et al.*'s work [54] in which features selected offline are used to identify changes in tissues as well as clustering in medical ultrasound images. Bankman *et al.* [55] classified neural waveforms successfully with the careful application of preprocessing techniques such as whitening via autocorrelation. Meanwhile, Kalayci *et al.* [56] detected EEG spikes by selecting eight wavelet coefficients from two different Debauchies wavelet transforms for application in a neural network. Tate *et al.* [57] similarly extracted wavelet coefficients as well as other prior information to attempt to identify vegans, vegetarians, and meat-eaters by their Magnetic Resonance spectra. Interestingly, vegetarians were more difficult to identify, with a classification accuracy of 80%. Mojsilovic *et al.* [58] applied wavelet multiresolution analysis to identify infarcted myocardial tissue from medical ultrasound images. Georgiou *et* *al.* [59, 60] used wavelet decomposition to calculate scale-averaged wavelet power up to a threshold and detected the presence of breast cancer in ultrasound waveforms by means of hypothesis testing. Also using multiresolution techniques, Lee *et al.* [61] further selected fractal features to detect liver disease. Alacam *et al.* [62] improved existing breast cancer characterization of ultrasonic B-mode images by adding fractional differencing and moving average polynomial coefficients as features.

The research discussed above demonstrates the utility of applying pattern classification to aid detection in the industrial and medical domains. In our research, we apply similar pattern classification techniques to four different applications.

1.2 Applications

In this dissertation, we apply pattern classification to features extracted from time-domain signals using wavelet transforms and image recognition techniques. In Chapter 2, we describe pattern classification formally, introduce wavelets, and describe the mechanism to extract features for pattern classification using the Dynamic Wavelet Fingerprint (DWFP). We demonstrate these techniques through an example to detect flaws in the hard tissue of teeth. In Chapter 3, we describe the development of an ultrasonographic probe to detect periodontal pocket depth. The binary classification algorithm, which was designed to perform pattern classification for an unbalanced class distribution, is also described. A method to characterize flaw type in Lamb wave tomographic scans of an aluminum pipe is described in Chapter 4, where the classifier was designed to identify hole flaws and gouge flaws as distinct from unflawed sections of the pipe. An unsupervised pattern classification application, predicting roof falls in mines using microseismic waveforms, is described in Chapter 5. Finally, Chapter 6 describes an application that distinguishes unique individual Radio Frequency Identification (RFID) tags from members of the same type in a procedure called Specific Emitter Identification (SEI). The intended application is for security applications such as verification of ID badges and passports. Conclusions and suggestions for future work are presented in Chapter 7.

Bibliography

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2$^{\text{nd}}$ ed. New York: Wiley, 2001.

[2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Boston: Academic Press, 1990.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning.* New York: Springer, 2006.

[4] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition.* New York: Springer, 1996.

[5] S. Theodoridis and K. Koutroumbas, *Pattern Recognition.* New York: Academic Press, 1999.

[6] F. van der Heijden, R. P. W. Duin, D. de Ridder, and D. M. J. Tax, *Classification, Pattern Estimation, and State Estimation: An Engineering Approach Using MATLAB.* Hoboken, NJ: Wiley, 2004.

[7] A. R. Webb, *Statistical Pattern Recognition*, 2$^{\text{nd}}$ ed. Hoboken, NJ: Wiley, 2002.

[8] B. D. Ripley, *Pattern Recognition and Neural Networks.* New York: Cambridge University Press, 1996.

[9] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[10] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[11] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, P. R. Krishnaiah and L. N. Kanal, Eds. Amsterdam: North-Holland, 1982, vol. 2, pp. 835–855.

[12] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245 – 271, 1997.

[13] J. Iannarilli, F.J. and P. Rubin, "Feature selection for multiclass discrimination via mixed-integer linear programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 779–783, June 2003.

[14] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[15] J. Wood, "Invariant pattern recognition: A review," *Pattern Recognition*, vol. 29, no. 1, pp. 1 – 17, 1996.

[16] K. Tumer and J. Ghosh, "Error correlation and error reduction in ensemble classifiers." *Connection Science*, vol. 8, no. 3-4, pp. 385–404, 1996.

[17] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[18] L. Kuncheva and J. Bezdek, “Presupervised and post-supervised prototype classifier design,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1142–1152, Sep 1999.

[19] T. G. Dietterich, “Ensemble methods in machine learning,” in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2000, vol. 1857, pp. 1–15.

[20] G. Valentini and F. Masulli, “Ensembles of learning machines,” in *Neural Nets*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2002, vol. 2486, pp. 3–20.

[21] L. I. Kuncheva, *Combining pattern classifiers methods.* New York: Wiley, 2004.

[22] S. Prabhakar and A. K. Jain, “Decision-level fusion in fingerprint verification,” *Pattern Recognition*, vol. 35, no. 4, pp. 861 – 874, 2002.

[23] G. L. Marcialis and F. Roli, “Fusion of multiple fingerprint matchers by single-layer perceptron with class-separation loss function,” *Pattern Recognition Letters*, vol. 26, no. 12, pp. 1830 – 1839, 2005, artificial Neural Networks in Pattern Recognition.

[24] A. T. Nelson, “Wavelet wavelet transform and pattern recognition method for heart sound analysis,” US Patent 20070191725, 2007.

[25] A. Aussem, J. Campbell, and F. Murtagh, "Wavelet-based feature extraction and decomposition strategies for financial forecasting," *Journal of Computational Intelligence in Finance*, vol. 6, no. 2, pp. 5–12, 1998.

[26] Y. Y. Tang, L. H. Yang, J. Liu, and H. Ma, *Wavelet Theory and Its Application to Pattern Recognition.* River Edge, NJ: World Scientific, 2000.

[27] R. R. Brooks, L. Grewe, and S. S. Lyengar, "Recognition in the wavelet domain: a survey," *Journal of Electronic Imaging*, vol. 10, no. 3, pp. 757–784, 2001.

[28] G. P. Nason and B. W. Silverman, "The stationary wavelet transform and some statistical applications," in *Wavelets and Statistics*, ser. Lecture Notes in Statistics, A. Antoniadis and G. Oppenheim, Eds. Germany: Springer-Verlag, 1995, pp. 281–299.

[29] S. Pittner and S. V. Kamarthi, "Feature extraction from wavelet coefficients for pattern recognition tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 1, pp. 83–88, 1999.

[30] A. M. Sabatini, "A digital-signal-processing technique for ultrasonic signal modeling and classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 1, pp. 15–21.

[31] R. Coifman and M. Wickerhauser, "Entropy based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, pp. 713–718, 1992.

[32] H. H. Szu, B. Telfer, and S. Kadambe, "Neural network adaptive wavelets for signal representation and classification," *Optical Engineering*, vol. 31, pp. 1907–1916, 1992.

[33] B. A. Telfer, H. H. Szu, G. J. Dobeck, J. P. Garcia, H. Ko, A. Dubey, and N. Witherspoon, "Adaptive wavelet classification of acoustic and backscatter and imagery," *Optical Engineering*, vol. 33, pp. 2192–2203, 1994.

[34] Y. Mallet, D. Coomans, J. Kautsky, and O. D. Vel, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1058–1066, 1997.

[35] S. Mallat, "A theory for multiresolution signal processing: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693.

[36] J.-P. Antoine, D. Barachea, R. M. C. Jr., and L. da Fontoura Costa, "Shape characterization with the wavelet transform," *Signal Processing*, vol. 62, no. 3, pp. 674–693, 1997.

[37] C. H. Yeh, "Wavelet-based corner detection using eigenvectors of covariance matrices," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2797–2806, 2003.

[38] J. O. Chapa and M. R. Raghuveer, "Optimal matched wavelet construction and its application to image pattern recognition," *Proceedings of SPIE*, vol. 2491, no. 1, pp. 518–529, 1995.

[39] J. Liang and T. W. Parks, "A translation-invariant wavelet representation algorithm with applications," *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 224–232, 1996.

[40] R. A. Maestre, J. Garcia, and C. Ferreira, "Pattern recognition using sequential matched filtering of wavelet coefficients," *Optics Communications*, vol. 133, pp. 401–414, 1997.

[41] F. Murtagh, J.-L. Starck, and M. W. Berry, "Overcoming the curse of dimensionality in clustering by means of the wavelet transform," *The Computer Journal*, vol. 43, no. 2, pp. 107–120, 2000.

[42] T. Yu, E. C. M. Lam, and Y. Y. Tang, "Feature extraction using wavelet and fractal," *Pattern Recognition Letters*, vol. 22, pp. 271–287, 2001.

[43] D.-M. Tsai and C.-H. Chiang, "Rotation-invariant pattern matching using wavelet decomposition," *Pattern Recognition Letters*, vol. 23, no. 1-3, pp. 191–201, 2002.

[44] T. Du, K. B. Lim, G. S. Hong, W. M. Yu, and H. Zheng, "2d occluded object recognition using wavelets," in 4^{th} *International Conference on Computer and Information Technology*, 2004, pp. 227–232.

[45] N. Saito and R. R. Coifman, "Local discriminant bases," *Proceedings of SPIE*, vol. 2303, no. 2, pp. 2–14, 1994.

[46] S. Livens, P. Scheunders, G. V. de Wouwer, D. V. Dyck, H. Smets, J. Winkelmans, and W. Bogaerts, "2D occluded object recognition using wavelets," in *Computer Analysis of Images and Patterns V*, V. Hlavác and R. Sára, Eds. Berlin: Springer-Verlag, 2004, pp. 538–543.

[47] I. N. Tansel, C. Mekdeci, O. Rodriguez, and B. Uragun, "Monitoring drill conditions with wavelet based encoding and neural networks," *International Journal of Machine Tools & Manufacture*, vol. 33, pp. 559–575, 1993.

[48] I. N. Tansel, C. Mekdeci, and C. McLaughlin, "Detection of tool failure in end milling with wavelet transformations and neural networks (WT-NN)," *International Journal of Machine Tools & Manufacture*, vol. 35, pp. 1137–1147, 1995.

[49] R. E. Learned and A. S. Wilsky, "A wavelet packet approach to transient signal classification," *Applied and Computational Harmonic Analysis*, vol. 2, pp. 265–278, 1995.

[50] Y. Wu and R. Du, "Feature extraction and assessment using wavelet packets for monitoring of machining processes," *Mechanical Systems and Signal Processing*, vol. 10, pp. 29–53, 1996.

[51] T. J. case and R. C. Waag, "Flaw identification from time and frequency features of ultrasonic waveforms," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 43, no. 4, pp. 592–600, 1996.

[52] R. Drai, N. Khelil, and A. Benchaala, "Time frequency and wavelet transform applied to selected problems in ultrasonics nde," *NDT&E International*, vol. 35, no. 8, pp. 567–572, 2002.

[53] M. Buonsanti, M. Cacciola, S. Calcagno, F. C. Morabito, and M. Versaci, "Ultrasonic pulse-echoes and eddy current testing for detection, recognition and characterisation of flaws detected in metallic plates," *Proceedings of the* 9^{th} *European Conference on Nondestructive Testing*, vol. 11, no. 11, 2006.

[54] R. Momenan, M. H. Loew, M. F. Insana, R. F. Wagner, and B. S. Garra, "Application of pattern recognition techniques in ultrasound tissue characterization," *Proceedings of the* 10^{th} *International Conference on Pattern Recognition*, vol. 1, pp. 608–612, 1990.

[55] I. N. Bankman, K. O. Johnson, and W. Schneider, "Optimal detection, classification, and superposition resolution in neural waveform recordings," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 8, pp. 836–841, 1993.

[56] T. Kalayci and O. Özdamar, "Wavelet preprocessing for automated neural network detection of eeg spikes," *IEEE Engineering in Medicine and Biology*, vol. 14, pp. 160–166, 1995.

[57] R. Tate, D. Watson, and S. Eglen, "Using wavelets for classifying human in vivo magnetic resonance spectra," in *Wavelets and Statistics*, A. Antoniadis and G. Oppenheim, Eds. New York: Springer-Verlag, 1995, pp. 377–383.

[58] A. Mojsilovic, M. V. Popovic, A. N. Neskovic, and A. D. Popovic, "Wavelet image extension for analysis and classification of infarcted myocardial tissue," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 9, pp. 856–866, 1995.

[59] G. Georgiou and F. S. Cohen, "Tissue characterization using the continuous wavelet transform. i. decomposition method," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 48, no. 2, pp. 355–363, 2001.

[60] G. Georgiou, F. S. Cohen, C. W. Piccoli, F. Forsberg, and B. B. Goldberg, "Tissue characterization using the continuous wavelet transform. ii. application on breast rf data," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 48, no. 2, pp. 364–373, 2001.

[61] W.-L. Lee, Y.-C. Chen, and K.-S. Hsieh, "Ultrasonic liver tissues classification by fractal feature vector based on m-band wavelet transform," *IEEE Transactions on Medical Imaging*, vol. 22, no. 3, pp. 382–392, 2003.

[62] B. Alacam, B. Yazici, N. Bilgutay, F. Forsberg, and C. Piccoli, "Breast tissue characterization using farma modeling of ultrasonic rf echo," *Ultrasound in Medicine and Biology*, vol. 30, no. 10, pp. 1397–1407, 2004.

Chapter 2

Fundamentals

The previous chapter described pattern classification and its application in the relevant literature. But pattern classification is a mathematical subject involving statistics, and therefore must be described with a certain amount of formal mathematics. The application of pattern classification within this work also requires knowledge of signal analysis, including wavelets, and image recognition. The particular tool used for feature extraction, the Dynamic Wavelet Fingerprint (DWFP) [1] will also be described.

We will begin with a sample data set as a pedagogical tool for understanding pattern classification.

2.1 Fisher Iris Data Set

The most commonly-used example of pattern recognition is called the Fisher Iris data set [2]. The data set consists of measurements made on different types of iris flowers collected by Edgar Anderson in 1935 as he was working in the field of taxonomy. The iris data set was widely popularized by Sir Ronald Aylmer Fisher when he was able to successfully apply statistical pattern recognition techniques to discriminate between one type of flower and the other two using only those taxonomic measurements [3]. There were three different types of flowers in the data set: *Iris setosa*, *Iris versicolor*, and *Iris virginica.* Fifty individuals from each species were studied, and four measurements were made on each individual. These measurements include:

1. sepal length [cm]
2. sepal width [cm]
3. petal length [cm]
4. petal width [cm]

Figure 2.1 shows photographs of two types of the irises used with the petals and sepals indicated[1]. The petals are small, usually upright with the longer sepal arms arcing underneath the petals. The length and width measurement of a sepal is also indicated in Fig. 2.1b.

Figure 2.2 shows histograms of each of the four measurements among the three different types of flowers. As the histograms show, there is a lot of overlap between the measurements of sepal length and sepal width between the three types of flowers. But *Iris setosa* has much smaller measurements of petal length and petal width than the other two types of irises. If these 150 measurements were sufficient to describe all the populations of these types of irises, our job would now be done. We could merely set a threshold on the measurements of petal size so that any flower with a petal length less than 2.5cm or a petal width less than 0.75cm would be classified as *Iris setosa.* In fact, we will do this later by devising a linear discriminant. But this is not sufficient to separate the other two types of flowers, and we may be uncertain that these flower

[1] *Iris versicolor* image by Jennifer Anderson, and *Iris virginica* image by Robert H. Mohlenbrock [4]. These photos are from the USDA PLANTS database and are used with permission [5].

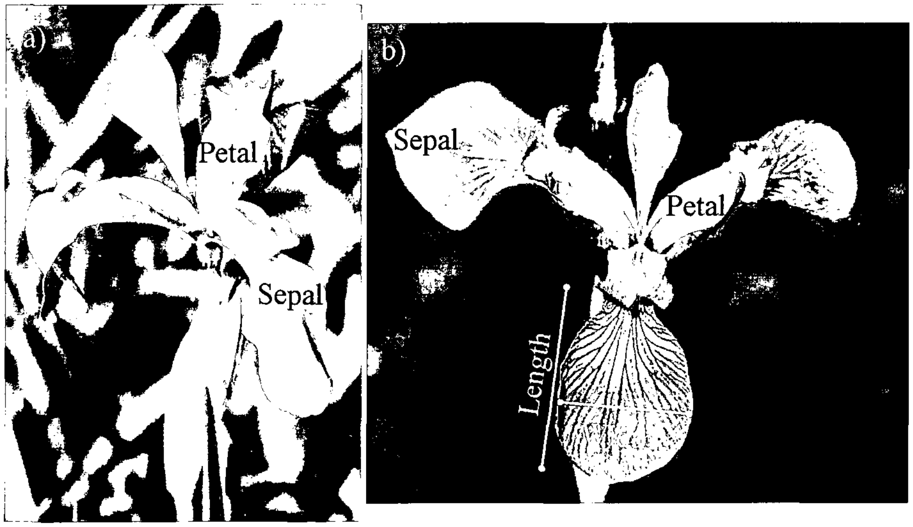


Figure 2.1: Two different types of irises from the Fisher Iris data set, including a) *Iris veriscolor* and b) *Iris virginica* from the USDA-NRCS PLANTS database [5]. The petals and sepals are indicated in the photos, with the length and width measurement of a sepal displayed in the second photo.

samples are representative of the whole population. In order to identify irises by their petal and sepal measurements, we will require more sophisticated mathematical tools, specifically, statistical pattern recognition.

2.2 Statistical Pattern Recognition

The goal of pattern recognition is to classify different objects into categories (*classes*) based on measurements made on those objects (*features*). In the Fisher Iris example, the different objects are the 150 different individual flowers studied, and the classes are the three different types of flowers. Pattern recognition is a subset of machine learning, in which computers use algorithms to learn from data. As described above, it would be possible to set a threshold on petal width or length in order to classify the *Iris setosa* as separate from the other two varieties. Statistical pattern recognition, however, requires a statistical characterization of the likelihood of each object belong-

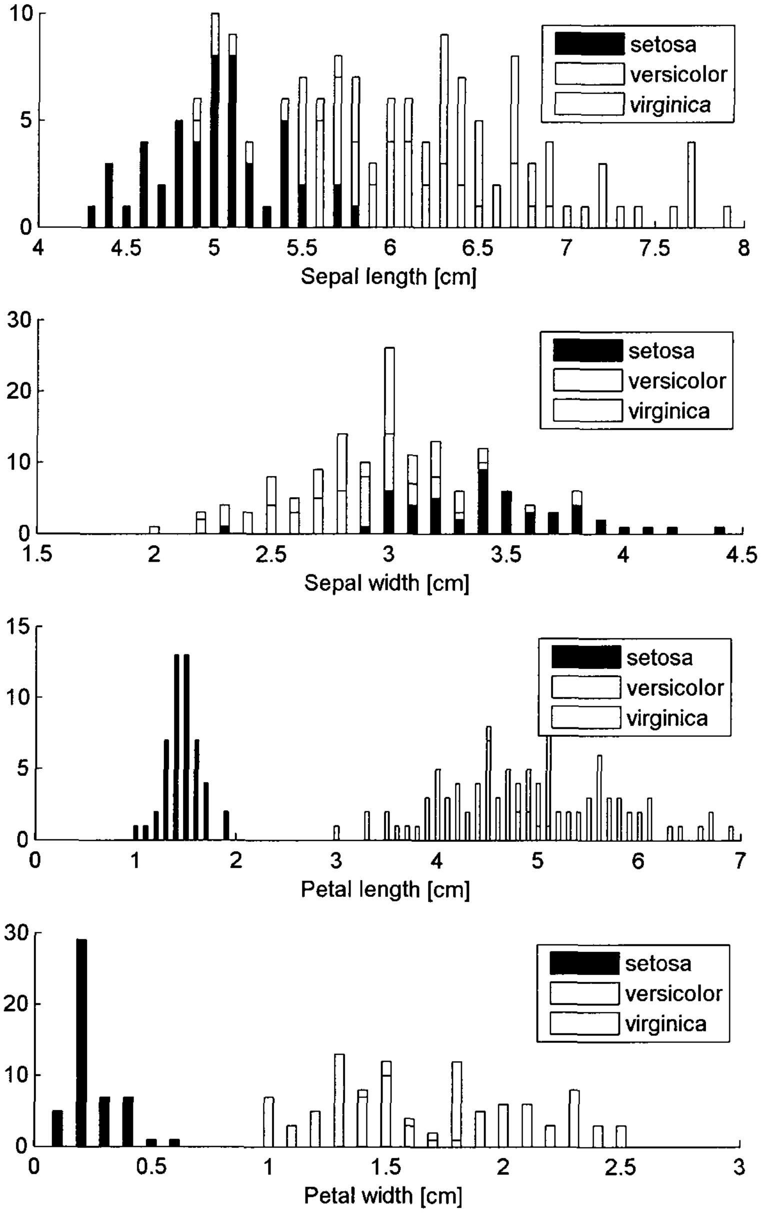


Figure 2.2: The distribution of measurements (features) for each of the three type of flowers (classes) from the Fisher Iris data set are shown here.

Table 2.1: A limited selection of the Fisher Iris data set is presented here to illustrate features and classes

	Features ($\mathbf{x}_i$)				
	Sepal	Sepal	Petal	Petal	
	Length	Width	Length	Width	
i	[cm]	[cm]	[cm]	[cm]	Class
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
⋮	⋮	⋮	⋮	⋮	⋮
50	5	3.3	1.4	0.2	setosa
51	7	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
54	5.5	2.3	4	1.3	versicolor
55	6.5	2.8	4.6	1.5	versicolor
56	5.7	2.8	4.5	1.3	versicolor
57	6.3	3.3	4.7	1.6	versicolor
58	4.9	2.4	3.3	1	versicolor
59	6.6	2.9	4.6	1.3	versicolor
60	5.2	2.7	3.9	1.4	versicolor
⋮	⋮	⋮	⋮	⋮	⋮
100	5.7	2.8	4.1	1.3	versicolor
101	6.3	3.3	6	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3	5.9	2.1	virginica
104	6.3	2.9	5.6	1.8	virginica
105	6.5	3	5.8	2.2	virginica
106	7.6	3	6.6	2.1	virginica
107	4.9	2.5	4.5	1.7	virginica
108	7.3	2.9	6.3	1.8	virginica
109	6.7	2.5	5.8	1.8	virginica
110	7.2	3.6	6.1	2.5	virginica
⋮	⋮	⋮	⋮	⋮	⋮
150	5.9	3	5.1	1.8	virginica

ing to a particular class. Pattern recognition usually proceeds in two different ways: *unsupervised* pattern recognition draws distinctions or clusters in the data without taking into account the actual class labels, while *supervised* pattern recognition uses the known class labels along with the statistical characteristics of the measurements to identify objects with the class that reduces the error. We will focus our attention on supervised pattern classification. The following description of pattern recognition is adapted from [6], with some additions from [7].

Table 2.1 shows a limited selection from the Fisher Iris data set. The rows correspond to the individual flowers that were measured and are indexed by i. In general, this index i has a range of values $i = 1, \ldots, N$, where N is the number of observations ($N = 150$ here). The four middle columns list the features measured from each individual, where there are in general M-many features, so $M = 4$ for the Fisher Iris data set. Therefore, $\mathbf{x}_i$ is the feature vector of measurements for object i. For example,

$$\mathbf{x}_9 = (4.4, 2.9, 1.4, 0.2).$$

The last column lists the class ω_k, $k = 1, 2, 3$ corresponding to each individual in the row, where ω_1=*Iris setosa*, ω_2=*Iris versicolor*, and ω_3=*Iris virginica.* We will often also use w to represent the vector of class labels associated with each feature vector, so

$$w_9 = \omega_1.$$

To make a decision about the most likely class given a known feature vector, we need to know some information about the distribution of features in the classes (Fig. 2.2). Let $P(\omega_k)$ represent the *a priori* probability (or prior probability) of class ω_k

occurring. In the Fisher Iris example, there are three classes of equal populations, so

$$P(\omega_1) = P(\omega_2) = P(\omega_3) = 0.33.$$

Let $p(\mathbf{x}|\omega_k)$ represent the class-conditional probability density function, which is the probability that the feature is $\mathbf{x}$ given that the class is ω_k. The class-conditional probability density function might be a known distribution of features for each class, but if there is no known theoretical probability density function for the features, then the experimental distribution of features in each class may be used (such as Fig. 2.2 where the probabilities sum to unity). Then given the prior probabilities $P(\omega_k)$ and the class-conditional densities $p(\mathbf{x}|\omega_k)$, we can derive the posterior probability using Bayes theorem [7]:

$$P(\omega_k|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_k)P(\omega_k)}{\sum_{k=1}^{3} p(\mathbf{x}|\omega_k)P(\omega_k)} \tag{2.1}$$

In general, the sum in the denominator in Eqn. 2.1 is over $k = 1, \ldots, C$, where there are C classes. The left hand side of Eqn 2.1, $P(\omega_k|\mathbf{x})$ is the probability that the object belongs to class ω_k given that the feature vector is $\mathbf{x}$.

Equation 2.1 can be rewritten as a decision rule [6]: assign $\mathbf{x}$ to ω_k if

$$p(\mathbf{x}|\omega_k)p(\omega_k) > p(\mathbf{x}|\omega_j)p(\omega_j) \quad j, k = 1, \ldots, C; \quad k \neq j \tag{2.2}$$

Equations 2.1 and 2.2 achieve our stated goal - to predict the most likely class of each object given its measured features. However, in reality $p(\mathbf{x}|\omega_i)$ is rarely known even if a form can be assumed from the empirical data. What follows are other pattern classifiers often used in the following chapters, some of which are parametric classifiers and assume a form for $p(\mathbf{x}|\omega_k)$, and others that are nonparametric.

2.2.1 Classifier Maps

For completeness, a description of the classifiers is provided below, but more complete information can be found in [6], or any standard pattern classifier text.

Quadratic and Linear Discriminant Classifiers (QDC and LDC)

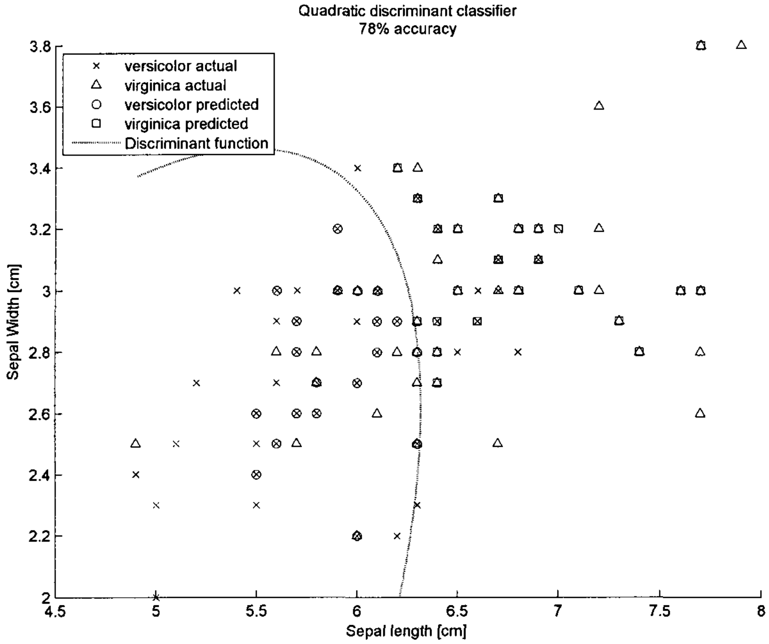


Figure 2.3: The sepal width and sepal length features from the two least distinguishable irises from the Fisher Iris data set are plotted here, along with the quadratic discriminant boundary formed.

Instead of assuming the form of $p(\mathbf{x}|\omega_k)$, we will introduce a discriminant function $g(\mathbf{x})$. A discriminant function can be very simple, such as thresholding. In the Fisher Iris case, if we consider only two classes ω_1, ω_2, we could write

$$\begin{aligned} g(x_3) &< 2.5 \rightarrow \mathbf{x} \in \omega_1 \\ &> 2.5 \rightarrow \mathbf{x} \in \omega_2 \end{aligned}$$

because a simple threshold on petal length is capable of distinguishing *Iris setosa* and *Iris versicolor.*

For a general number of classes C, we need C-many discriminant functions $g_i(\mathbf{x})$ such that

$$g_i \geq g_j \rightarrow \mathbf{x} \in \omega_i \quad i,j = 1 \ldots, C \quad i \neq j \tag{2.3}$$

In this way, Equation 2.2 could be reproduced by allowing $g_i(\mathbf{x}) = p(\mathbf{x}|\omega_i)p(\omega_i)$.

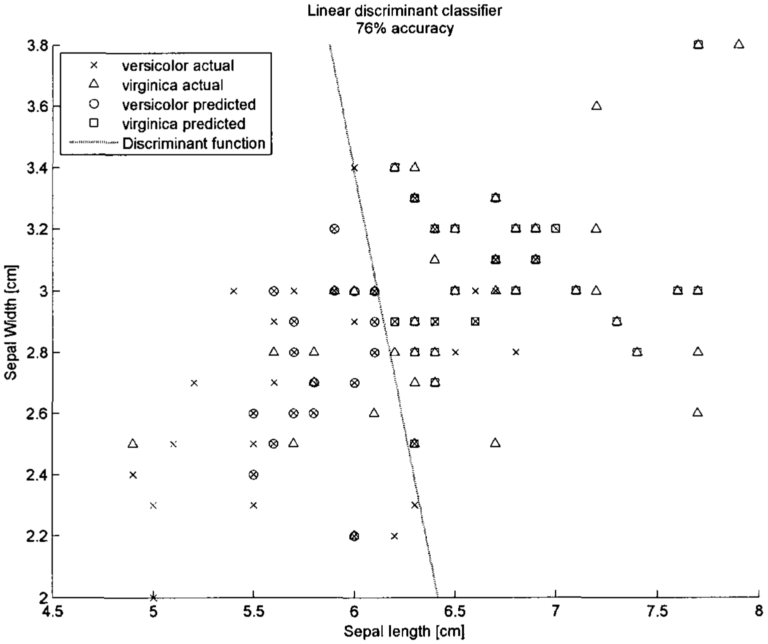


Figure 2.4: The same data from Fig. 2.3 is plotted here but a linear discriminant classifier was used.

Forms for the discriminant function can be derived [6] by assuming a Gaussian distribution for $p(\mathbf{x}|\omega_i)$. This leads to a discriminant function $g_i(\mathbf{x})$ given by

$$g_i(\mathbf{x}) = \log(p(\omega_i)) - \frac{1}{2}\log(|\hat{\Sigma}_i|) - \frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^T \hat{\Sigma}_i^{-1}(\mathbf{x} - \mathbf{m}_i) \tag{2.4}$$

where $\mathbf{x}$ is the feature vector as before, $\hat{\Sigma}_i$ represents the sample covariance matrix, and $\mathbf{m}_i$ represents the maximum likelihood estimate of the mean of the features averaged over $i = 1, \ldots, N$. The sample covariance matrix and mean of the features are statistics of a data set thought to be random and are often used in multivariate statistics. In classification, these quantities are estimated from the training data set:

$$\begin{aligned} \mathbf{m} &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathbf{x}_i \\ \hat{\mathbf{\Sigma}} &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \end{aligned} \tag{2.5}$$

Using the estimates of $\mathbf{m}_i$ and $\hat{\mathbf{\Sigma}}_i$ in Eqn 2.4 results in the *quadratic discriminant classifier* (QDC). Figure 2.3 shows the results of applying the quadratic discriminant classifier using the sepal length and sepal width features from *Iris versicolor* and *Iris virginica*. The training and testing sets were randomly drawn from half of the available observations, along with the decision boundary formed by the classifier. Those measurements without a predicted value were included in the training set alone.

The *linear discriminant classifier* (LDC) is as above except the class covariance matrices $\mathbf{\Sigma}_i$ are all the same, so we substitute $\mathbf{S}_W$, which is the common group covariance matrix:

$$g_i(\mathbf{x}) = \log(p(\omega_i)) - \frac{1}{2}\mathbf{m}_i^T\mathbf{S}_W^{-1}\mathbf{m}_i + \mathbf{x}^T\mathbf{S}_W^{-1}\mathbf{m}_i \tag{2.6}$$

with the common group covariance matrix given by

$$\mathbf{S}_W = \frac{N}{N-C}\sum_{i=1}^{C}\frac{N_i}{N}\hat{\mathbf{\Sigma}}_i$$

Figure 2.4 shows the application of the linear discriminant classifier to the Fisher Iris data set. As the figure shows, the linear discriminant classifier is less accurate

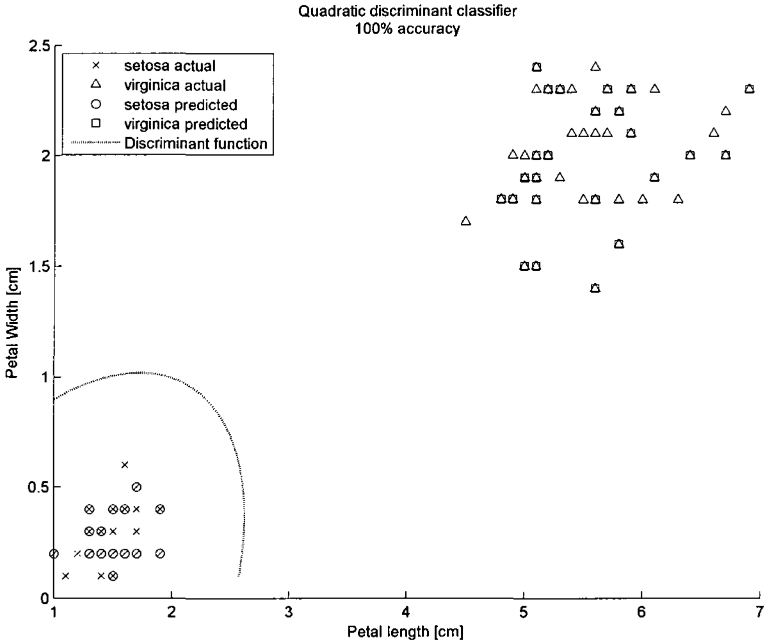


Figure 2.5: A similar classification technique as Fig. 2.4 was performed here to discriminate between Iris setosa and Iris virginica using only petal length and petal width.

and less flexible than the quadratic discriminant classifier applied to the same data, as in Figure 2.3. Note that the training and testing sets from the two examples are the same, so the accuracy can be directly compared. In this example, LDC performs slightly worse. Fisher showed in his paper that linear discriminants are adequate at classifying one of the irises from the others, though not the one selected in this example. Instead, Figure 2.5 shows similar classification results discriminating between Iris setosa and Iris virginica using quadratic discriminants with 100% accuracy.

Lastly, a third discriminant classifier was used, which is a combination of LDC and QDC. The computation of QDC requires inverting the sample covariance matrix, $\hat{\Sigma}$. Computational difficulties occur whenever $\hat{\Sigma}$ is singular. This difficulty can be overcome by calculating the Moore-Penrose pseudoinverse of $\hat{\Sigma}$ instead of $\hat{\Sigma}^{-1}$ [8]. However, singularity of $\hat{\Sigma}$ can also occur whenever QDC is not an appropriate classifier choice, perhaps because the class covariance matrices do not differ. In such cases, it may be more appropriate to use LDC. Therefore, the third discriminant classifier used, called QDC-LDC, applies QDC whenever $\hat{\Sigma}$ is not singular and LDC otherwise.

k-Nearest-Neighbor (kNN)

A nonparametric classifier does not require knowledge of the class-conditional probability densities. LDC and QDC make simplifying assumptions about the densities, such as that they are normally-distributed. In kNN, the density approximation is given by [6]

$$p(\mathbf{x}) = \frac{k}{NV} \tag{2.7}$$

where k is an integer, V is a given volume, and N is the number of measurements, as before. The kNN method fixes the integer k for the number of samples N and looks at the volume V of feature space centered at $\mathbf{x}$. For a given volume V, a measurement

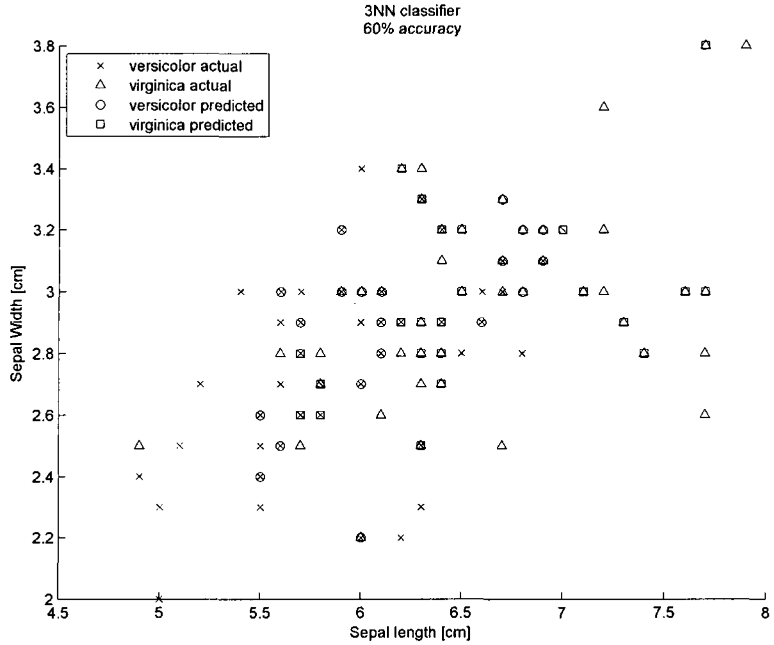


Figure 2.6: The same Fisher data set as Fig. 2.3 and 2.4 was classified using the kNN classifier for $k = 3$. No discriminant can be plotted but the accuracy of the classifier is clearly low.

$\mathbf{x}$, and its kth nearest neighbor $\mathbf{x}_k$, the volume is a hyper-sphere centered at $\mathbf{x}$ of radius $\|\mathbf{x} - \mathbf{x}_k\|$. The decision rule then assigns $\mathbf{x}$ to the most frequently-occurring class in the volume, so that $\mathbf{x}$ is assigned to class ω_i if

$$p(\omega_i|\mathbf{x}) \geq p(\omega_j|\mathbf{x}), \forall j. \tag{2.8}$$

In sum, the decision rule assigns $\mathbf{x}$ to the largest-populated class in the volume of feature space V centered at that point.

Support Vector Machines (SVM)

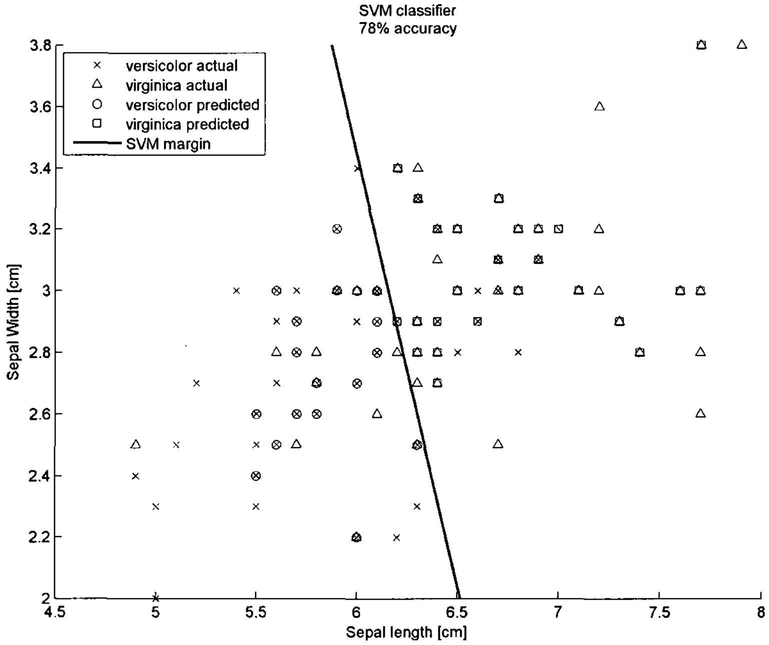


Figure 2.7: The same Fisher data set as Fig. 2.3, 2.4, and 2.6 was classified using a linear SVM classifier. The margin maximized by the optimization calculation is indicated.

SVMs are uniquely suited to binary (two-class) problems, such as the RFID pattern classification algorithm discussed in Chapter 6. The goal is to map feature vectors to a higher dimensional space where a hyperplane that maximizes the margin in between the classes is found. Given a feature vector $\mathbf{x}_i$, $i = 1, \ldots, n$, and two classes, ω_1, ω_2 with labels $y = \pm 1$, we can write the linear discriminant function as

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \tag{2.9}$$

so that if $g(\mathbf{x}) > 0$, assign $\mathbf{x}$ to class ω_1 with label $y_i = 1$. Similarly, if $g(\mathbf{x}) < 0$, assign $\mathbf{x}$ to class ω_2 with label $y_i = -1$. Then correct classification requires finding a margin $b > 0$ such that

$$y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq b \tag{2.10}$$

The support vectors are used to generate the maximal margin in order to optimize computation time. In practice, a kernel formulation is often used to solve the SVM problem [9]. Figure 2.7 shows the classification results applying a support vector machine classifier to the same data set as the previous figures for QDC, LDC, and kNN. The training and testing sets remain the same. Note that the accuracy is quite high though the margin is linear, like LDC.

For multiclass problems, combinations of binary classifiers are used. There are two main techniques [10]:

1. one-versus-one: The classifier is trained on samples drawn only from two classes, ω_i and ω_j, and tested with samples drawn from the remaining classes (and possibly from classes ω_i and ω_j not already used for training). The process is repeated for all unique binary pairs of ω_i, ω_j, $i, j = 1, \ldots, C$. A decision is made by a winner-take-all rule.

2. one-versus-all: The class labels are redrawn so that class ω_i becomes class ω_1, and all other classes become ω_2. The process is repeated for all ω_i, $i = 1, \ldots, C$.

The decision is made by assigning the predicted class to the ω_i with the highest $g(\mathbf{x})$.

2.2.2 Sampling

In applying pattern recognition techniques to finite, real-world data sets, merely classifying the data is not sufficient. To train the classifier and also evaluate its performance, we have to withhold some of the data to test the classifier. A classifier that is tested with a subset of the data it was trained on is said to be *overtrained*, which means the classifier may perform very well on data it was trained on but its performance on unseen data is not measured [11]. Therefore, for finite, real-world data sets, the available observations need to be split into a subset used for training and a subset used for testing. First, we will discuss the type of sampling methods used in this work. Then we will address the problem of class imbalance encountered in real-world data sets and describe how this alters these standard sampling methods.

Sampling Methods

Let $\mathbf{X}$ represent the available data set, where $\mathbf{X}$ is formed of all the feature vectors $\mathbf{x}_i$, $i = 1, \ldots, N$ of $M-$many features each. Let $\mathcal{C}$ represent the classifier of choice, and $\mathbf{X}_R$ and $\mathbf{X}_S$ represent the data used for training and testing, respectively, where $R, S \subset \{1, \ldots, N\}$. Then the types of sampling methods include [11]:

1. Hold-out: Split $\mathbf{X}$ once into $\mathbf{X}_R$ and $\mathbf{X}_S$, so that $|R| + |S| = N$. Traditionally $|R| = |S| = \frac{1}{2}N$ is used. Predicted classes y_i are only available for those $\mathbf{x}_i$ selected for testing. This is the method used to generate the Figures 2.3-2.7. The hold-out method is often selected for ease of computation on large data sets. However, splitting $\mathbf{X}$ into $\mathbf{X}_R$ and $\mathbf{X}_S$ in this way can lead to an unfortunate

split, so that the results are pessimistically biased.

2. Leave-one-out: A version of *cross-validation*, leave-one-out sets $|S| = 1$, $|R| = N - 1$. The classifier $\mathcal{C}$ is trained on $\mathbf{X}_R$ and tested on $\mathbf{X}_S$. The process is repeated for all $S = 1, \ldots, N$ and $R = \{1, \ldots, N\} - S$. In this way, each $\mathbf{x}_i$ becomes associated with one predicted value y_i, and no averaging is required.

3. Bootstrap: Randomly select $S \in \{1, \ldots, N\}$ so that $|S| < N$. Train $\mathcal{C}$ on the remainder $R = \{1, \ldots, N\} - S$ and test it on $\mathbf{X}_S$. Repeat the random selection of S, possibly with replacement, and determine the error of the classifier by averaging the error of each individual selection of $R, S \in \{1, \ldots, N\}$.

Class Imbalance

The sampling methods listed above assume an equal cost for selecting members of each class ω_j, $j = 1, \ldots, C$ for training and testing. A new problem emerging in the literature is the problem of *class imbalance*, in which the data set contains unequal representations of the classes ω_j and one or more classes are much more heavily represented than the others. In an early example, Lewis and Catlett studied text characterization using ten different words with unequal representations and found that when the true occurrence of a word in text was rare, the classifier stubbornly refused to recognize any of the positive cases and classified all the test samples as negative [12]. In this example, only 0.2% of the examples were positive, so 99.8% accuracy could be obtained by merely classifying all of the test cases as negative. The classifier is performing as expected - it minimizes the misclassification by assigning all of the test cases to be negative, even though not all of them are negative. This observation also demonstrates the relative uselessness of total accuracy on unbalanced data sets. Instead, the measure most often used in this work is average accuracy per class,

$$A(\omega_k) = \frac{|(w_i = \omega_k)\&(y_i = \omega_k)|}{|(w_i = \omega_k)|}, \; i = 1, \ldots, N; \; k = 1, \ldots, C \qquad (2.11)$$

As before, ω_k represents the classes, w_i represents the vector of class labels associated with each observation $\mathbf{x}_i$, and y_i represents the predicted class vector.

The problem of class imbalance has achieved some attention and solutions to the problem vary. Many were tested and studied on binary class problems. Some include [13]:

1. Oversampling: The minority-represented class is resampled to present more representation in the data set [14]

2. Undersampling: The majority-represented class is sampled at a lower rate to bring its representation in the data set closer to that of the minority-represented class [15].

3. Sampling Compromise: A combination of oversampling and undersampling is performed to reach a compromise between the two strategies [16].

In these strategies, the exact amount of undersampling and oversampling depends on the domain. Naively, one might assume that the class sizes should be equal, as they are in the Fisher Iris data set (and many other synthetic data sets). However, some researchers have found success by varying the proportion of class representation through undersampling or oversampling [13, 16]. Sometimes the bootstrap method is sufficient to overcome the class imbalance problem alone, and other methods include weighting training instances by class representation [15]. The method chosen in the following chapters is undersampling, sometimes with the inclusion of bootstrap sampling.

As an example of the class imbalance problem, consider the Fisher Iris data set.

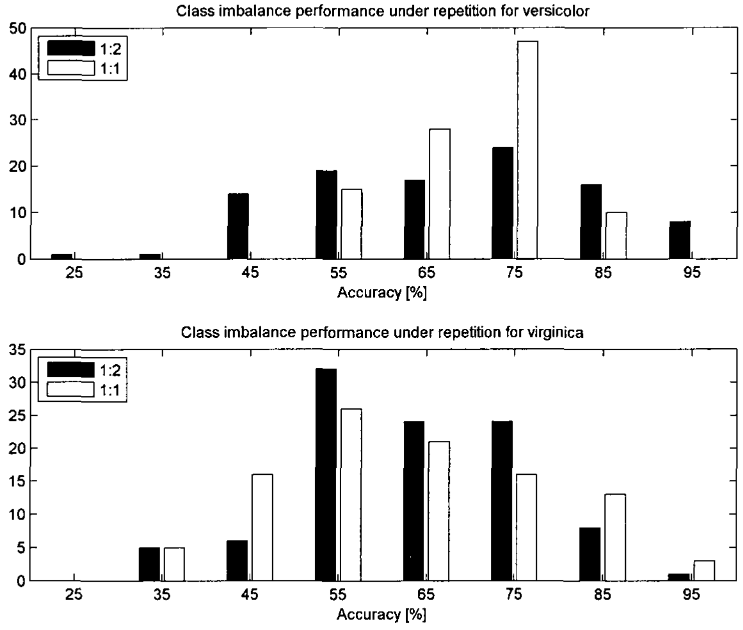


Figure 2.8: The same Fisher data set as Fig. 2.3, 2.4, 2.6, and 2.7 was classified using QDC with holdout sampling, but with different proportions of class representations, 1 : 2 and 1 : 1 versicolor:virginica. The first plot shows the accuracy of the classifier for the unbalanced and balanced data sets at correctly identifying *Iris versicolor* under 100 repetitions of the sampling and classification. The second plot shows the accuracy for *Iris virginica* under repetition.

There are equal representations of the classes, and previous examples have displayed the accuracy of different classifier techniques, all of which used the holdout sampling technique. To illustrate the class imbalance problem, we randomly chose half of the *Iris versicolor* observations to be used for classification with all 50 of the *Iris virginica* observations, so that the ratio of versicolor to virginica examples was $1:2$. In order to remain consistent with former examples, only sepal length and width were selected as features. We performed holdout classification using QDC and measured the accuracy per class, $A(\omega_k)$. At the same time, we performed a similar classification using the same randomly selected training sets and testing sets but with additional versicolor samples so that the ratio of versicolor to virginica observations matched the natural $1:1$ distribution. In this second case, we also performed holdout QDC but measured the accuracy per class only on the testing samples held in common between the $1:2$ and $1:1$ examples.

Figure 2.8 shows the results for the classification of both flowers under balanced and unbalanced class representation for 100 repetitions of the holdout sampling and QDC classification. The histograms show that the $1:1$ representation has more highly-accurate classifier results while the $1:2$ unbalanced data set has more less-accurate classifier results for the minority class (*Iris versicolor*). The majority class, however, is more equally represented by both the $1:2$ and $1:1$ data sets. This figure demonstrates how the classifier accuracy can be skewed under a very small class imbalance, $1:2$. In the chapters that follow, the class imbalance can be quite a bit higher, such as $1:54$ for some pairs of classes.

2.2.3 Summary

In the work above, we have introduced Bayes formula for statistical pattern recognition, summarized some types of classifiers and shown examples of their application to the Fisher Iris data set. We have also described different ways of splitting finite data sets into training and testing sets for the purpose of applying pattern recognition to real-world examples. Lastly, the problem of class imbalance has been addressed and its importance illustrated through a simple example.

In the work that follows, these methods will be relied upon to classify time domain signals. However, two main requirements for the successful application of pattern recognition have not been addressed: feature extraction and feature selection. In what follows, one main feature extraction technique using Dynamic Wavelet Fingerprinting will be discussed. Feature selection has been used only moderately in the following chapters mainly due to computational time constraints. We have focused on selecting features for each application based on physical insight instead of applying automatic feature selection techniques, such as Principal Component Analysis, that require long computation times.

2.3 Wavelets

Wavelets were developed in order to introduce a local formulation of time-frequency analysis for signals and were first used by Jean Morlet in 1982 to analyze nonstationary signals [17]. The goal of wavelet analysis is to convert a signal into coefficients that can be used for compression, filtering, storage, reconstruction, or other types of analysis [18]. The following description of wavelets was adapted mostly from [19] with some notational changes and filter descriptions from MATLAB's Wavelet Toolbox documentation[2].

Wavelet transforms are a form of time-frequency analysis of a signal. One of the

[2] We used the Wavelet Toolbox help menu from MATLAB ®2009a (The Mathworks, 2008, Natick, MA), as well as additional information from the Mathworks website, including the "Advanced Topics: Fast Wavelet Transform (FWT) Algorithm" at http://www.mathworks.com/access/helpdesk/help/toolbox/wavelet/ch06_ad4.html

most common transformations for examining the frequency content of a signal $f(t)$ is the Fourier transform,

$$\mathcal{F}(\omega) = \frac{1}{\sqrt{2\pi}} \int e^{-i\omega t} f(t)\, dt \tag{2.12}$$

where we have assumed $f(t)$ is a continuous function. However, Equation 2.12 is not well-localized in time. A better transformation for time-frequency localization is the *continuous wavelet transform*,

$$C(a,b) = \int_{-\infty}^{+\infty} f(t)\psi_{a,b}(t)\, dt. \tag{2.13}$$

In Equation 2.13, $\psi(t)$ is the mother wavelet, and $\psi_{a,b}$ is given by

$$\psi_{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right). \tag{2.14}$$

The constants a, b are real numbers, where a is called the scale factor and relates to how much the mother wavelet is stretched or compressed, and b is the translation factor and relates to the time localization of ψ [19]. For each value of a, b there is one value of the integral $C(a,b)$ in Eqn 2.13. Different values of the scale correspond to different frequency components of the signal. For example, if a is small, then the wavelet is compressed, and rapidly changing details of the signal are visible. This is related to the high frequency component of the signal. Similarly, large a relates to low frequency information about the signal.

The continuous wavelet transform is only useful for continuous signals, but a full time-frequency localization requires calculating $C(a,b)$ for all $a, b \in \mathbb{R}$. The *discrete wavelet transform* is given by

$$C(m,n) = a_0^{-m/2} \int f(t)\psi(a_0^{-m}t - nb_0)\, dt \tag{2.15}$$

where the conversion from Eqn. 2.13 to Eqn 2.15 is made by

$$\begin{aligned} a &= a_0^m \\ b &= nb_0a_0^m \\ m,n &\in \mathbb{Z} \end{aligned} \tag{2.16}$$

In order for the $\psi_{m,n}$ to form an orthonormal basis, we need to pick $a_0 = 1$, $b_0 = 1/2$, so that [20]

$$\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n) \tag{2.17}$$

This later leads to the fact that, since the constants a, b are factors of two, some a, b values are redundant.

In order for wavelets to satisfy requirements for multiresolution, in which the resolution increases as the constants decrease, wavelet bases require a scaling function in addition to the mother wavelet. In general, this scaling function is defined by

$$\phi_k(t) = \phi(t-n),\ n \in \mathbb{Z},\ \phi \in \mathbb{L}^2 \tag{2.18}$$

We want a scaling function that can be defined like the mother wavelet in 2.17, with time and scale constants, yielding [20]

$$\phi_{m,n}(t) = 2^{m/2}\phi(2^m t - n) \tag{2.19}$$

Figure 2.9 shows the scaling function ϕ and mother wavelets ψ for some of the wavelet transforms used later in this work.

In order to satisfy multiresolution, we need the subspace created by the span of scaling functions to be nested in an exclusionary way. That is, let $V_m = \text{Span}\{\phi_{m,n}(t)\}$ represent the subspace created by the span of scaling functions, and if we insist that

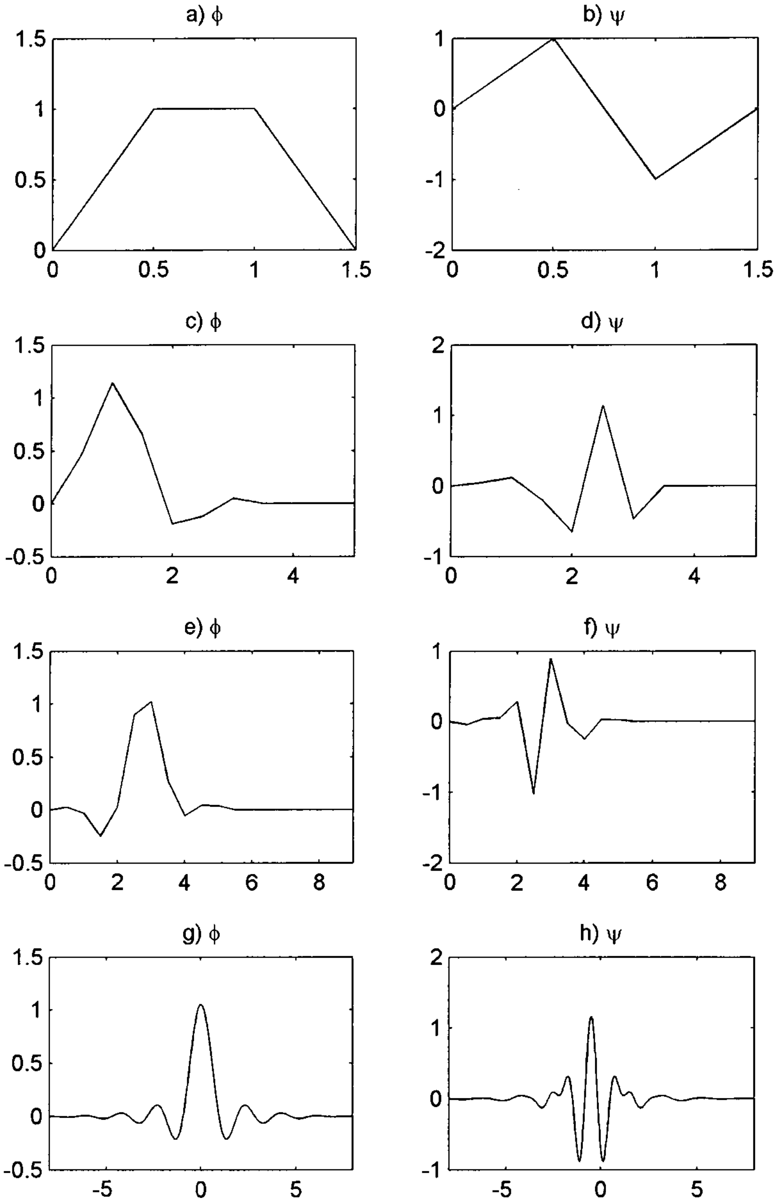


Figure 2.9: The scaling function ϕ and mother wavelet ψ for a)-b) Haar, c)-d) Daubechies-3 (db3), e)-f) Symelet-5 (sym5), and g)-h) Meyer (meyr) wavelets.

the subspaces are nested so that $V_m \subset V_{m+1}$ to satisfy the multiresolution condition:

$$f(t) \in V_m \iff f(2t) \in V_{m+1} \tag{2.20}$$

then the scaling function and the mother wavelet yield the *two-scale relations* [20]

$$\begin{aligned} \phi(t) &= \textstyle\sum_{n\in\mathbb{Z}} h(n)\sqrt{2}\phi(2t-n) \\ \psi(t) &= \textstyle\sum_{n\in\mathbb{Z}} g(n)\sqrt{2}\psi(2t-n) \end{aligned} \tag{2.21}$$

Now the scaling and wavelet functions $\phi(t), \psi(t)$ are written in terms of low- and high-pass filters, h and g. When the discrete wavelet transform is computed, these filter coefficients h and g are related by downsampling to the approximation (cA) and detail (cD) coefficients of the transform [20]. The process of downsampling (or decimating) in wavelet decomposition involves taking the even terms of the coefficients, which follows directly from our requirement in Equation 2.20. The discrete wavelet transform proceeds by successively decomposing the approximation coefficients into another set of approximation and detail coefficients. The decomposition at level n has the cA_n approximation coefficient and all the other detail coefficients $cD_1, \ldots, cD_n$ up to that level. Figure 2.10 shows the process of decomposing a signal to approximation and detail coefficients. When used in filtering, the details (high frequency components) are removed at one of these iterative steps, resulting in a low-pass filtering operation.

2.3.1 Dynamic Wavelet Fingerprinting (DWFP)

Dynamic Wavelet Fingerprinting (DWFP) is a technique developed by Jidong Hou to use wavelet transforms to create a 2D binary image of the original waveform [1, 21, 22]. It has previously shown promise for a variety of applications, including an ultrasonographic periodontal probe [21, 23–25], detection of ultrasonic echoes in thin

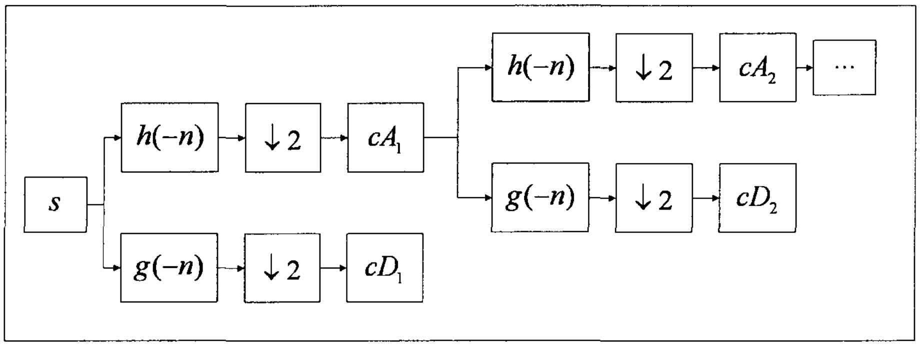


Figure 2.10: The signal s is decomposed into h (low-pass filter) and g (high-pass filter). The downsampling process results in approximation and detail coefficients cA_1 and cD_1 at the first level of decomposition. The next level of decomposition is performed by similarly breaking down the first-level approximation coefficients. This figure is adapted from several diagrams in MATLAB's Fast Wavelet Transform documentation and is similar to Figure 3.3 in [20].

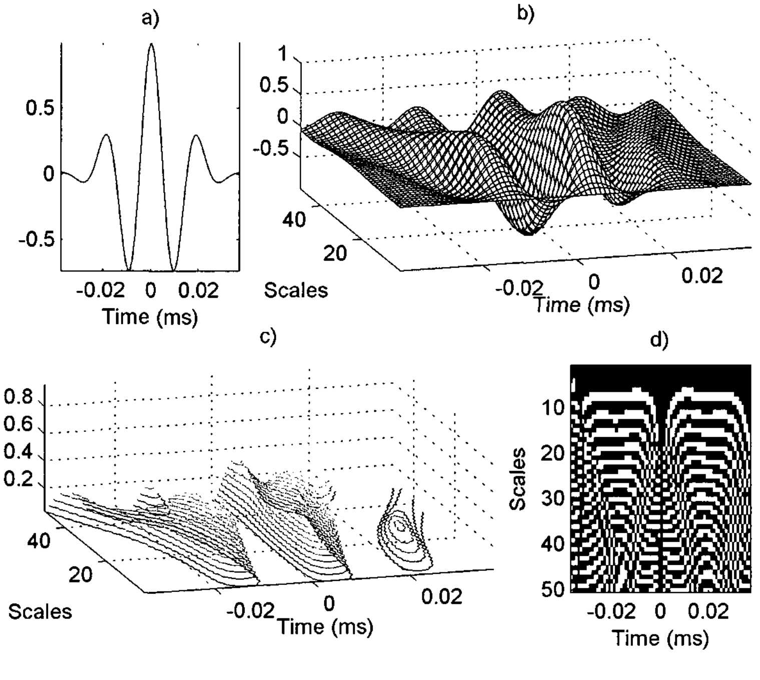


Figure 2.11: The DWFP technique (adapted from [1]) begins with a) the signal, where it generates b) wavelet coefficients indexed by time and scale, where scale is related to frequency. Then c) the coefficients are sliced and projected onto the time-scale plane (d).

multilayered structures [26], robot navigation [27], and structural monitoring with Lamb waves [28–31].

The DWFP process is illustrated in Figure 2.11 which relates to the following steps below:

1. Signal extraction:We begin with the recorded signal, which has usually been sensed and digitized. At this point the signal can be windowed to a region of interest, or it may be entered whole into the algorithm (Fig. 2.11a).

2. CWT: The wavelet coefficients are generated using MATLAB's continuous wavelet transform. Though the procedure is actually digital, MATLAB's continuous wavelet transform uses a range of real numbers for the scale and translation coefficients (a, b), while the discrete wavelet transform uses the dyadic scale $(a = 2^m$, $b = n2^m$, $n, m \in \mathbb{Z}^2)$. A normal range of scales includes $a = 1, \ldots, 50$, and $b = 1, \ldots, N$ for a signal of length N. This yields a 2D array of coefficients, $C(a, b)$, which are normalized to the range of $[-1, 1]$ (Fig. 2.11b).

3. Projection: The coefficients are projected onto the 2D time-scale plane resulting in an image like a contour. To do this in a uniform way, the DWFP algorithm finds the a, b coefficients associated with the desired slice thickness and sets the associated 2D image pixels to "on". Those coefficients not selected in the slice are "off" (Fig. 2.11c). In order to assist with segmentation of the binary image, the "peaks" $(C(a, b) >= 0)$ and "valleys" $(C(a, b) < 0)$ are projected separately.

4. Final image: The result of the slicing algorithm is a 2D binary image related to the scales a and times $b \propto t$ at which the wavelet coefficients were sliced.

The resulting 2D image corresponds to the sliced coefficients $C(a, b)$ on the time-scale plane. The values of the slice thickness and the number of slices made can vary the appearance but it is possible to project the wavelet coefficients in such a way that the image looks like human fingerprints. Thus, the method described above is called Dynamic Wavelet Fingerprinting. In some applications, when several fingerprints are viewed at once in an image, the resulting image is called a "thumbprint," but here we may use the terms interchangeably so that a single fingerprint shape in the image is called a "fingerprint" while several fingerprints in an image is merely pluralized to "fingerprints." Figure 2.12 shows the DWFP transform applied to a simulated signal generated by a 500 Hz cosine wave sampled at 10 kHz with some added 10% white noise. The signal is filtered and the DWFP transform is shown for the whole signal as well as just the peaks and valleys separately.

Filtering is sometimes performed before the DWFP transform. The amount and type of filtering depends on the application from which the signals were gathered. Sometimes wavelet fingerprinting is used, which applies the decomposition described in Figure 2.10, removes some details at the final level, and recomposes the signal by the same process in reverse but with upsampling instead of downsampling.

2.3.2 DWFP Feature Extraction

Image preprocessing

The 2D, discrete, binary image resulting from the DWFP is fairly simple to process. Unlike most image recognition tasks, no thresholding is required since the image is already binary. In the work that follows, we use image recognition techniques to extract features from the DWFP image for statistical pattern recognition[3]. We will describe those features that can be extracted from MATLAB, as well as some original additions related to boundary fitting. In general, the process of feature extraction

[3] We used MATLAB's Image Processing Toolbox (MATLAB, 2008, The Mathworks, Natick, MA).

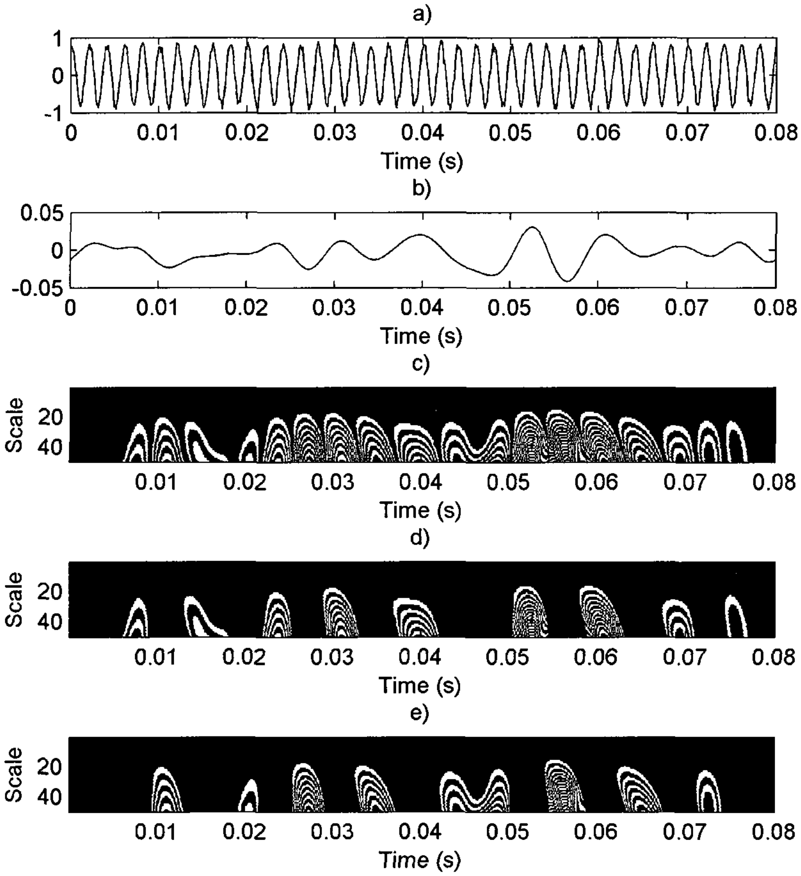


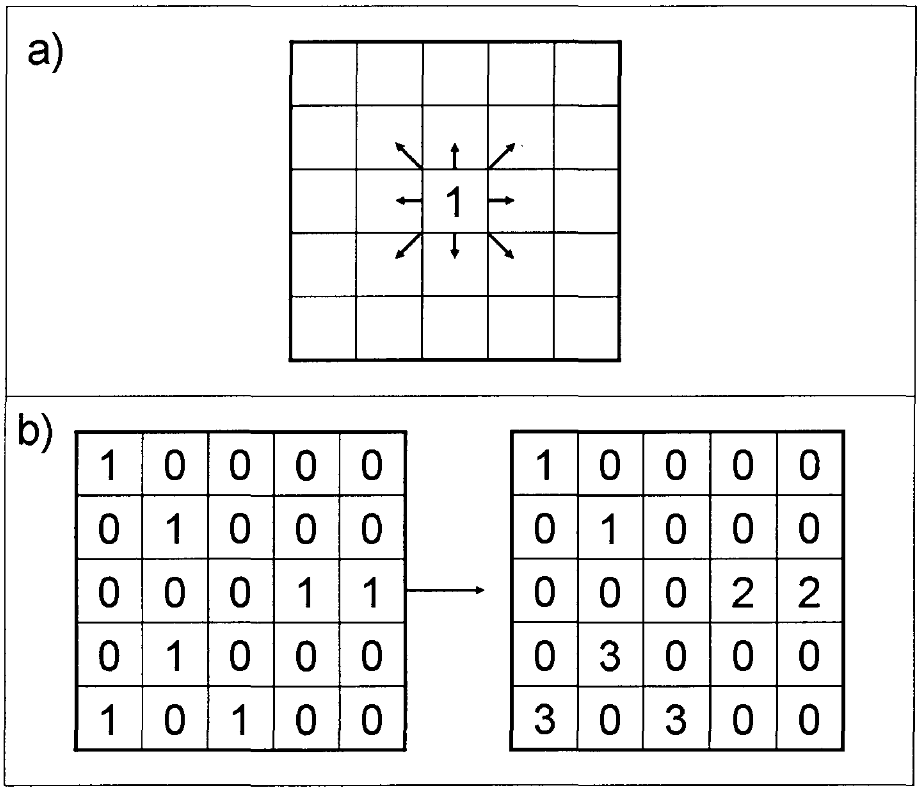
Figure 2.12: The application of DWFP to a) a cosine signal with some added noise and b) after filtering with a stationary wavelet filter. Then c) the resulting DWFP transform which can be performed separately for d) peaks and e) valleys.

includes:

1. Flip the image so its origin is on the bottom left-hand side (MATLAB defaults set image origins at the upper left-hand side)
2. Label the binary image with separate index values for all the "on" pixels that are connected.
3. Adjust some of objects that were labeled separately even though they are actually the same fingerprint.
4. Apply thresholds to select for only important objects, ignoring, for example, those with an area too small to be considered a fingerprint.
5. Extract features from each labeled object using MATLAB.
6. Add features by fitting polynomials to the boundary.

The first step is to recognize each fingerprint as a separate object. This process will take the 2D binary image and assign the same value to pixels that are "connected" to each other. In this application, we use 8-connectivity, which is depicted in Figure 2.13a. Pixels are considered connected if they are adjacent in one of the eight directions shown in Fig. 2.13a. In this way, any "on" pixels in a binary image that share 8-connectivity are labeled with the same pixel value, as shown in Figure 2.13b. This job is made even easier by the DWFP process, since as already mentioned, the DWFP transform is performed separately on the peaks and valleys of the coefficients so that fingerprints going through the labeling process are never touching (Fig. 2.12d-e).

Because of the concentric shapes of the fingerprints, some of these get mislabeled. Figure 2.14a shows an example of a fingerprint from Figure 2.12d. After 8-connected labeling, the fingerprint centered at about 39.5 ms has different labels for the different



$$\begin{array}{|c|c|c|c|c|}\hline 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|c|c|}\hline 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 2 & 2 \\ \hline 0 & 3 & 0 & 0 & 0 \\ \hline 3 & 0 & 3 & 0 & 0 \\ \hline \end{array}$$

Figure 2.13: The sketch demonstrates a) 8-connectivity and b) its application on a binary image, where each square in the matrix represents a pixel in the image.

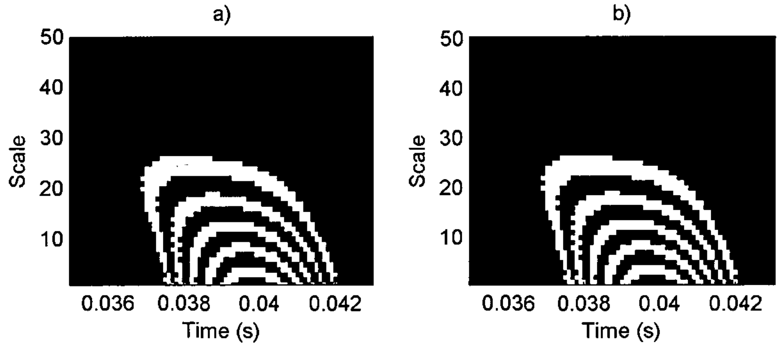


Figure 2.14: a) A fingerprint from Fig. 2.12d that received four different labels for the ridges. Each label has received a different color (pixel value) in the image. b) These connected objects have been re-labeled with the same pixel value.

ridges of the fingerprint so that there were 4 different labels applied to this one fingerprint. An additional layer to the image recognition process is to recognize these concentric labels as a single object. This is done by measuring the center of mass of each object, which is the average t-coordinate of all the pixels with the same label in the image. If the center of mass is less than a given threshold, then all of the concentrically-labeled objects are re-labeled to have the same pixel value (Fig. 2.14b). Since the fingerprints are already segmented from the different image generation for peaks and valleys, a good value for the threshold is the average difference in the center of mass of the objects in the image, which can be manually estimated. In the example in Fig. 2.14a, the center of mass of the four objects occurs at 38.98, 39.42, 39.64, and 39.78 ms respectively. These values differ by about 0.35 ms, which is much smaller than the average difference between the fingerprints in the image (Fig. 2.12d), which is 40 ms.

One last manual adjustment to the fingerprints is to exclude the labeled objects whose area (sum of the pixels) is below a particular threshold. The goal is to exclude objects that are too small and will not be considered a fingerprint. The threshold varies depending on the mother wavelet and can be somewhat arbitrary. As long as the area threshold is consistent, however, the exact value does not matter. A lower bound on the area measurement of between $20-190$ pixels was empirically determined and is often used.

Feature List

The list below provides some of the feature measurements made on the fingerprints that have survived the preprocessing steps above[4]. These measurements are performed on an image matrix $I(a,b)$, where a, b are the same scale and translation parameters used in the DWFP transform over ranges $a = 1, \ldots, n_a$ and $b = 1, \ldots, n_b$. Let ℓ represent the pixel label for the 8-connected objects in the image I and let x, y represent the coordinates For each ℓ, the properties measured include:

1. The coordinates of the pixels in I: Let $\overrightarrow{p}$ represent the list of pixel coordinates so that $p(i,1)$ represents the i^{th} b-coordinate and $p(i,2)$ represents the i^{th} a-coordinate of the pixel with label ℓ in I:

$$\overrightarrow{p} = \{(a,b) \,|\, I(a,b) = \ell\} \tag{2.22}$$

2. The area (or total number of pixels in I with label ℓ)

$$A = \{\#(a,b)|I(a,b) = \ell\} \tag{2.23}$$

[4] These features are copyrighted by MATLAB and can be discussed only in brief.

3. The centroid (or center of mass) (c_b, c_a):

$$\begin{aligned} c_b &= \tfrac{1}{A}\textstyle\sum_{i=1}^{A} p(i,1) \\ c_a &= \tfrac{1}{A}\textstyle\sum_{i=1}^{A} p(i,2) \end{aligned} \tag{2.24}$$

4. The diameter of a circle with the same area as the region [32]:

$$D_c = \sqrt{\frac{4A}{\pi}} \tag{2.25}$$

5. Extent, which is the number of total "on" pixels divided by the total pixels surrounded by the region

$$E_x = \frac{A}{[\min_i p(i,1) - \max_i p(i,2)] \cdot \max_i p(i,2)} \tag{2.26}$$

6. Filled area, which is the number of "on" pixels in the region where the holes have been filled in. The image with the holes filled in is called the filled image and is calculated by removing all off pixels that are not connected to the image border [33, 34].

7. The Euler number is a topological attribute [35]. Using the concept of neighborhoods and connectivity, the Euler number is the number of objects in an image minus the number of holes in the object. For 8-connectivity, the Euler number for a binary image is calculated by

$$E = \frac{1}{4}\left(n\{Q_1\} - n\{Q_3\} - 2n\{Q_D\}\right) \tag{2.27}$$

In Equation 2.27, the Q_i are 2×2 sections of the image called bit quads. Here, Q_1 is a bit quad with a single 1 and three 0's, while Q_3 is similarly a bit quad with three 1's and a single 0. Lastly, Q_D is a bit quad with diagonal 1's. The $n\{Q_i\}$ represents the number of bit quads of that type in the image.

8. The convex image is produced by finding the smallest convex polygon that can contain the "on" pixels with value ℓ. From this image, the convex area (A_C) is calculated, which is the total number of pixels in the convex image.

9. The solidity, which is calculated by

$$s = \frac{A}{A_C} \tag{2.28}$$

Some of the properties computed by MATLAB's image processing routines require measuring the second central moments of the fingerprint. The formula for central moments is given by [36]

$$\mu_{p,q} = \sum_x \sum_y (x - \overrightarrow{x})^p (y - \overrightarrow{y})^p f(x, y) \tag{2.29}$$

for a digital image $f(x, y)$ at centroid location $(\overrightarrow{x}, \overrightarrow{y})$. Therefore, the normalized second central moment has three components:

$$\begin{aligned} \mu_{2,0} &= \mu_{2,0}/\mu_{0,0} = M_{2,0}/M_{0,0} - \overrightarrow{x}^2 \\ \mu_{0,2} &= \mu_{0,0}/\mu_{0,0} = M_{0,2}/M_{0,0} - \overrightarrow{y}^2 \\ \mu_{1,1} &= \mu_{1,1}/\mu_{0,0} = M_{1,1}/M_{0,0} - \overrightarrow{x}\,\overrightarrow{y} \end{aligned} \tag{2.30}$$

where $M_{i,j}$ is the raw moment given by

$$M_{i,j} = \sum_x \sum_y x^i y^j I(x, y) \tag{2.31}$$

for pixel intensity $I(x,y)$. Using these equations, we get the three components for the normalized second central moment for $I(x,y)=1$ as

$$\begin{aligned} \mu_{2,0} &= \tfrac{1}{A}\textstyle\sum_{i=1}^{A}(p(i,1)-c_b)^2+\tfrac{1}{12} \\ \mu_{0,2} &= \tfrac{1}{A}\textstyle\sum_{i=1}^{A}(p(i,2)-c_a)^2+\tfrac{1}{12} \\ \mu_{1,1} &= \tfrac{1}{A}\textstyle\sum_{i=1}^{A}(p(i,1)-c_b)(p(i,2)-c_a) \end{aligned} \tag{2.32}$$

where we have used the fact that 1/12 is the normalized second central moment of a pixel with unit length.

From the object's normalized second central moments we can find properties of an ellipse that has the same normalized second central moments[5]. These properties include major axis length, minor axis length, eccentricity, and orientation.

1. The major axis length (Equation A.78 in [37]), which is given by

$$x_{\text{maj}} = 2\sqrt{2}\sqrt{\mu_{2,0}+\mu_{0,2}+\gamma} \tag{2.33}$$

   where $\gamma = \sqrt{(\mu_{2,0}-\mu_{0,2})^2+4\mu_{1,1}^2}$

2. The minor axis length (Equation A.80 in [37]), which is similarly

$$x_{\text{min}} = 2\sqrt{2}\sqrt{\mu_{2,0}+\mu_{0,2}-\gamma} \tag{2.34}$$

3. The eccentricity (e_{cc})

$$e_{cc} = \frac{2\sqrt{\left(\frac{x_{\text{maj}}}{2}\right)^2-\left(\frac{x_{\text{min}}}{2}\right)^2}}{x_{\text{maj}}} \tag{2.35}$$

[5] See Appendix A of [37].

4. The orientation (Equations A.81-A.82 in [37]) is calculated by

$$\begin{aligned} \theta_o &= \arctan\left(\frac{\mu_{0,2}-\mu_{2,0}+\gamma}{2\mu_{1,1}}\right), \quad \mu_{0,2} > \mu_{2,0} \\ \theta_o &= \arctan\left(\frac{2\mu_{1,1}}{\mu_{0,2}-\mu_{2,0}+\gamma}\right), \quad \mu_{2,0} \geq \mu_{0,2} \end{aligned} \tag{2.36}$$

An additional property was added from the algorithm for the orientation of a 2D object [38]. This algorithm uses the same second central moments computed in Equation 2.32 but uses trigonometry to compute the angle of the object (θ_n):

$$\tan(2\theta_n) = \frac{\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}$$

It is actually calculated by finding the positive solution of

$$\begin{aligned} \sin(2\theta_n) &= \frac{\mu_{1,1}}{\sqrt{\mu_{1,1}^2+(\mu_{2,0}-\mu_{0,2})^2}} \\ \cos(2\theta_n) &= \frac{\mu_{2,0}-\mu_{0,2}}{\sqrt{\mu_{1,1}^2+(\mu_{2,0}-\mu_{0,2})^2}} \end{aligned} \tag{2.37}$$

In order to differentiate this property from the ellipse's orientation, we call it inclination. Usually orientation (θ_o) and inclination (θ_n) give very similar measurements.

Several additional properties were measured from the fingerprints by fitting polynomials to the boundaries. From early observations of the wavelet fingerprint shapes from different mother wavelets, it was observed that the shape of the fingerprints changes slightly over time. Therefore, in order to describe the shapes of the boundaries, an algorithm was designed to extract the fingerprint-shaped boundary and fit $n = 2$ and $n = 4$ polynomials. The properties used are the coefficients of each type of polynomial fit. These properties could not be generated in general for any image, such as the above list, because the algorithm is unique to the fingerprint shape from the DWFP. Figure 2.15 shows an example of the boundary extraction and curve fitting.

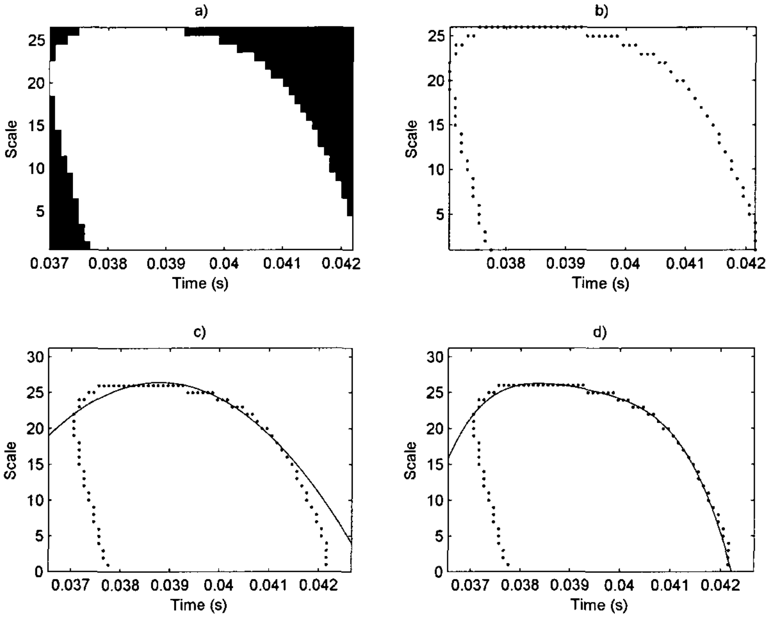


Figure 2.15: a) The convex image of the fingerprint from Fig. 2.14b and b) its boundary, c) the quadratic fit to the boundary, and d) the $n = 4$ fit to the boundary. Note that the polynomials do not perfectly describe the boundary of the fingerprints in either case but remain a consistent method of measuring the boundary shape.

The process includes:

1. The convex image is extracted using the same algorithm as above that calculated the convex area [33]
2. The convex image is padded with zeros on all sides
3. The algorithm records the position of the pixels that change from "off" to "on" as the boundary of the fingerprint starting from each of the four sides of the image
4. The boundary positions are sorted to trace the boundary of the fingerprint
5. These sorted boundary positions are used to fit $n = 2$ and $n = 4$ polynomials using the linear least squares method
6. The coefficients of these two polynomials are returned as additional features

These three groups of features - basic pixel relations (Equations 1-2.28), ellipse measurements (Equations 2.36-2.37), and the coefficients of polynomial fits to the boundary - are all collected from the DWFP binary image and used as features. Often different mother wavelets are used to create the DWFP images, and these features are extracted for each mother wavelet used. Feature selection varies by application but often involves finding a distance measure between the extracted features $\mathbf{x}$ to find the most robust discriminative features between classes. This will be discussed in more detail in each of the application chapters. Formal feature selection techniques, such as Principal Component analysis, were not used here because of the large size of the data sets leading to long computation times.

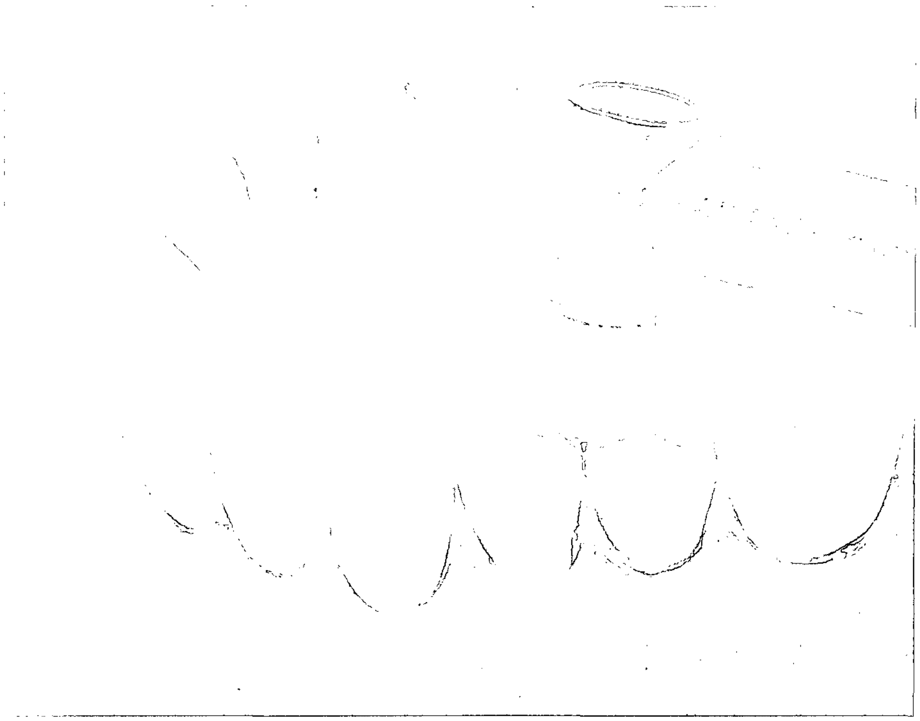


Figure 2.16: A photo of the ultrasonic handpiece, delay line, and phantom tooth model.

2.3.3 An Example: Ultrasonographic Detection of Tooth Flaws

Here we will offer an experiment that illustrates the usefulness of the DWFP feature extraction process on time-domain signals. The example is an application of dental ultrasonography. Although ultrasonography of dental tissues was first demonstrated in the early 1960's, it has not yet become a diagnostic tool in the clinical practice of oral health management [39]. This is in sharp contradistinction to every other part of the body where diagnostic ultrasonography is ubiquitous. When appropriate, it is the favored diagnostic tool because it is safe, portable, real-time, and perhaps most importantly, it is inexpensive and easy to use. It is also readily adaptable to specialized diagnostic applications, since interchangeable handpieces can be used with the same basic instrumentation. It seems clear that diagnostic ultrasonography is coming to clinical dentistry eventually, but it's less clear which set of applications will be the ones that lead the way to wide-scale adoption of the modality. In this work we are focusing on adapting ultrasonography to detect a variety of defects of interest in teeth, including those for which x-rays are not well suited, but also with an eye towards direct substitution for x-rays in some cases. The non-ionizing nature of ultrasonography holds the potential for reducing the radiation dose to patients whenever it can be used instead, which will have a positive impact on the health and safety of both dental patients and practitioners.

In this study, we will use surface wave modes, which are highly sensitive to small surface and near surface flaws. In addition, the surface waves can propagate down between teeth following the curvature of the tooth, so they can interrogate interproximal sites while the transducer remains in more accessible locations on the occlusal (biting) surface of the tooth. The goal of this preliminary study is to demonstrate the ability to detect flaws in human and phantom (model) teeth using a freehand ultrasonic handpiece intended to be compatible with clinical applications. All ultrasonographic data were acquired using the commercially-available Sonopen handpiece operating at 10 MHz, shown in Figure 2.16, which uses a hard plastic tapered delay line to couple ultrasound into and back out of the tooth. In order to assist coupling to the hard tooth surfaces, the delay line was coated with a single layer of latex mold. Future tests will require optimizing the material and shape of the delay line, since reliable coupling is a key challenge necessary for making dental ultrasonography clinically useful. The instrument itself is operated by foot pedal and is fully automated by computer. However, as the results will show, the ultrasonic reflections from the handpiece are subtle and features due to flaws are difficult to distinguish by eye alone. Therefore, we will use the Dynamic Wavelet Fingerprinting technique to generate 2D binary images from the RF waveforms in order to enhance the possibility of feature detection.

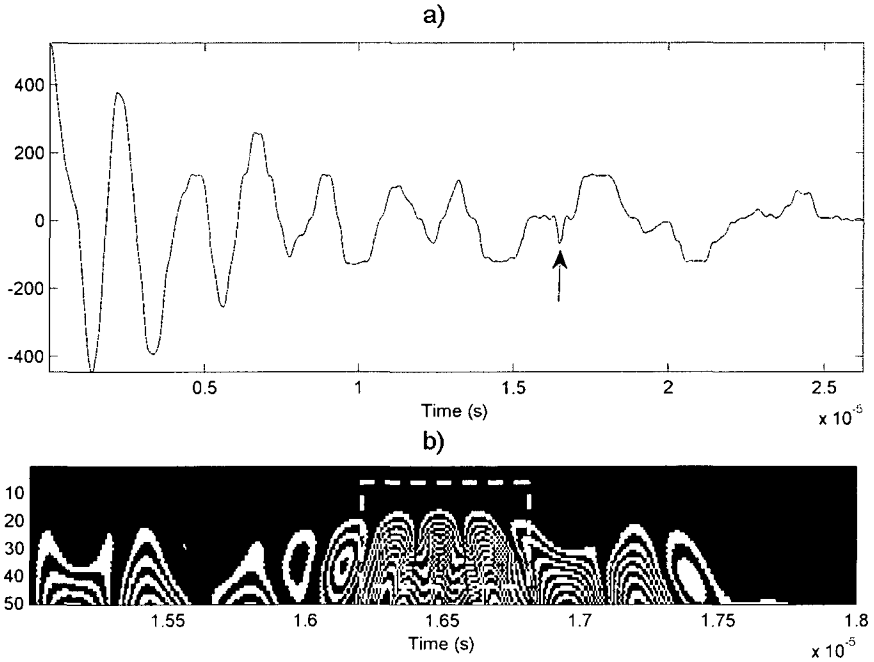
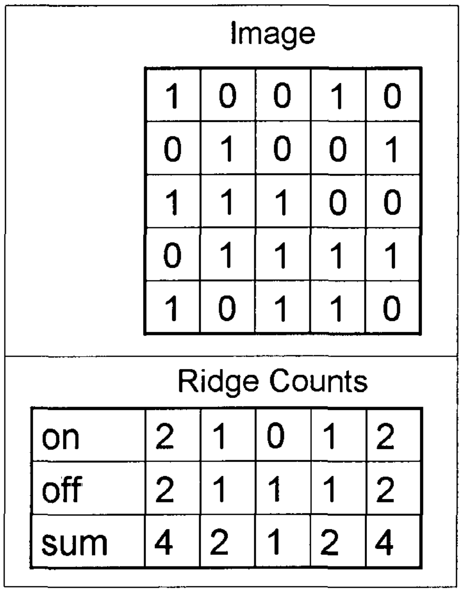


Figure 2.17: a) An example of a surface wave generated from an aluminum plate and b) the associated DWFP with the region of interest indicated by a box

The example in Figure 2.17 demonstrates the generation of a surface wave mode in an aluminum plate. Surface wave reflection from a small pit is indicated with an arrow, which could clearly be seen to move as the wedge was moved in relation to the flaw. The associated wavelet fingerprint is centered on that surface wave reflection and highlights the unique characteristic features that can be detected with the DWFP's time-and-frequency analysis method.

The feature extracted from the DWFP images used in this study is a measure called *ridge counts*. The general idea is to count the number of arches in the fingerprint-like shapes from the DWFP image. In actuality, the process involves counting the number of times pixels change on or off for each time position in the image. Figure 2.18 shows the process of extracting the ridge count feature from a binary image. The number of times pixels change from 0 to 1 is recorded as the number of "on" pixel in a column (along the scale-dimension), and likewise the number of times pixels change from 1 to 0 is recorded as the number of "off" pixels. The sum of these measures is used as the ridge count feature and is usually smoothed to compare ridge counts between different images. Figure 2.19 shows an example of the ridge count measure from an actual DWFP image. Note that the ridge count feature is smaller in the center of the fingerprint (where there are not many ridges) and larger in the arches of the fingerprint, as desired.

Next, the handpiece was used to collect ultrasonic RF echoes from phantom teeth selected from the tooth model shown in Fig. 2.16. Two phantom teeth of the same shape were selected for testing. A flaw was introduced in one of the teeth, and the occlusal surface of both teeth was probed at nine different sites along the same direction using the handpiece with medical ultrasound gel for additional coupling. The waveforms over the tooth sites were averaged in order to eliminate transient influences of noise and hand stability. The resulting comparison (Fig. 2.20) shows a subtle but



Image

1	0	0	1	0
0	1	0	0	1
1	1	1	0	0
0	1	1	1	1
1	0	1	1	0

Ridge Counts

on	2	1	0	1	2
off	2	1	1	1	2
sum	4	2	1	2	4

Figure 2.18: The diagram explains ridge count feature extraction. Note that the starting pixel value is not included in the ridge count.

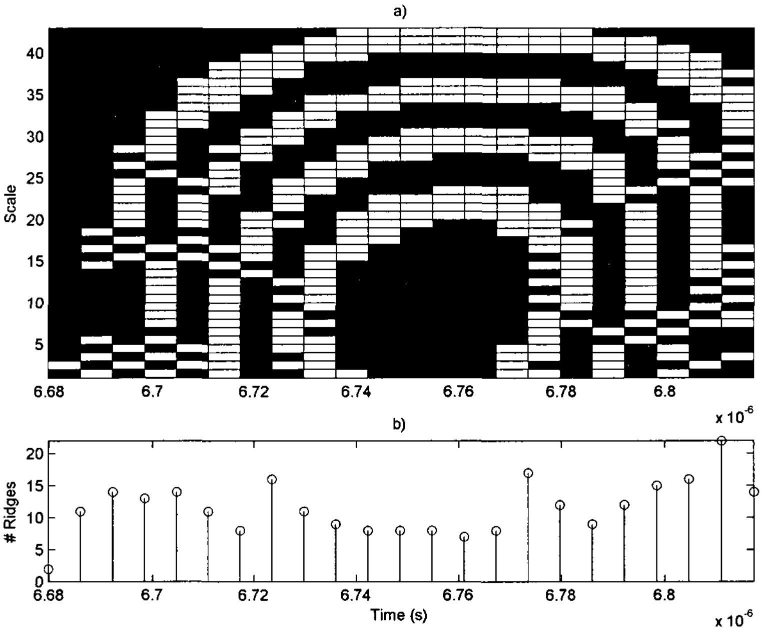


Figure 2.19: a) An actual DWFP image is shown with b) the number of ridge counts over time using the ridge count process in Figure 2.18.

clear difference in the RF echo between the flawed and unflawed phantom teeth. The wavelet fingerprint comparisons of a single waveform from each phantom tooth are shown in Figure 2.21. The ridge count feature was calculated, twice smoothed, and plotted below the fingerprints. The region of interest marked by the box demonstrates a measured difference between the flawed and unflawed samples.

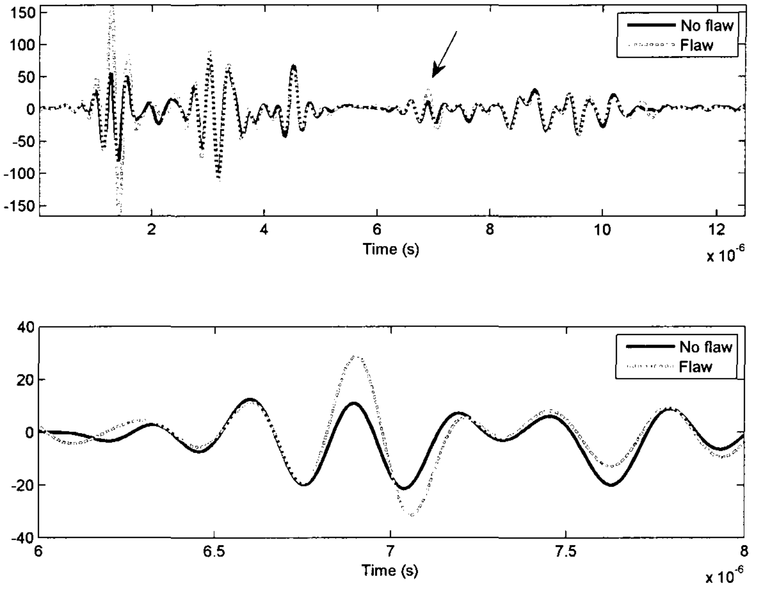


Figure 2.20: A comparison of flawed and unflawed phantom tooth waveforms followed by a close-up of the region of interest.

The same process can be performed for the human cadaver teeth. The waveform comparison in Figure 2.22 was taken from flawed and unflawed sites of the same human tooth, with the arrows indicating features of interest. Note that the human cadaver RF echoes are less smooth than those from the phantom.

A further example is displayed in Figure 2.23, again verifying the detection capabilities of the ultrasonographic handpiece in conjunction with the wavelet fingerprinting feature extraction method on a human tooth. At top left are waveforms gathered

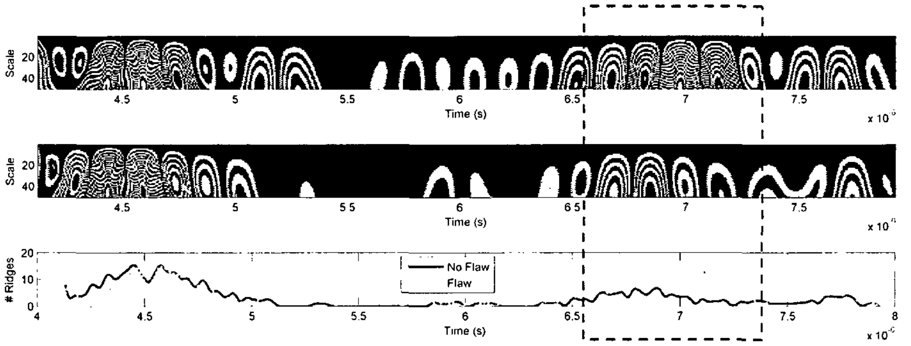


Figure 2.21: A comparison of the DWFP transform of flawed (top) and unflawed phantom (middle) tooth waveforms as well as the extracted feature, ridge counts.

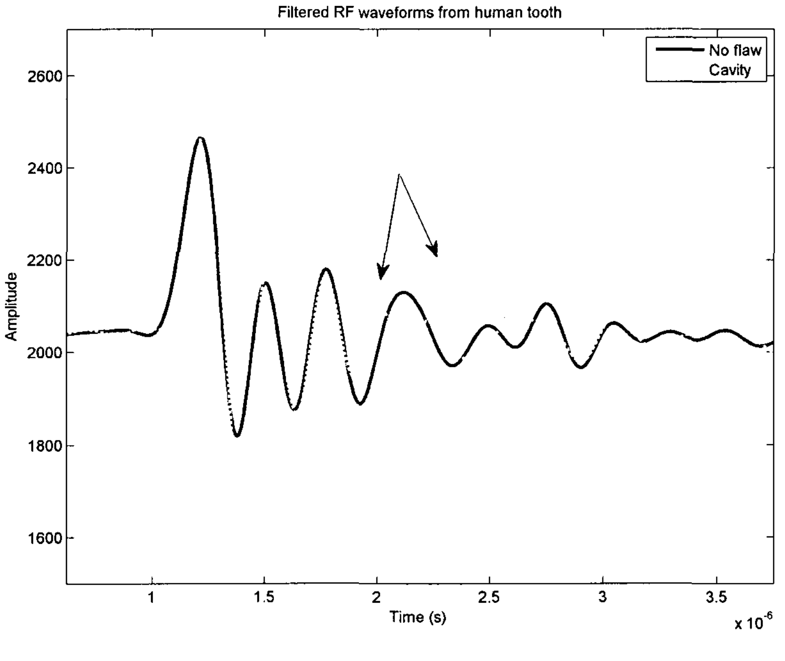


Figure 2.22: A comparison of flawed and unflawed waveforms gathered by hand from a human cadaver tooth. The region of interest is indicated with arrows.

with the handpiece from the tooth with an artificial flaw as shown in an in vitro X-ray CT slice (Fig. 2.23e). Waveforms from flawed and unflawed portions of the tooth are shown in Fig. 2.23a, and corresponding DWFP transforms are shown in Fig. 2.23b and 2.23c from those respective waveforms. The plot in Fig. 2.23d shows the ridge counts feature plotted versus depth for flawed and unflawed tooth sites, which provides a metric to discern the presence of flaws in the echoes. The box indicates the region of interest between the corresponding sections of DWFP fingerprints and the ridge count feature. This figure shows that the ridge count feature distinguishes between the flawed and unflawed section of a human cadaver tooth using a handheld probe. The location of the flaw can be inferred by the tooth site at which the ultrasonographic waveforms were recorded.

In this example, ultrasonographic data from phantom and human cadaver teeth was gathered freehand using a simple ultrasonic handpiece connected to a portable laptop computer. The teeth were probed at different sites on the occlusal surface. In combination with the DWFP and ridge count properties, we have been able to demonstrate clear distinctions between flawed and unflawed tooth sites. The DWFP algorithm highlights subtle differences in the waveform that are not otherwise detectable. The DWFP technique allows for specific features to be identified in waveforms which correspond to specific identifiable interactions of the surface waves with specific flaws in teeth.

2.3.4 Summary

This chapter has described statistical pattern classification as well as the generation of the DWFP image and the features extracted from it. The example described at the end used the DWFP algorithm and the ridge count feature extraction to distinguish between healthy and damaged tooth samples taken from a phantom model

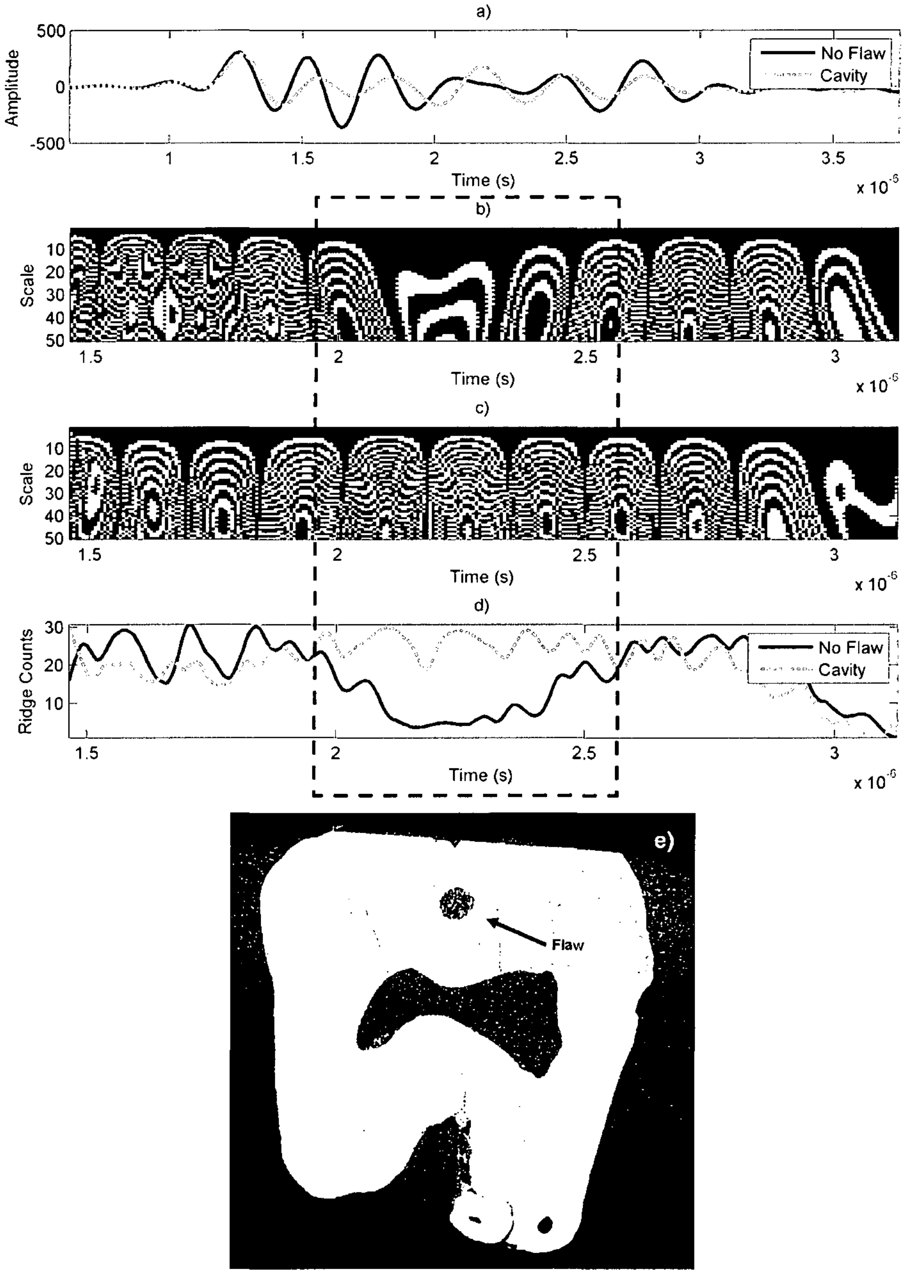


Figure 2.23: The human cadaver tooth shown in section e) was scanned with a handheld probe. The plots include a) comparisons of waveforms from flawed and unflawed sections of the tooth, as well as DWFP fingerprints of the b) unflawed and c) flawed RF echoes. The number of ridges in the DWFP transform is used as a metric to distinguish between the flawed and unflawed RF echoes (d), as indicated by the region of interest inside the box.

and a human cadaver. This same method will be used in the following chapters to classify unique events or flaws in fields that use time domain signals, though more sophisticated feature extraction, feature selection, and pattern classification will be required in these larger domains.

Bibliography

[1] J. Hou, "Ultrasonic signal detection and characterization using dynamic wavelet fingerprints," Ph.D. dissertation, Department of Applied Science, College of William and Mary, Williamsburg, VA, 2004.

[2] A. Asuncion and D. Newman, "UCI machine learning repository," 2007.

[3] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188.

[4] R. H. Mohlenbrock, "Northeast wetland flora: Field office guide to plant species," Northeast National Technical Center, Chester, Tech. Rep., 1995.

[5] U. S. D. of Agriculture Natural Resources Conservation Service, "The plants database," http://plants.usda.gov, National Plant Data Center, Baton Rouge, LA 70874-4490 USA, August 2009.

[6] A. R. Webb, *Statistical Pattern Recognition*, 2^{nd} ed. Hoboken, NJ: Wiley, 2002.

[7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2^{nd} ed. New York: Wiley, 2001.

[8] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses.* New York: Springer-Verlag, 2003.

[9] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines.* New York: Cambridge University Press, 2000.

[10] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[11] L. I. Kuncheva, *Combining pattern classifiers methods.* New York: Wiley, 2004.

[12] D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceedings of the* 11^{th} *International Conference on Machine Learning.* New Brunswick, NJ: Morgan Kaufman, 1994, pp. 148–156.

[13] N. Japkowicz and S. Stephen, "The class imbalance problem: a systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[15] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in *Proceedings of the* $14^t h$ *International Conference on Machine Learning.* Morgan Kaufmann Publishers, Inc, 1997, pp. 179–186.

[16] G. M. Weiss and F. Provost, "Effect of class distribution on classifier learning: An empirical study," Rutgers University Department of Computer Science, Tech. Rep. ML-TR-44, August 2001.

[17] L. Debnath, *Wavelet Transforms and Their Applications.* Boston: Birkhäuser, 2002.

[18] B. B. Hubbard, *The World According to Wavelets.* Wellesley, Massachusetts: A K Peters, 1996.

[19] I. Daubechies, *Ten Lectures on Wavelets.* Philadelphia: Society for Industrial and Applied Mathematics, 1992.

[20] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms.* Upper Saddle River, NJ: Prentice Hall, 1998.

[21] J. Hou and M. K. Hinders, "Dynamic Wavelet Fingerprint identification of ultrasound signals," *Materials Evaluation*, vol. 60, no. 9, pp. 1089–1093, 2002.

[22] J. R. Hou, S. T. Rose, and M. K. Hinders, "Ultrasonic periodontal probing based on the dynamic wavelet fingerprint," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 1137–1146, 2005.

[23] M. K. Hinders and J. R. Hou, "Ultrasonic periodontal probing based on the dynamic wavelet fingerprint," in 31^{st} *Review of Progress in Quantitative Nondestructive Evaluation 24b*, D. O. Thompson and D. E. Chimenti, Eds. Melville, New York: AIP Conference Proceedings, 2005, pp. 1549–1556.

[24] J. E. Lynch, M. K. Hinders, and G. B. McCombs, "Clinical comparison of an ultrasonographic periodontal probe to manual and controlled-force probing," *Measurement*, vol. 39, no. 5, pp. 429–439, 2006.

[25] M. K. Hinders and G. B. McCombs, "The potential of the ultrasonic probe," *Dimensions of Dental Hygiene*, vol. 4, no. 4, pp. 16–18, 2006.

[26] M. K. Hinders, Jidong, Hou, and J. C. P. McKeon, "Ultrasonic inspection of thin multilayers," in 31^{st} *Review of Progress in Quantitative Nondestructive Evalua-*

tion 24b, D. O. Thompson and D. E. Chimenti, Eds. AIP Conference Proceedings, 2005, vol. 760, pp. 1137 – 1144.

[27] W. Fehlman and M. K. Hinders, *Mobile Robot Navigation with Intelligent Infrared Image Interpretation*, ser. Springer Tracts in Advanced Robotics. New York: Springer, 2009.

[28] J. Hou, K. R. Leonard, and M. K. Hinders, "Automatic multi-mode lamb wave arrival time extraction for improved tomographic reconstruction," *Inverse Problems*, vol. 20, pp. 1873–1888, 2004.

[29] J. P. Bingham and M. K. Hinders, "Lamb wave characterization of corrosion-thinning in aircraft stringers: experiment and 3D simulation," *Journal of the Acoustical Society of America*, vol. 126, pp. 103–113, 2009.

[30] J. P. Bingham, M. K. Hinders, and A. Friedman, "Lamb wave detection of limpet mines on ship hulls," *Ultrasonics*, vol. *in press*, 2009.

[31] J. P. Bingham and M. K. Hinders, "Lamb wave detection of delaminations in large diameter pipe coatings," *Open Acoustics*, vol. *in press*, 2009.

[32] J. C. Russ, *The Image Processing Handbook*, 2nd ed. Ann Arbor, MI: CRC Press, 1995.

[33] P. Soille, *Morphological Image Analysis: Principles and Applications.* New York: Springer-Verlag, 1999.

[34] P. Soille and C. Gratin, "An efficient algorithm for drainage networks extraction on DEMS," *Journal of Visual Communication and Image Representation*, vol. 5, no. 2, pp. 181–189, 1994.

[35] W. Pratt, *Digital Image Processing*, 2^{nd} ed. Hoboken, NJ: John Wiley and Sons, 1991.

[36] M. Hu, "Visual pattern recognition by moment invariants," *IEEE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[37] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision.* Boston, MA: Addison-Wesley, 1992, vol. I.

[38] B. K. P. Horn, *Robot Vision.* Cambridge, MA: MIT Press, 1986.

[39] S. R. Ghorayeb, C. A. Bertoncini, and M. K. Hinders, "Ultrasonography in dentistry," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 6, pp. 1256–1266, 2008.

Chapter 3

An Ultrasonographic Periodontal Probe

Periodontal disease, commonly known as gum disease, affects millions of people. The current method of detecting periodontal pocket depth is painful, invasive, and inaccurate. As an alternative to manual probing, an ultrasonographic periodontal probe is being developed to use ultrasound echo waveforms to measure periodontal pocket depth, which is the main measure of periodontal disease. Wavelet transforms and pattern classification techniques are used in artificial intelligence routines that can automatically detect pocket depth. The main pattern classification technique used here, called a binary classification algorithm, compares test objects with only two possible pocket depth measurements at a time and relies on dimensionality reduction for the final determination. The method correctly identifies up to 90% of the ultrasonographic probe measurements within the manual probe's tolerance.

3.1 Introduction

In the clinical practice of dentistry, radiography is routinely used to detect structural defects such as cavities. However, ionizing radiation is harmful to the patient, and has been shown to eventually lead to cavities via the demineralization of teeth. Radiography can also only detect defects parallel to the projection path, so cracks are difficult to detect, and it is useless for identifying conditions such as early stage gum disease because soft tissues are transparent to x-rays. Medical ultrasound, however, is safe to use as often as indicated, and computer interpretation software can make disease detection automatic. The structure of soft tissues can be analyzed effectively with ultrasound, and even symptoms such as inflammation can be registered with this technology [1].

One application of ultrasound in dentistry, a periodontal probe, is being developed as a spin-off of NASA technology [2–10]. Periodontal disease is caused by bacterial infections in plaque, and the advanced stages can cause tooth loss when the periodontal ligament holding the tooth in place erodes [11]. Periodontal disease is so widespread worldwide that $10 - 15\%$ of adults have advanced stages of the disease with deep periodontal pockets that put them at risk of tooth loss [12]. The usual method of detection is with a thin metal probe scribed with gradations marking depth in millimeters (Figure 3.1a). The dental hygienist inserts the probe down into the area between the tooth and gum to estimate the depth to the periodontal ligament. At best, this method is accurate to $\pm$ 1 mm and depends upon the force the hygienist uses to push the probe into the periodontal pocket. Furthermore, this method is somewhat painful and often causes bleeding [13]. The ultrasonographic periodontal probe uses high-frequency ultrasound to find the depth of the periodontal ligament non- invasively. An ultrasonic transducer projects high frequency ($10 - 15$ MHz) ultrasonic energy in between the tooth and the gum and detects echoes of the returning wave (Figure 3.1b). In the usual practice of ultrasonography[1], the time delay of the reflection is converted to a distance measurement by using the speed of sound in water (1482 m/s). However, both experimental and simulation waveforms show that the echoes from the anatomy of interest are smaller than the surrounding reflections and noise. Further mathematical techniques, including a wavelet transform and pattern classification techniques, are required in order to identify pocket depth from these ultrasound waveforms.

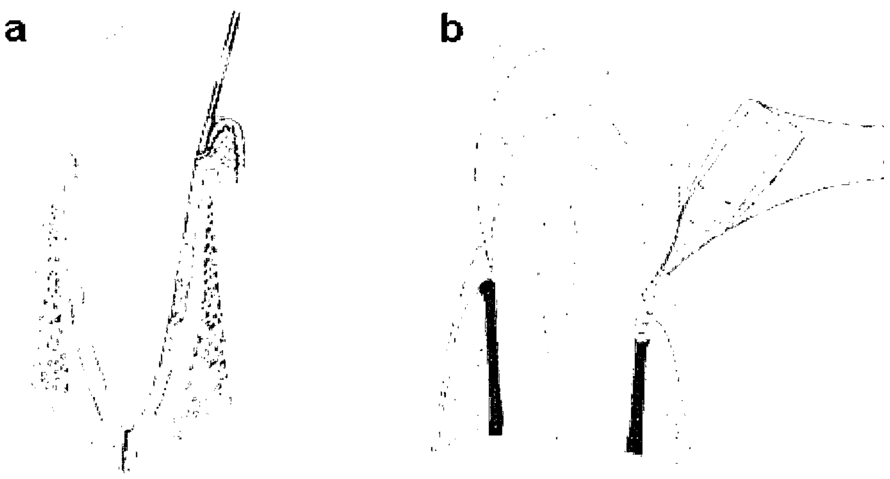


Figure 3.1: The figure compares a) manual periodontal probing, in which a thin metal probe is inserted in between the tooth and gums, versus b) the ultrasonographic periodontal probe, in which ultrasound energy propagates through water noninvasively

The following sections describe the development of artificial intelligence algorithms for an ultrasonographic periodontal probe. Section 3.2 describes past research involving the development of diagnostic ultrasound tools for the detection of soft tissue dental diseases. Section 3.3 describes 3D elastodynamic simulations that were tested

[1] In the medical field, ultrasound (ultrasonic) and ultrasonography (ultrasonographic) are used interchangeably to describe MHz-frequency acoustic waves being used for diagnostic applications. However, in dentistry, ultrasound or ultrasonic refers to kHz-frequency vibrating tools (scalers) used for cleaning, which is a different application of the same technology. Here, we prefer to use "sonographic" or "ultrasonographic" to imply the diagnostic application of ultrasound energy.

to determine that energy was reaching the anatomy of interest. Section 3.4 describes the materials and methods involving gathering the ultrasound data. Sections 3.5 and 3.6 describe the feature extraction and selection procedures in preparation for a binary classification algorithm described in Section 3.7. Section 3.7.2 describes dimensionality reduction, and Section 3.7.3 explores classifier combination. The results are presented in Section 3.8 and statistically analyzed in Section 3.9, with conclusions presented in Section 3.10.

3.2 Related Work

Ultrasound has promise for study and diagnosis of a variety of dental diseases and researchers have been studying the application of diagnostic ultrasound to the dental anatomy for more than 40 years [1]. Much of the work has focused on detecting diseases in the hard tissue of the tooth, especially early experiments, which involved applying standard medical ultrasound devices to teeth to investigate the internal structure by simple peak-detection techniques [14, 15]. Early measurements of the soft tissues of teeth focused on ultrasonic measurement of gingival thickness [16], while the first reported use of ultrasound to diagnose periodontal disease was performed by Spranger [17]. Palou *et al.* [18] attempted to measure bone loss due to periodontal disease using a clinical opthalmological ultrasound instrument, which was the smallest available probe at the time. The probe was inaccurate and difficult to use, and the authors suggested redesigning the probe with a fine tip and a 90° bend to make an ultrasonic probe more similar to current contra-angle dental handpieces. Meanwhile, Löst *et al.* [19–21] attempted to image the periodontal ligament using ultrasound directed perpendicular to the tooth surface but found the resulting 1D waveforms difficult to interpret. Keller [22] concluded that 20 MHz B-scan ultrasound can clearly image periodontal anatomy, but quantitative interpretation of the image was impossible without the ability to interpret the 1D A-scan waveforms. Ohshima [23] applied ultrasonic imaging equipment to periodontal diagnosis but found difficulties standardizing the examination procedure. Oikarinen *et al.* [24] compared plain radiographs, computed tomography, magnetic resonance imaging, and ultrasound for visualizing foreign bodies in soft tissues and found that the ultrasound resulted in the best sensitivity and specificity over the other modalities.

As technological advances and computing power increased at the end of the 20^{th} century, research in soft tissue dental ultrasound became more successful. This research included measurements of gingival thickness [25, 26], notably in which Eger *et al.* [25] used a handheld ultrasonic thickness measuring device to automatically determine gingival thickness with excellent validity and reproducibility. Tsiolis et. al [27] similarly used a commercially-available 20 MHz ultrasound scanner to demonstrate that ultrasonography can produce images suitable for qualitative assessment of the periodontum. Periodontology research during this time also includes prototypes of ultrasonic tools. In 1998 Loker and Hagenbuch [28] constructed a prototype ultrasonic periodontal diagnostic instrumentation system where the ultrasound was instead intended to couple to the tissues via a tapered delay line, with the area of the tip designed so that the tip would make good contact with the gum surface in approximately the same area and orientation in which mechanical probing is traditionally done [29].

The predecessor to the ultrasonographic probe in this study was first reported by Companion and Hinders [2, 3], which had been under development at NASA Langley Research Center for several years [5]. This probe has a hollow tapered tip that is flooded with water for coupling of the ultrasonic beam into the tissues, with the very small probe tip in contact with the gum line in the same orientation as traditional manual probing. As generations of this ultrasonographic periodontal probe proceeded, the internal shape of the hollow probe tip was optimized via a combination of computer simulations and systematic experiments, and a sequence of increasingly more practical clinical prototypes were developed and used in several pilot studies comparing ultrasonic to calibrated-manual and controlled-force probing [8, 9, 30, 31].

A critical development was the recognition of the need for narrowly-tailored artificial intelligence algorithms to automatically identify the very subtle echo-waveform features corresponding to the anatomy of interest. Hou developed the Dynamic Wavelet Fingerprint (DWFP) method to transform wavelet coefficients to 2D binary images [10, 32]. The method is general, but when directly applied to data obtained from a 4th generation ultrasonographic periodontal probe tested on 14 patients [33, 34], the authors were able to resolve at best around 75% of the pocket depths accurately within a tolerance of 1.5 mm. When the tolerance is adjusted to the 1 mm precision of the manual probe, the highest success rate per patient drops to about 60%. The authors did not use pattern classification but instead relied on image recognition techniques to detect pocket depth. The aim of this chapter is to improve the detection capabilities of a newer prototype ultrasonographic periodontal probe by applying pattern classification techniques.

3.3 Simulation

Before manufacturing the newest prototype ultrasonographic periodontal probe, 3D simulations were performed to model the proposed probe and the anatomy of interest to determine the acoustic propagation of the ultrasonic energy [35, 36]. These simulations were performed on William and Mary's scientific computing cluster using a 3D elastodynamic finite integration technique. The cross section of the anatomy

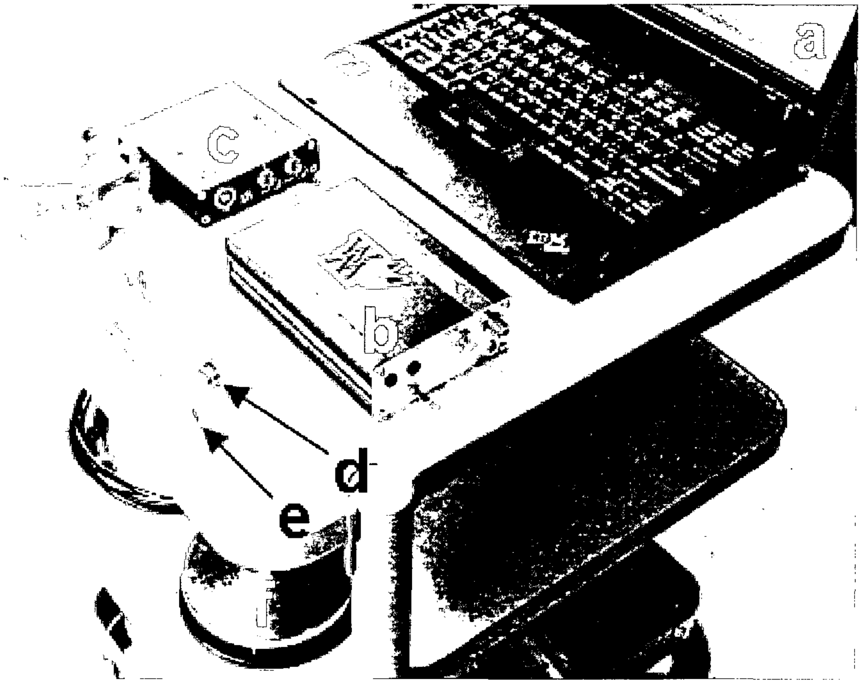


Figure 3.2: All devices necessary to conduct experiments using the ultrasonographic periodontal probe are shown, including a) labtop computer, b) interface device, c) pulser-receiver, d) ultrasonographic probe, e) manual probe, and f) foot pedal to control water flow.

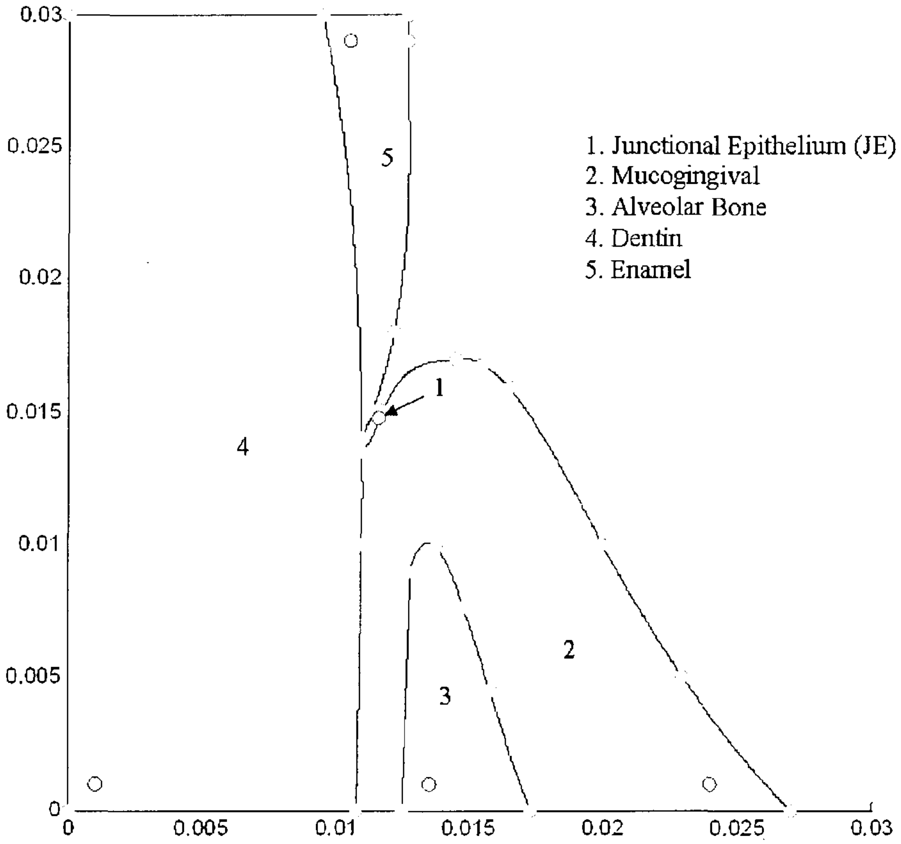


Figure 3.3: The 2D simulation geometry to scale with indications of the different zones in the region. The circles indicate the flexible geometry points that can be moved to adapt to different anatomies. This figure is from [35] and is used with permission.

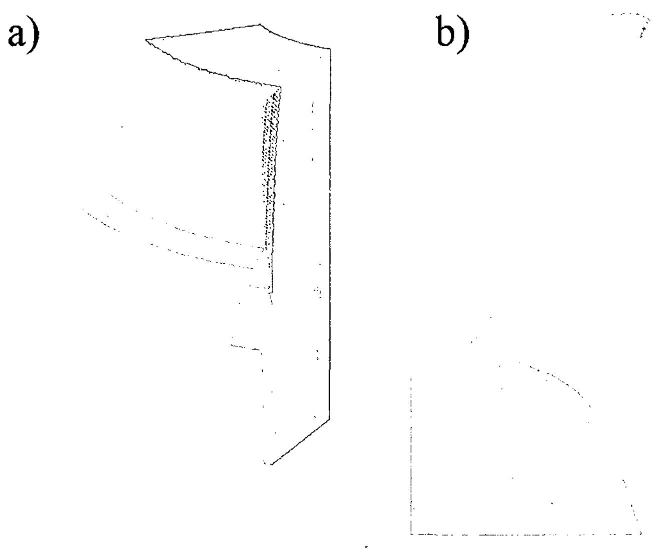


Figure 3.4: a) The 3D periodontal simulation geometry, including tooth, gingiva, bone, water, and probe tip with b) a close-up of the tip placement on the gingiva. This figure is from [35] and is used with permission.

was approximated from standard textbooks in the field [11, 37, 38]. The 2D section (Fig. 3.3) is swept as though around a cylinder to produce the 3D geometry for the simulation (Fig. 3.4). The ringing of the transducer, reflections of the acoustic energy inside the tip, and propagation of the acoustic energy are tracked as the simulation progresses in time. One goal was to determine how much energy reaches the periodontal ligament and the junctional epithelium, which is a layer of soft tissue at the bottom of the periodontal pocket in which the manual periodontal probe bottoms out during clinical probing.

To calculate the amount of energy that reaches the anatomy of interest, the points of the simulation corresponding to the tissue zone of interest are calculated by finding the outline of the tissue zone in the 2D simulation (Fig. 3.3) and rotating them in the cylindrical sweeping technique originally used to create the 3D geometry (Fig. 3.4).

Then, for each cross-sectional depth, we can find the energy in the tissue zone at that depth by

$$\dot{E} = I \cdot \mathrm{d}a = \sum_{grid} \frac{p^2}{\rho c} A \approx \sum p^2. \tag{3.1}$$

In Equation 3.1, energy flux ($\dot{E}$) is related to the intensity (I) over an area element (da). We have also used the fact that the acoustic energy of a plane wave is directly related to the square of the pressure (p) divided by density (ρ) times acoustic speed of sound (c) [39]. In our case, these last two quantities are similar enough for all the regions of interest, being close to that of water, and will drop out due to normalization. Therefore, in the simulation, the 3D pressure values correspond to the energy by the sum of the square of the pressure values at each point in the tissue zone. Rather than averaging over time, discrete values of time were chosen because of constraints on the simulation run time. Three zones of interest were selected: the junctional epithelium, where the manual periodontal probe is thought to bottom out, as well as the gingiva and the water in the periodontal pocket.

Simulations were performed by approximating the junctional epithelium as skin ($\rho = 1020$ kg/m^3, $c = 1540$ m/s) and the gingva as muscle ($\rho = 1080$ kg/m^3, $c = 1550$ m/s) [40, 41]. The transducer frequency used was 10 MHz. The energy values were normalized two different ways:

1. The energy values in each tissue zone were normalized with respect to the total energy values at that depth (this is the usual definition of normalization)

2. Because of internal reflections and constructive/destructive interference within the tip, the energy in the tip is not constant. At any point in the simulation, interference of reflections in the tip may seem to cause more energy in different regions than in others, and so 8 different points in the tip were chosen for normalization

Also, two different types of simulations were performed. In the first, one pocket depth was chosen (0.060 m from the bottom of the junctional epithelium, which is 1.5 mm thick) and pressure values were recorded for 20 different time intervals. In the second, the pocket depth was varied from $0.025 - 0.095$ m and the pressure values were recorded at one time snapshot corresponding to 1.37×10^{-5} s in the simulation. In the simulation geometry, the origin of the anatomy occurs toward the root of the tooth, so that large values of anatomy height (up to 0.300 m) correspond to regions closer to the transducer part of the tip, while low values of anatomy height correspond to regions below the pocket. For all of the energy studies, the maximum anatomy height displayed corresponds to the point where the tip meets the anatomy, while the lowest anatomy height displayed corresponds to points well below the junctional epithelium. Figure 3.3 shows the relative periodontal anatomy in the simulation.

Figure 3.5 shows the energy ratio in the three different regions normalized by energy available at that height for simulations varying by time or pocket depth. As expected, the gingiva holds the largest percentage of the available energy, as it does take up most of the space. The water in the pocket holds the next largest available energy, which is located mostly closest to where the tip meets the anatomy. The most energy in the junctional epithelium is located earlier in space and time, which makes sense, since the energy has the least opportunity to be diffused. When normalized relative to the energy available at that height, there is at most 27.6% of the energy collected in the junctional epithelium, which is considerable.

Figure 3.6 shows the max energy ratio in the zones of interest normalized by the energy in different points of the tip. The energy ratio results for the gingiva are illuminating, because at best, 15.3% of the energy in the tip is translated into the anatomy at all. In all cases, the largest energy ratio results when the zone's energy is normalized with respect to the point in the tip closest to the anatomy. It is also

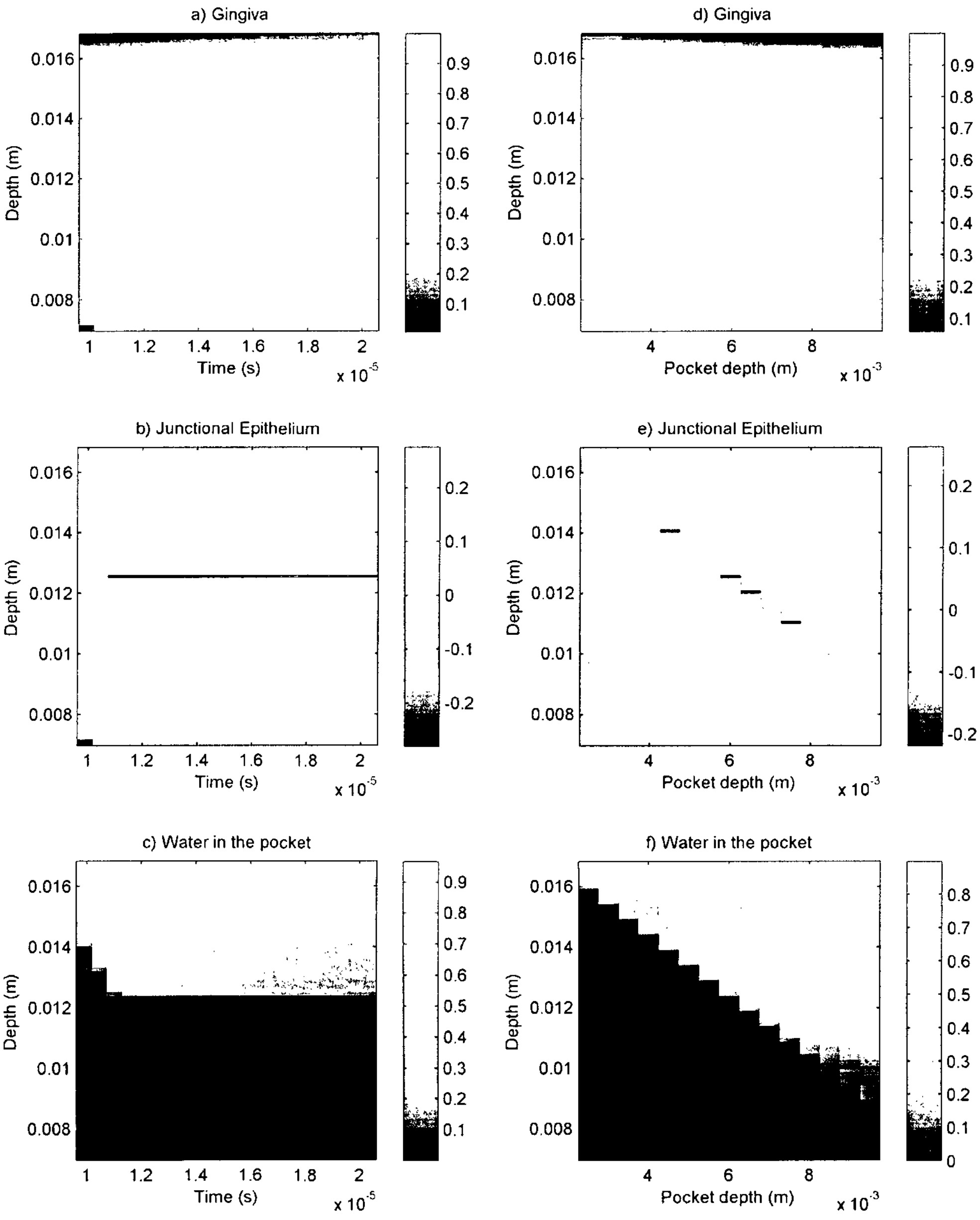


Figure 3.5: Energy values of the soft tissue approximation are displayed in a phase space. The z-axis of the physical simulation space (Figure 3.3) is the same as the vertical of these phase space plots. The horizontal axis in these phase space plots correspond to the independent variable of the simulation, either time in the simulation or energies of different pocket depths. Figures a)-c) are simulations in which pocket depth is held constant while pressure values were captured at varying times in the simulation, while d)-f) are simulations in which pressure values are captured at only one time but pocket depth varies. The color bar legend on the right of each graph indicates amplitude of the resulting energy ratio normalized by energy available at the indicated height in the anatomy.

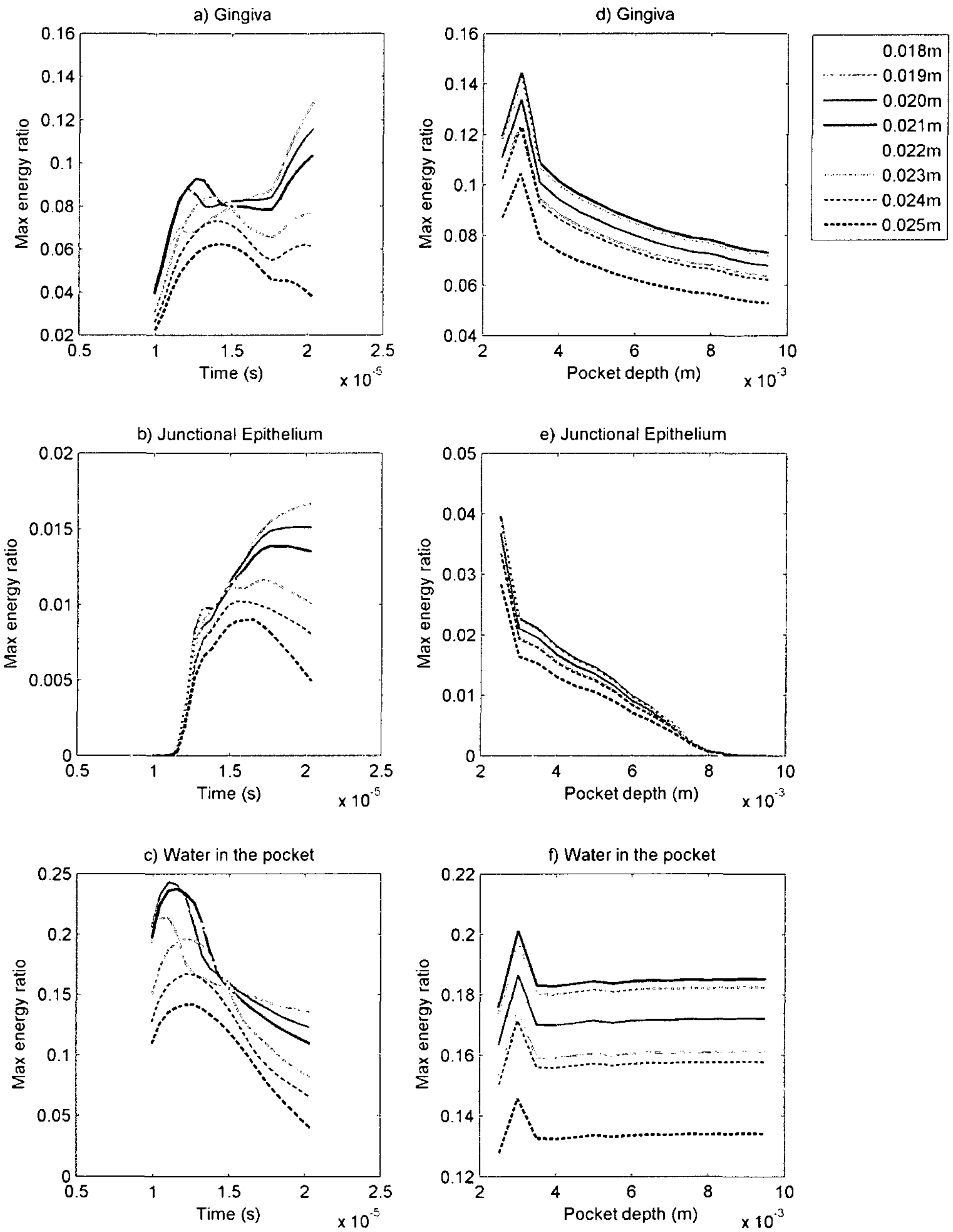


Figure 3.6: The energy values are now normalized by energies at 8 different points in the tip, and the maximum of the energy was selected and plotted against the dependent variable of the simulation.

encouraging that as much as 24.3% of the energy in the tip propagates to the water in the pocket. However, very little of the energy in the tip propagates to the junctional epithelium, less than 4%. This is a manifestation of the fact that no sizable reflection from the junctional epithelium is visible in the soft tissue approximation simulation A-lines. Table 3.1 gives a numerical summary of these results.

Table 3.1: The maximum values of energy values for each of the three zones are displayed here for both types of normalizations and both simulations. In the case of those normalized by tip energy, the maximum value of the energy ratio over the 8 different tip locations was selected.

	Normalized by total energy			Normalized by tip energy		
	Gingiva	JE	Pocket	Gingiva	JE	Pocket
Time varies	1.000	0.276	0.967	0.146	0.019	0.243
Pocket depth varies	1.000	0.262	0.899	0.153	0.042	0.213

In all, little of the energy present in the tip propagates to the junctional epithelium, a reflection from which would correspond to the manual periodontal probe measurement. Hence, standard peak location measurements will not be sufficient to translate the ultrasonographic probe measurements to manual probe measurements but sophisticated mathematical techniques may be able to resolve the 4% of energy present in the junctional epithelium.

3.4 Data Collection

The previous publications discussed above described the fabrication of several generations of prototype ultrasonographic probes. The body of the 5th generation probe is manufactured similarly to other dental hand pieces, with the 10 MHz piezoelectric transducer located in the head of the probe. Water is the ultrasonic coupling agent, so the fabrication of the probe allows water to be funneled through the custom-shaped tip. The rest of the components used to control the probe, including the general purpose pulser-receiver and the custom built water flow interface device, are shown in Figure 3.2.

Figure 3.7: Pictured is the apparatus shown in Fig. 3.2 being operated by Gayle McCombs, with Jonathan Stevens monitoring the computer.

In April and May of 2007, clinical tests were performed at Old Dominion University's (ODU) Dental Hygeiene Research Center on 12 patients using both the ultrasonographic periodontal probe and the traditional manual periodontal probe (Figure 3.7). The human subjects protocol was approved by IRBs at both William and Mary and ODU. Most of the measurements were for healthy subjects, with 76% of the data measuring 3 mm or less. Figure 3.8 shows a distribution of the data versus manual pocket depth. The ultrasonographic probe measurement was always performed first, and 30 repeated ultrasonic waveforms were recorded at the standard 6 tooth sites in at most 28 teeth per patient. Simulations using a finite integration technique were also performed by our group prior to the clinical tests [36]. The simulations animate the generation of ultrasonic energy in the transducer and its propagation through the tissues. The output of the simulations includes a time-domain waveform recorded by the transducer. Figure 3.9 compares the simulated waveform with a sample filtered experimental waveform. Note that the region of interest in between the first and second reflections from the tip does not register any obvious reflections from the periodontal pocket.

To detect pocket depth, further mathematical abstractions will be required. The basic steps of the artificial intelligence are:

1. Feature Extraction: Get wavelet fingerprints with DWFP and find fingerprint properties using image recognition software.

2. Feature Selection: Find μ and σ (Equations 3.4 and 3.5) of each property for all waveforms and collect key values where the average property varies per pocket depth.

3. Binary Classification: Compare the selected features in a leave-one-out routine using well-known pattern classification schemes and only two possible pocket depths at a time.

4. Dimensionality Reduction: Evaluate binary labels using four different methods to collapse each binary choice to one label.

5. Classifier Combination: Combine the predicted labels from the most precise tests to improve accuracy and/or spread of labels.

Each step is explained in further detail in the sections that follow. It is also important to note that because of the computation time of this task, the computer algorithms were adapted to run on William and Mary's Scientific Computing Cluster[2].

[2] `http://www.compsci.wm.edu/SciClone/`

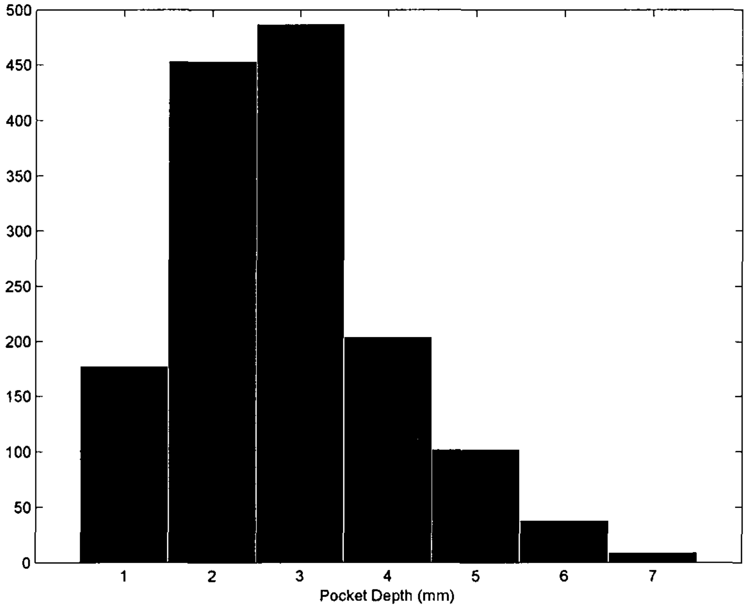


Figure 3.8: The histogram shows the distribution of manual periodontal pocket depth measurements for the ODU clinical tests. Note that the majority of the population has healthy ($\leq$ 3 mm) pocket depths.

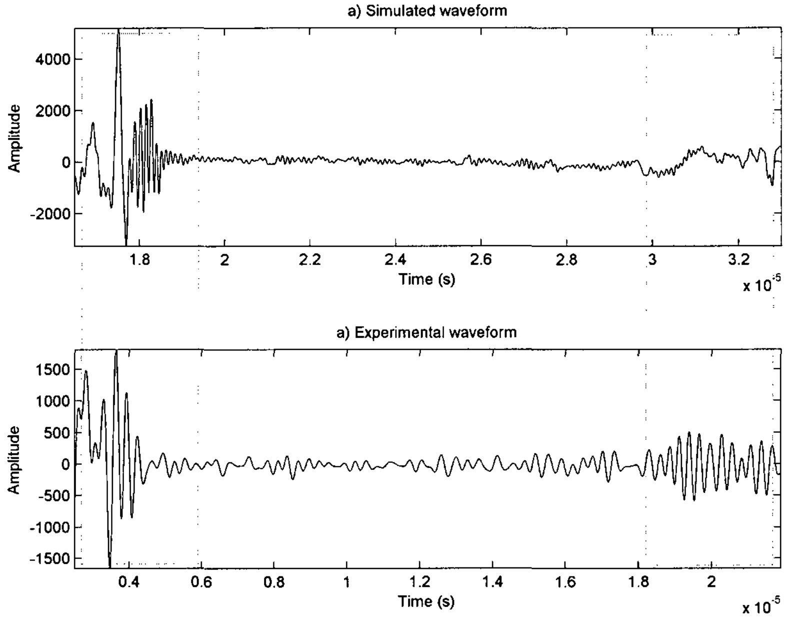


Figure 3.9: A sample waveform from a) simulation and b) experiment are compared here. The large reflections indicated in the boxes are artifacts from the tip geometry. The echo from the bottom of the pocket, which would occur in between the rectangles, is not apparent.

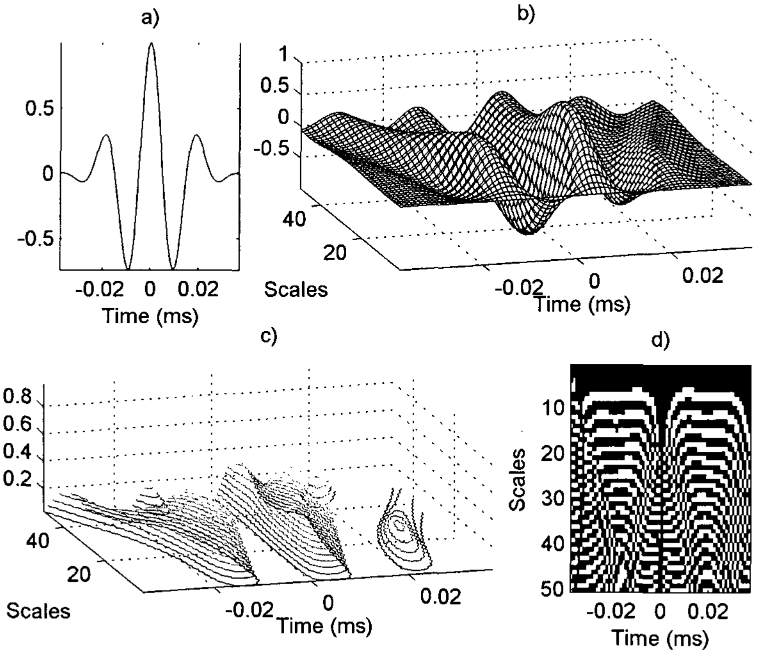


Figure 3.10: The DWFP technique [32] begins with a) the ultrasonic signal, where it geneates b) wavelet coefficients indexed by time and scale. Then c) the coefficients are sliced and projected onto the time-scale plane (d). The slicing operation produces a 2D binary image that is like a contour plot but easier to interpret using image recognition techniques.

3.5 Feature Extraction

Since reflections from the periodontal ligament are not apparent in the ultrasound waveforms, advanced mathematics are needed to identify pocket depth. The clinical trials yielded ultrasonic waveforms $w_{s,k}(t)$, where there are $k = 1, \ldots, 30$ waveforms recorded for each tooth site $s = 1, \ldots, 1470$. The continuous wavelet transform can be written

$$C(a,b) = \int_{-\infty}^{+\infty} w(t)\overline{\psi_{a,b}}dt \tag{3.2}$$

Here, $w(t)$ represents a square-integrable 1D function, and $\psi(t)$ represents the mother wavelet. The mother wavelet is scaled in frequency (f) and transformed in time (t) and using $a, b \in \mathbb{R}$, respectively, in order to form the $\psi_{a,b}(t)$ in Equation 3.2.

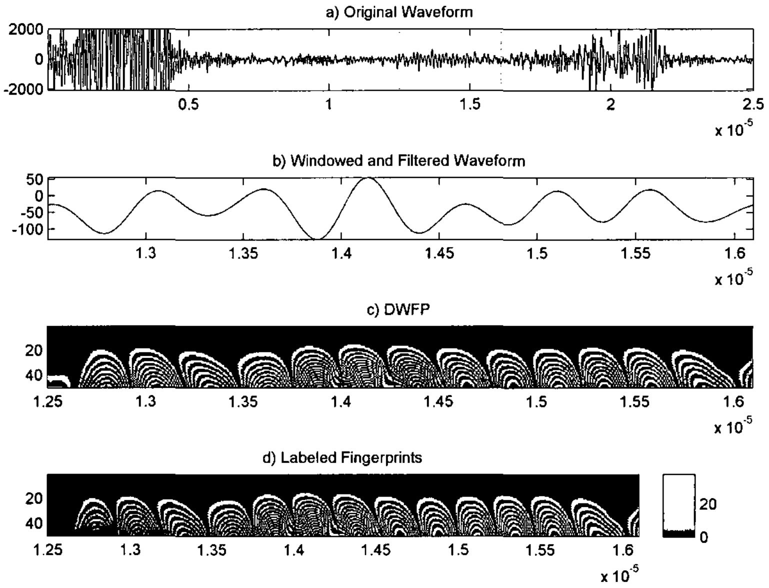


Figure 3.11: In a)-c), the DWFP is shown applied to a filtered window of the original data. Image recognition software through MATLAB are used to recognize each individual fingerprint (d) and measure their image properties.

To extract features from $w_{s,k}(t)$ for classification, we use the Dynamic Wavelet Fingerprinting Technique (DWFP) (Figure 3.10), which creates binary contour plots of the wavelet transform coefficients $C(a,b)$. The DWFP, along with a mother wavelet $\psi(t)$ and some scaling and translation parameters a, b, applied to the waveforms $w_{s,k}(t)$ yields an image, I (Figure 3.11):

$$w_{s,k}(t) \overset{DWFP(\psi_{a,b})}{\longrightarrow} I_{s,k}(a,b) \tag{3.3}$$

Preliminary tests showed that mother wavelets Debauchies 3 (db3) and Symelet 5 (sym5) showed promise for this application, and so both were applied in this technique. The resulting image I contains fingerprint-like binary contours of the initial waveform $w_{s,k}(t)$ at tooth site s.

The next step is to perform image processing through MATLAB's toolbox in order to gather properties of the fingerprints in the waveform $w_{s,k}$. First, the binary image I is labeled with the 8-connected objects (Figure 3.11d), allowing each individual fingerprint in I to be recognized as a separate object using the procedure in Haralick and Shapiro [42]. Next, properties are measured from each fingerprint. Some of these properties include counting the on- and off-pixels in the region, but many involve fitting an ellipse matching the second moments of the fingerprint and measuring properties of that ellipse, such as eccentricity. In addition to the orientation measure provided by the ellipse, another measurement of inclination relative to the horizontal axis was determined by Horn's method for a continuous 2D object [43]. Lastly, further properties were measured by determining the boundary of the fingerprint and fitting 2nd or 4th order polynomials. Table 3.14 in the appendix summarizes the features as well as the selected indices from Section 3.6.

The image processing routines result in fingerprint properties $p_{s,k,n}[t]$, where n

represents the image processing- extracted fingerprint properties. These properties are discrete in time because the values of the properties are matched to the time value of the fingerprint's center of mass. Linear interpolation yields a smoothed array of property values, $p_{s,k,n}(t)$. Figure 3.12a shows the sparse values of the DWFP orientation values for one tooth site, while Figure 3.12b shows the smoothed values.

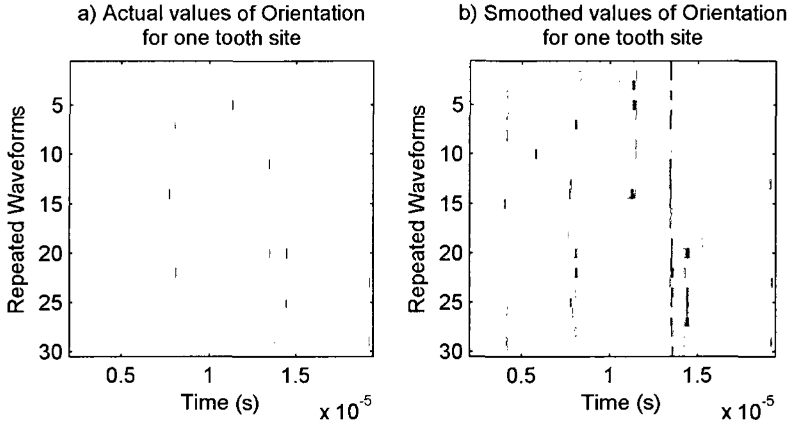


Figure 3.12: The image created by the actual wavelet fingerprint properties (a) has been smoothed using a linear approximation for the intervening points (b). The smoothing shown here is indexed by one value of s, ψ, n but all values of k.

3.6 Feature Selection

It is now possible to begin reducing the dimensionality of the features extracted from the DWFP technique. The reasoning is three-fold. First, most of the measurements do not likely correspond to pocket depth, since the periodontal pocket is a discrete event in the ultrasonic waveform while the features extracted from the DWFP technique are a 1D array in the time domain. Another reason to cull information from the wavelet fingerprint property data set is that it is too large to manipulate even on the available computing cluster. Lastly, the set of extracted features, $p_{s,k,n}(t)$, is too large to use directly in pattern classification routines.

One dimension can be eliminated immediately by averaging over the repeated waveforms, k, so $p = p_{s,n}(t)$. The remaining dimensionality reduction will be performed by selecting values of p at times t_i for particular fingerprint properties n_i so the feature vector at tooth site s, f_s, will be formed of $f_s(i) = p_{s,n_i}(t_i)$. Then the classification matrix will have rows corresponding to s and columns corresponding to $f_s(i)$.

To select features of interest, we look for values of the fingerprint properties that are different, on average, for different measured values of pocket depth. Therefore, for each property n, we find the mean (3.4) and standard deviation (3.5) over that property for all tooth sites:

$$m_n(t) = \frac{1}{N}\sum_{s=1}^{N} p_{s,n}(t) \tag{3.4}$$

$$\sigma_n(t) = \sqrt{\frac{1}{N}\sum_{s=1}^{N}(p_{s,n}(t) - m_n t)^2} \tag{3.5}$$

The selected features correspond to the times t_i at which $m_n(t_i)$ varies greatly for different pocket depths while $\sigma_n(t_i)$ remains small. For a particular set of properties, n_i, and their corresponding times, t_i, the feature vector for tooth site s would become

$$f_s = \{p_{s,n_i}(t_i)\}$$

Figure 3.13 shows an example, with the regions of interest marked out by boxes. This feature selection process was performed interactively, not automatically, though it would be possible to apply this technique with thresholds on the mean and standard deviation. Also, it is important to note that the black vertical lines mark the boundaries of the window regions used in the original wavelet fingerprinting. The wavelet fingerprints distort along the edges of the window, so the often extreme behavior of the feature vectors near these points is disregarded. In the end, 54 different features were selected from the DWFP properties from two different mother wavelets (see Table 3.14 in the Appendix).

Two other methods of reducing the dimensionality of the extracted feature space are possible, however. The method described above first averaged over the k repeated waveforms, but it is also possible to avoid reducing dimensionality until after classification. One option is to build feature vectors for each of the k repeated waveforms and classify them separately. This yields feature vectors of the form

$$f_{s,k} = \{p_{s,k,n_i}(t_i)\},$$

which are used for training and testing of the classifier related to each of the $k = 1 \ldots 30$ waveforms. The extra kth dimension is collapsed later. Another way is to use the f_s for training and the $f_{s,k}$ for testing the classifier, again reserving dimensionality reduction until after classification. All three of these basic methods were performed:

1. Train and test the classifier with features f_s

2. Train and test the classifier $f_{s,k}$ for each $k = 1 \ldots 30$

3. Train the classifier with f_s and test it with $f_{s,k}$

3.7 Classification: A Binary Classification Algorithm

Once the features have been generated for the wavelet fingerprint properties, we then apply various standard pattern classification techniques from the PRTools catalog of MATLAB functions [44]. Many of these are Bayesian techniques (Table 3.2), except

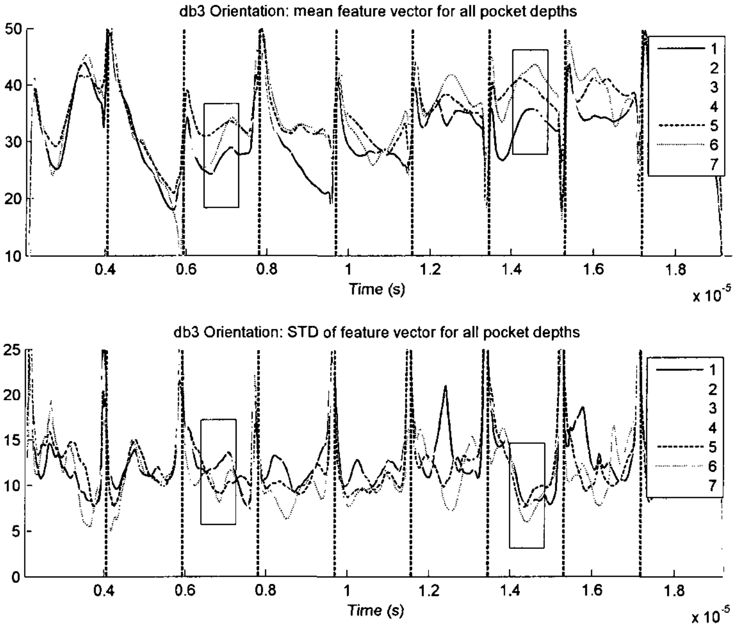


Figure 3.13: The feature vector created by averaging the properties in Figure 3.12 over the repeated waveforms is plotted here. The first is the average over k for each pocket depth (Eqn 3.4), and the second is the standard deviation (Eqn 3.5). The red boxes indicate time values of the property selected for classification (Table 3.14), when the mean differs while the standard deviation is low.

Table 3.2: A list of the standard classifiers from the PRTools catalogue of classification functions for MATLAB [44]. These maps were also later used in the binary classification algorithm

Classifier	Description
LDC	Linear classifier using normal densities
QDC	Quadratic classifier using normal densities
KLLDC	Linear classifier using Karhunen-Loève expansion of covariance matrix
PCLDC	Linear classifier using principle component analysis
LOGLC	Linear logistic classifier
FISHERC	Linear minimum least square classifier
NMC	Nearest mean classifier
NMSC	Scaled NMC
POLYC	Untrained classification with additional polynomial features
SUBSC	Subspace classifier
KNNC	k-nearest-neighbor

in this case we have set the prior probabilities as equal for all classes. A leave-one-out sampling technique was used so that for each $s = 1, \ldots, 1470$ tooth site used as a testing feature vector, the classifiers in Table 3.2 were trained on the remaining $s - \{1, \ldots, 1470\}$ feature vectors. There is then a $1-1$ correspondence between the predicted class label and the actual manually-measured pocket depth.

Tables 3.3-3.5 show the accuracy of this leave-one-out classification on the features gathered from the ultrasonographic periodontal probe data set. The accuracy was measured in two ways - the average of the accuracy for each pocket depth bin, and the total accuracy for all labels. The accuracy within the manual probe's tolerance of ±1 mm was also measured. Lastly, different ranges of possible pocket depths were tested, including $1-7$ mm (Table 3.3), $1-5$ mm (Table 3.4), and $2-5$ mm (Table 3.5).

Some of the results seem quite good, so that the total accuracy within ±1 mm is as high as 80%. However, the classifier achieves this accuracy by classifying most of the waveforms to those pocket depths that have the largest number of objects, namely, 2 and 3 mm, because they are the most populated pocket depths in the clinical data set. Figure 3.14 shows the actual and predicted class label distributions for manually-measured pocket depths ranging from $1-7$ mm (the natural distribution), as well as restricting the possible pocket depths to more-evenly-distributed ranges of $1-5$ mm and $2-5$ mm. The classifier here was k-nearest-neighbor (KNNC) tested in a leave-one-out method, which achieved some of the highest accuracies from Tables 3.3-3.13. Note that in all three cases, 2 and 3 mm received by far the most predicted labels, far out of proportion to their natural distribution.

Table 3.3: The accuracy of the classifiers from Table 3.2 using leave-one-out sampling for all $1-7$ mm manually-measured pocket depths is shown. Two measures are given - the average accuracy per pocket depth ("Avg") and the total accuracy for all labels ("Total"). Also shown is the accuracy within the manual probe's tolerance of ±1 mm.

	% Correct		% Correct (±1 mm)	
Map	Avg	Total	Avg	Total
LDC	22.1	23.3	54.2	57.6
QDC	14.3	0.6	28.6	3.2
KLLDC	22.1	23.3	54.2	57.6
PCLDC	22.1	23.3	54.2	57.6
LOGLC	23.7	26.7	54.2	60.5
FISHERC	18.3	38.4	50.2	81.6
NMC	22.2	17.0	47.2	44.4
NMSC	24.1	22.2	56.0	53.9
POLYC	18.3	38.4	50.2	81.6
SUBSC	18.4	21.9	44.2	52.5
KNNC	17.3	38.3	47.2	80.7

In order to counter this predisposition of the standard classification schemes to classify all the objects into the highest volume classes, a binary classification scheme was developed. The procedure is similar to the one-versus-one technique of Support Vector Machines [45]. The basic idea is to classify the waveform associated with any tooth site against only two possible classes at a time using standard classifiers

Table 3.4: Similarly to Table 3.3, the accuracy of different classifiers using leave-one-out sampling for all 1 – 5 mm manually-measured pocket depths is shown.

	% Correct		% Correct (±1 mm)	
Map	Avg	Total	Avg	Total
LDC	30.7	28.5	69.6	68.6
QDC	32.2	36.6	72.2	80.3
KLLDC	30.7	28.5	69.6	68.6
PCLDC	30.7	28.5	69.6	68.6
LOGLC	30.1	30.6	69.1	70.3
FISHERC	25.3	39.8	69.3	84.1
NMC	27.6	22.7	59.6	59.0
NMSC	32.1	26.6	66.9	64.6
POLYC	25.3	39.8	69.3	84.1
SUBSC	26.4	28.3	58.2	65.8
KNNC	22.5	37.2	65.0	82.9

Table 3.5: Similarly to Tables 3.3 and 3.4, the accuracy of different classifiers using leave-one-out sampling for all 2 – 5 mm manually-measured pocket depths is shown.

	% Correct		% Correct (±1 mm)	
Map	Avg	Total	Avg	Total
LDC	34.7	36.0	72.9	74.2
QDC	35.5	41.2	72.5	82.7
KLLDC	34.7	36.0	72.9	74.2
PCLDC	34.7	36.0	72.9	74.2
LOGLC	35.4	38.8	73.2	76.1
FISHERC	30.8	45.2	66.9	85.3
NMC	31.3	33.2	65.3	70.2
NMSC	35.9	36.6	69.6	70.5
POLYC	30.8	45.2	66.9	85.3
SUBSC	29.1	35.7	66.3	73.7
KNNC	29.1	44.1	66.5	86.2

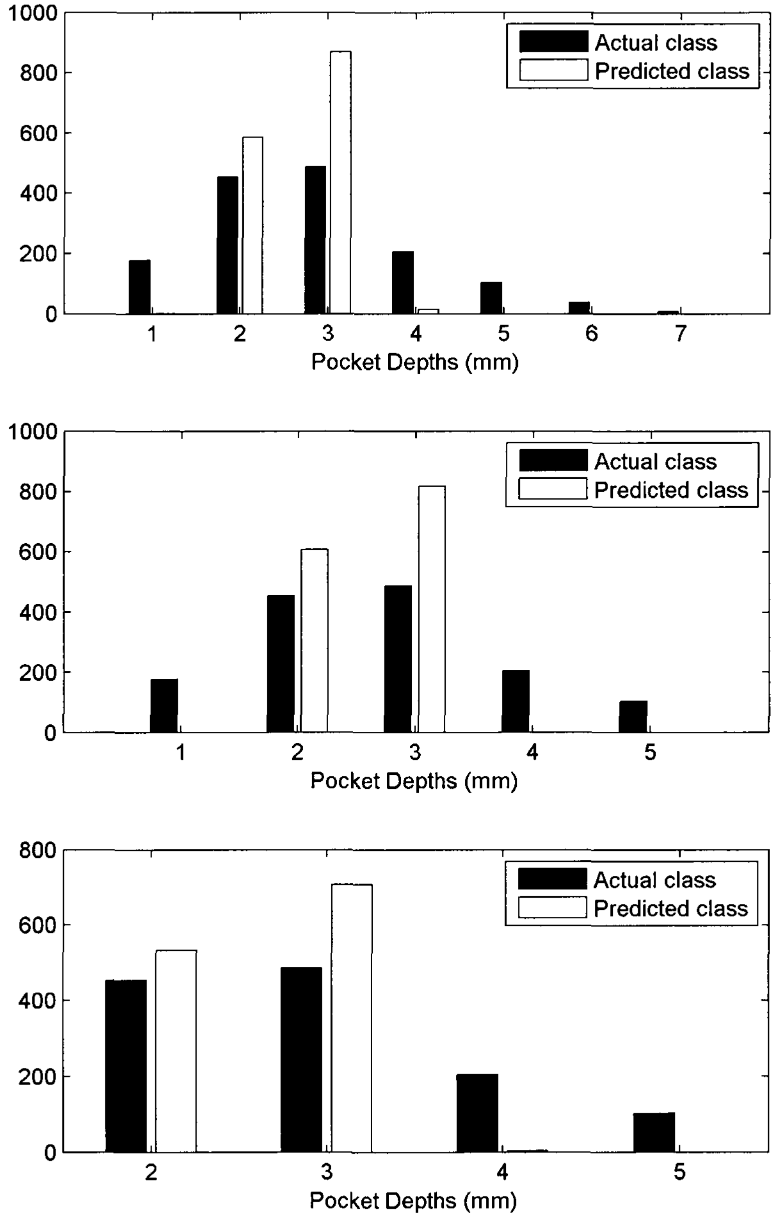


Figure 3.14: Each plot shows the distribution of actual and predicted labels using leave-one-out classification with the classifier KNNC described in Table 3.2. Three different subsets of manually-measured pocket depths were tested in this way with no restriction on the natural distribution.

from Table 3.2. The training and test sets are divided from the data using a leave-one-out technique. If the number of objects in each class differs, a random sample from the larger class of equal size to the smaller class is chosen for training. This process is repeated until at least 90% of the waveforms from the larger class size have been sampled. With each repetition, the predicted labels are stored. The procedure is labeled a binary algorithm because each classification is restricted to only two possible pocket depth values. Program 1 shows a flow chart of our binary classification algorithm.

Program 1 A sketch of the steps in the binary classification algorithm are shown below. The classification step was performed with [44]

```
INPUT:        $\mathcal{C}$            classifier map (such as KNN)
              $f_{s,k}(i)$      array of features
              $d_{s,k}$          array of manually-measured
                        pocket depth values
INDICES:
              $s$            tooth site index
              $k$            index of repeated waveforms
              $i$            feature vector index
              $r$            repetition index
FOR LOOPS:    pick      $k \in 1,\ldots,30$
              pick      $s \in 1,\ldots,1470$
              pick      binary pocket depth pairs
                        $pd1$, $pd2 \in 1,\ldots 7$, $pd1 \neq pd2$
              pick      $\{S1|(s \notin S1)\&(d_{S1,k} = pd1)\}$
              and       $\{S2|(s \notin S2)\&(d_{S2,k} = pd2)\}$
              with      $|S1| = |S2|$
REPEAT        $r$            until 90% data sampled
                        under these conditions
CLASSIFY      T         $= \mathcal{C}(f_{S_{1,2},k}, d_{S_{1,2},k})$
TEST          $\mathcal{L}$            $= \mathcal{C} \cdot f_{s,k}(i)$
SAVE          labels    $\mathcal{L}(k, s, pd1, pd2, r)$
```

In general, the predicted pocket depth is calculated as the most frequently labeled value, but before discussing the dimensionality reduction of the matrix of labels, we first examine the viability of the binary classification algorithm.

3.7.1 Binary Classification Algorithm Examples

In our binary classification algorithm, the waveform from one tooth site is tested against binary pairs of all possible pocket depths using the classification matrices, even if the manually-measured pocket depth of that tooth site is not one of the classes in the binary choice of pocket depths. To examine the accuracy of this technique, we first test the binary classification algorithm on a set of waveforms when the manually measured pocket depth value is one of the binary choices. Table 3.6 shows the percentage of manually measured pocket depths correctly identified by the binary classification algorithm using a linear Bayes normal classifier (LDC). The grid is indexed by the binary pocket depth pairs, so that since no pocket depth class is tested against itself, the diagonal is set to zero. Note that the numbers near the diagonal are smaller, which implies that the scheme finds it difficult to resolve adjacent pocket depth measurements. The fact that adjacent pocket depths are poorly resolved is a positive feature of the classifier, since that may imply that they share similar characteristics, which we would expect from pocket depths only 1 mm apart. Also, the precision of the manual probe is itself only within 1 mm at best, because the markings on the probe are spaced at 1 mm intervals. So it is possible that a reported manual pocket depth measurement could actually differ by 1 mm, due only to the imprecision of that probe.

It is much more difficult to quantify how well the classification scheme works when the manually measured pocket depth is not one of the choices. Consider three examples from the same classifier as above, LDC, using the same classification matrix (Table 3.7). Three different binary pairs are shown. The percent of tooth sites classified in each of the binary pairs are displayed in the table. Note that waveforms will be labeled with the member of the binary pair closest to its manual pocket depth measurement. The accuracy increases the more the binary pocket depth pairs differ.

Table 3.6: In any binary pocket depth pairs *pd*1, *pd*2, the table shows the percent of pocket depths measurements accurately predicted whenever the test object is one of the binary pairs. The rows correspond to *pd*1 and the columns to *pd*2. Note that the larger accuracies occur for largely different pocket depth pairs, which means the classifier tends to classify the tooth site as closer to its manually measured pocket depth value.

	1	2	3	4	5	6	7
1	0	55.4	70.1	75.7	76.3	79.1	74.6
2	57.2	0	62.5	66.2	70.6	73.7	71.5
3	64.3	58.9	0	56.3	61.8	61.6	69
4	70.1	64.7	54.9	0	60.3	66.2	61.8
5	74.5	73.5	59.8	45.1	0	54.9	71.6
6	81.6	68.4	63.2	68.4	44.7	0	65.8
7	66.7	44.4	44.4	44.4	55.6	77.8	0

The results imply that the binary classification scheme not only accurately classifies the ultrasonic waveforms when their manual measurement is one of the binary choices (Table 3.6), but when it is not one of the choices, the binary classification algorithm applies a label closer to the actual value.

Table 3.7: The percent of tooth sites classified in each of the predicted binary pairs is shown for three different binary pair choices.

		Actual Pocket Depth (mm)						
	Label	1	2	3	4	5	6	7
*pd*1	1	55.4	42.8	31.2	26.5	21.6	28.9	44.4
*pd*2	2	44.6	57.2	68.8	73.5	78.4	71.1	55.6
*pd*1	1	74.6	74.6	63.2	57.8	51.0	47.4	33.3
*pd*2	7	25.4	25.4	36.8	42.2	49.0	52.6	66.7
*pd*1	2	83.1	70.6	50.3	38.2	26.5	18.4	22.2
*pd*2	5	16.9	29.4	49.7	61.8	73.5	81.6	77.8

3.7.2 Dimensionality Reduction

As Program 1 shows, the predicted class labels $\mathcal{L}$ returned by the binary classification algorithm are higher-dimensional, even if the index k is averaged over before classification, as discussed in Section 3.6. Four different methods of dimensionality reduction were performed to yield only one label $\mathcal{L}_p(s)$ per tooth site s:

1. Majority Rule: The most frequently labeled pocket depth is declared the predicted pocket depth.

$$\mathcal{L}_p(s) = \text{mode}(\mathcal{L}(k, s, pd1, pd2, r))$$

2. Weighted Probability 1: The first method unfairly weights the labels from the repeated index r. This method first finds the most frequently labeled pocket depth in the repeated index and then calculates the most frequently labeled pocket depth.

$$\begin{aligned} \mathcal{L}'(k, s, pd1, pd2) &= \text{mode}(\mathcal{L}(k, s, pd1, pd2, r)) \\ \mathcal{L}_p(s) &= \text{mode}(\mathcal{L}') \end{aligned}$$

3. Weighted Probability 2: This method creates a normalized vector of weights from the binary pocket depth choice to find the most probable pocket depth.

$$\begin{aligned} w(k, s, pd1, pd2) &= \tfrac{1}{7} \textstyle\sum \mathcal{L}'(k, s, pd1, pd2) \\ \mathcal{L}_p &= \max(w \cdot \mathcal{L}') \end{aligned}$$

4. Process of elimination: Statistics from the above four methods are combined in a process of elimination to predict the final pocket depth.

3.7.3 Classifier Combination

The binary classification algorithm as discussed in Section 3.7 can be configured in three different ways with respect to the waveform index k, and the dimensionality can be reduced in four different ways as discussed in Section 3.7.2. All of these methods were applied using eleven different maps from Table 3.2. To combine the results of these classifiers, each individual classifier was sorted by average accuracy within 1 mm and the highest percent, where the percent ranged from $10 - 95\%$, were combined using the same dimensionality reduction methods described above. The mean or mode of the labels can be calculated, or the most probable label can be calculated. Of these, the mean and highest probability methods are majority voting methods, while the mean method is a democratic voting method. Combining classifiers can often reduce the error of pattern classifiers but can never substitute for proper classification techniques applied when the individual classifiers are formed [46].

3.8 Results and Discussion

The binary classification algorithm was applied as described in the procedure above. The primary method of measuring the success of each technique is finding the percent of waveforms accurately described within 1 mm per pocket depth and averaging over the accuracy per pocket depth. If we instead tried to measure the total number of waveforms accurately described within the manual probe's 1 mm precision regardless of pocket depth measurement, we tend to select for the classification techniques that accurately describe only the most populated pocket depths. We performed these tests for all 7 possible pocket depths from the manually measured data set, but we also restricted the data sets to the $1-5$ mm pocket depths, since there are so few measurements in the $6-7$ mm range in our patient population, and we further restricted the possible pocket depths to the $2-5$ mm set, since we felt the 1 mm data sets might be poorly described by the ultrasonographic probe because overlapping reflections from inside the tip seem to cover that range. We show here results before classifier combination and after, but only the highest average percent accurately identified within 1 mm will be shown. The following results give the percent correctly identified per pocket depth (% correct) as well as the percent accurately identified within 1 mm tolerance (% close). We also show the results graphically in a chart indexed by the manual periodontal probe and the ultrasonographic periodontal probe. These charts are useful in determining the strengths and weaknesses of each classification routine. Each row was normalized by the number of manual measurements per pocket depth to yield the percent of predicted pocket depth labels for each manual pocket depth.

The results displayed in Figure 3.15a-Figure 3.15c show results without combining labels from different binary classification schemes for three different collections of possible pocket depths. Figure 3.15d-Figure 3.15f similarly show results when classifiers are combined. There is a small improvement when classifiers are combined, and the time required for the extra classification is slight as well. Note that only the most accurate results are displayed here, up to 1 mm confidence. If the goal is to label all the waveforms as closely as possible, regardless of spread, then these results are sufficient. However, as the figures of manual versus ultrasonographic probe results show, especially in the reduced pocket depth cases, the lowest and highest pocket depth values do not receive labels at all in the ultrasonographic probe case. A compromise may be desirable between accuracy and large spread of labels. Figure 3.15g-Figure 3.15j show results for the restricted pocket depth cases with smaller accuracy but more spread in the labels. These were reconfigured to display a balance between a high precision and a larger spread of labels. Figure 3.16 shows how the spread of labels changes in these specially revised cases. Table 3.10 summarizes the average

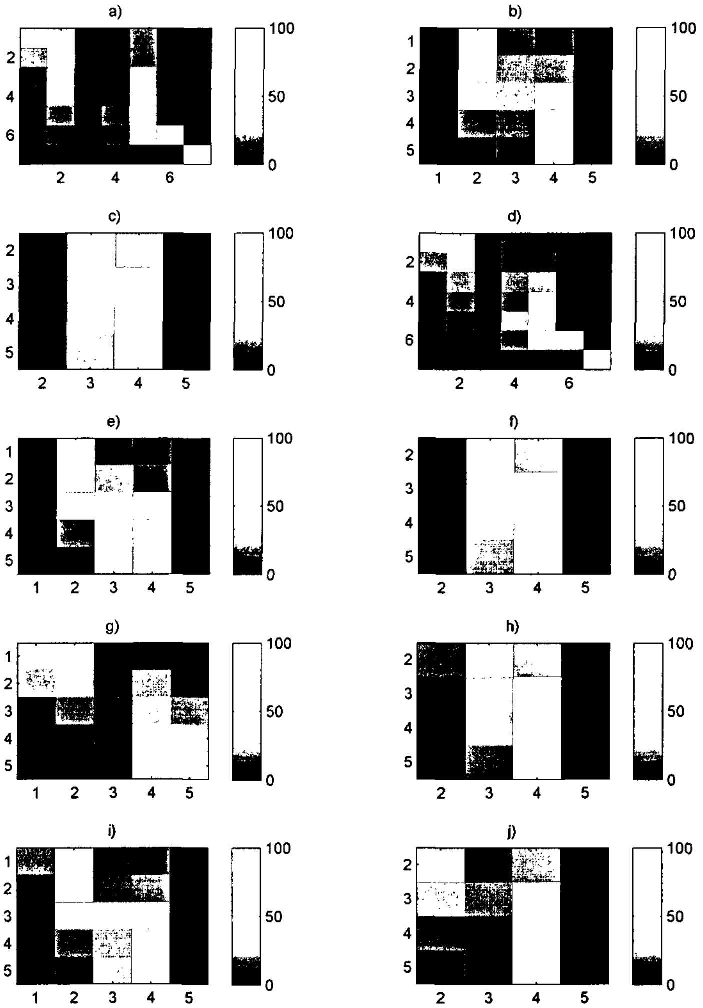


Figure 3.15: The percent of tooth sites with a given manual pocket depth measurement (mm) are plotted versus their ultrasonographic periodontal probe measurement. Plots a)-c) and g)-h) were classifier results obtained without combination, but d)-f) and i)-j) demonstrate the classifier combination results. Plots a)-f) demonstrate the highest possible average accuracy within 1 mm, while g)-j) were manually adjusted to include more spread of labels. The array of possible pocket depths used in the classifier is evident in the axes labels.

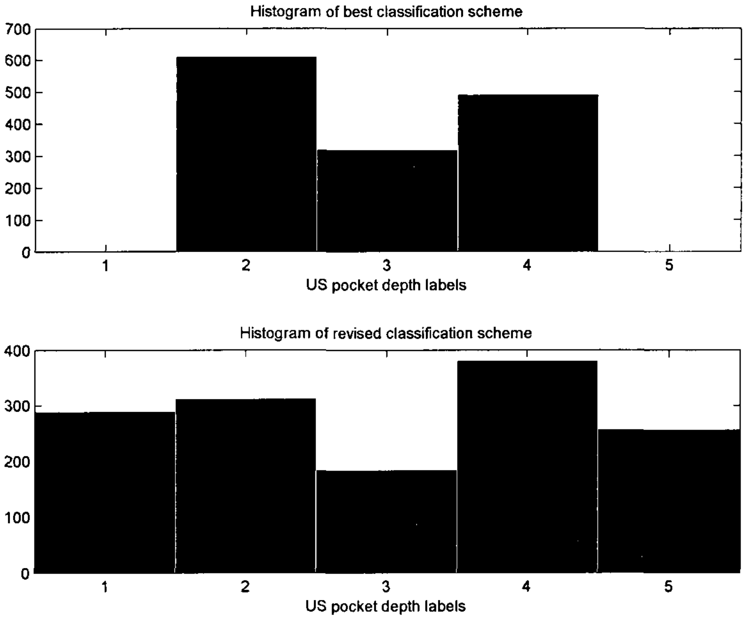


Figure 3.16: A comparison of labeling in the original (Fig 3.15b) and revised (Fig 3.15g) classification schemes for pocket depths 1 – 5 is shown here to illustrate the improved spread of labels.

percent of measurements accurately classified within 1 mm per pocket depth.

Table 3.8: Percent of tooth sites accurately classified by the ultrasonographic probe

Classifier	PDs	% Correct per pocket depth (mm)						
Combination	Used (mm)	1	2	3	4	5	6	7
No	1-7	40.7	42.2	6.6	8.3	50.0	28.9	100.0
No	1-5	0.6	43.9	24.6	56.9	0.0	0.0	0.0
No	2-5	0.0	3.1	46.0	60.3	0.0	0.0	0.0
Yes	1-7	35.0	36.9	8.4	20.1	48.0	34.2	100.0
Yes	1-5	0.0	55.8	28.7	51.5	0.0	0.0	0.0
Yes	2-5	0.0	3.3	49.9	60.3	0.0	0.0	0.0
No-revised	1-5	0.6	43.9	24.6	56.9	0.0	0.0	0.0
No-revised	2-5	0.0	3.1	46.0	60.3	0.0	0.0	0.0
Yes-revised	1-5	0.0	55.8	28.7	51.5	0.0	0.0	0.0
Yes-revised	2-5	0.0	3.3	49.9	60.3	0.0	0.0	0.0

Table 3.9: Percent of tooth sites accurately classified within 1 mm by the ultrasonographic probe

Classifier	PDs	% Correct per pocket depth (mm)						
Combination	Used (mm)	1	2	3	4	5	6	7
No	1-7	71.2	67.8	46.0	53.9	75.5	76.3	100.0
No	1-5	69.5	77.5	99.6	78.9	71.6	0.0	0.0
No	2-5	0.0	70.9	100.0	99.0	73.5	0.0	0.0
Yes	1-7	66.7	63.8	52.4	65.7	84.3	71.1	100.0
Yes	1-5	71.8	81.9	100.0	81.4	64.7	0.0	0.0
Yes	2-5	0.0	71.7	100.0	99.0	75.5	0.0	0.0
No-revsed	1-5	69.5	77.5	99.6	78.9	71.6	0.0	0.0
No-revsed	2-5	0.0	70.9	100.0	99.0	73.5	0.0	0.0
Yes-revised	1-5	71.8	81.9	100.0	81.4	64.7	0.0	0.0
Yes-revised	2-5	0.0	71.7	100.0	99.0	75.5	0.0	0.0

The results above show that the highest average percent accuracy for all the data collected from 12 patients in the ODU clinical trials using a 5th generation prototype is at best 86.6% within a tolerance of 1 mm. Meanwhile, the best accuracy at that

Table 3.10: Summary table of accuracy within 1 mm (%). The table below shows accuracy (%) within the tolerance of 1 mm for the most accurate and revised classification schemes, which involve more widely-spread labels than the original.

Pocket Depths	Classifier Combination			
Used (mm)	No	Revised	Yes	Revised
1-7	70.1	-	72.0	-
1-5	79.4	71.3	79.9	76.8
2-5	85.9	81	86.6	80.6

tolerance using the 4th generation probe from previous methods without using pattern classification was 60% for a single patient [34].

3.9 Bland-Altman Statistical Analysis

We compared the binary classification labels with the manual probe labels using the Bland-Altman method, which is recommended for comparing different measurement schemes [47]. Figure 3.17 shows a sample plot of this method, in which the difference of the ultrasonographic measurements is plotted against the mean of those measurements. Any trend between the difference and the mean of the labels would indicate bias in the measurements. No trend is visible here; the grid-like distribution results from the discrete nature of the labels. The numerical results are displayed by pocket depth in Tables 3.11-3.13. Included are the values and the confidence intervals at 95% of the mean and the mean plus or minus twice the standard deviation ($\mu \pm 2\sigma$). The results show that the mean of the difference is low, less than 0.5 mm in most cases, and the standard deviation is just above 1 mm for most classification schemes. However, the limits of agreement of $\mu \pm 2\sigma$ are large, around 2.5 mm, indicating that some ultrasonographic labels differ widely from the manual labels.

It remains to be determined whether the accuracy of the ultrasonographic probe is sufficient. Bassani *et al.* [47] showed that differences of 1.5 mm in manual pocket

Table 3.11: The table shows the values and confidence intervals at 95% for the mean and the limits of agreement, which is $\mu \pm \sigma$, for classifiers using $1 - 7$ mm.

Pocket Depths	Classifier Combination	
$1 - 7$ mm	No	Yes
μ(mm)	0.44	0.62
Lower Confidence	0.36	0.55
Upper Confidence	0.51	0.69
$\mu - 2\sigma$	-3.12	-2.83
Lower Confidence	-3.25	-2.95
Upper Confidence	-2.98	-2.70
$\mu + 2\sigma$	3.99	4.07
Lower Confidence	3.86	3.94
Upper Confidence	4.12	4.20
σ (mm)	1.77	1.72

Table 3.12: Bland-Altman statistical analysis for classification schemes using pocket depths $1 - 5$ mm.

Pocket Depths	Classifier Combination			
$1 - 5$ mm	No	Revised	Yes	Revised
μ(mm)	0.21	0.30	0.18	0.12
Lower Confidence	0.16	0.24	0.13	0.07
Upper Confidence	0.25	0.36	0.22	0.17
$\mu - 2\sigma$	-1.98	-2.58	-1.98	-2.17
Lower Confidence	-2.06	-2.69	-2.06	-2.25
Upper Confidence	-1.89	-2.47	-1.90	-2.08
$\mu + 2\sigma$	2.39	3.18	2.33	2.40
Lower Confidence	2.31	3.07	2.25	2.32
Upper Confidence	2.47	3.29	2.42	2.49
σ (mm)	1.09	1.44	1.08	1.14

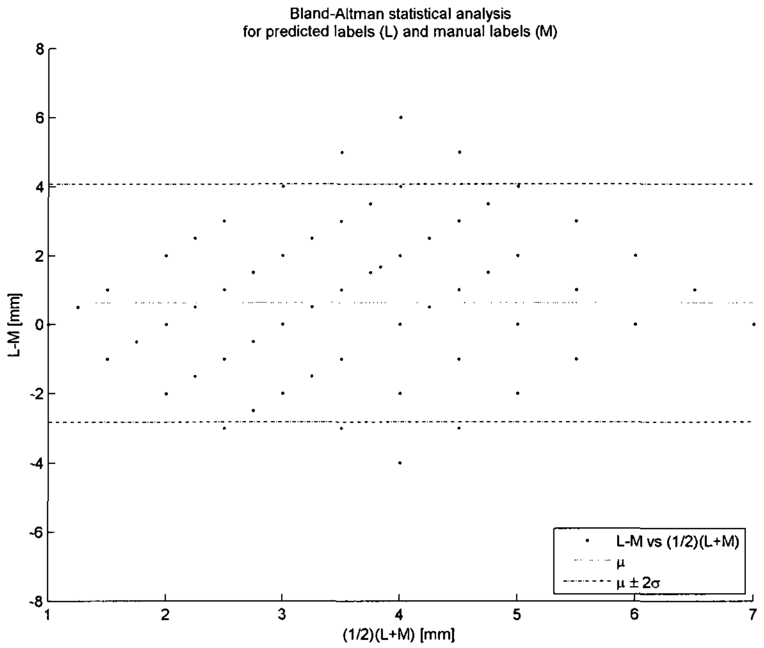


Figure 3.17: A sample Bland-Altman statistical plot is generated here from the best combined classifier results for possible pocket depths 1 − 7 mm. In this figure, only 6% of the labels were outside the range of $\mu \pm 2\sigma$.

depth measurements between different observers are not uncommon, which is closer to the limits of agreement in the 2 − 5 mm classification schemes. Ahmed [48] compared manual to controlled force probes using Bland Altman statistics and found that the limits of agreement ($\mu + \sigma$ to $\mu - \sigma$) was ± 3.31 mm, which is larger than any but the upper limits of agreement of the 1 − 7 mm ultrasonographic probe configuration described here. Yang *et al.* [49] compared measurements of a controlled force probe 4 different ways and determined that, in most cases, the measurement error was around 0.5 mm, which is lower than the manual probe's 1 mm precision. However, controlled force probes tend to be more reproducible than the manual probe, which we are using as our gold standard. Velden [50] showed that, when using controlled-force probes, which are more accurate than manual probes, the best agreement occurred when 0.75

Table 3.13: Bland-Altman statistical analysis for classification schemes using pocket depths $2-5$ mm.

Pocket Depths	Classifier Combination			
$2-5$ mm	No	Revised	Yes	Revised
μ(mm)	0.47	0.32	0.47	0.22
Lower Confidence	0.43	0.27	0.43	0.17
Upper Confidence	0.52	0.37	0.51	0.28
$\mu-2\sigma$	-1.37	-1.77	-1.37	-1.99
Lower Confidence	-1.45	-1.85	-1.45	-2.08
Upper Confidence	-1.30	-1.68	-1.30	-1.90
$\mu+2\sigma$	2.32	2.41	2.31	2.44
Lower Confidence	2.25	2.32	2.24	2.35
Upper Confidence	2.39	2.49	2.39	2.53
σ (mm)	0.92	1.04	0.92	1.11

N of force was applied, and since the manual probe was used as the gold standard here, we cannot be entirely sure what force was being applied. Rams and Slots [51] compared controlled force probes to manual probes and found that the standard deviation between measurements was between $0.9-0.95$ mm for lower pocket depths and $1.0-1.3$ mm for higher pocket depths, and the manual probe almost always had a higher standard deviation than controlled force probes. These values are closer to the standard deviation of the difference in the $2-5$ mm and $1-5$ mm ultrasonographic probe configurations. The authors also found that the measurements were reproducible within 1.0 mm only in $82-89\%$ of the cases, which is within the range of the most accurate ultrasonographic probe classification schemes described here. Mayfield *et al.* [52] found a higher degree of reproducibility, $92-96\%$, when testing 2 different observers. Lastly, Tupta-Veselicky *et al.* [53] found that the reproducibility up to 1 mm with a conventional probe was 92.8% when tested at 20 sites 2 hours apart by one examiner, again demonstrating the inaccuracy of the manual probe itself.

3.10 Conclusion

We have described the development of the ultrasonographic periodontal probe and the clinical tests on 12 patients. The resulting waveforms were analyzed with DWFP processing and image recognition techniques applied to extract numerical measurements from these wavelet fingerprints. These parameters were optimized and averaged to yield training data sets for pattern recognition analysis, where testing data sets were configured for a leave-one-out technique. The binary classification algorithm was described and applied to these classification sets, and the labels from this technique were combined to strengthen the results. Different sets of possible pocket depths can be used, since there are small numbers of measurements in several of the pocket depths. The results can be configured either to yield the highest average percent of waveforms correctly identified within 1 mm, or they can be configured to yield a larger spread in the type of labels with approximately a 5% decrease in accuracy within 1 mm. Overall, the results from the classification scheme can identify ultrasonographic periodontal probe measurements closely to the manual probe, so that $70.1 - 86.6\%$ of the ultrasonographic measurements are within 1 mm of the manual measurement. These values are close to those in the literature comparing other periodontal probes to the manual probe and may be due to the imprecision and low reproducibility of the manual probe itself.

We conclude that the ultrasonographic periodontal probe is a viable technology and that we can use sophisticated mathematical techniques to extract pocket depth information. To yield better results, a larger training data set will be required.

3.11 Appendix

The feature selection method described in Section 3.6 was applied to collect sample number indices to later construct feature vectors from the fingerprint properties. Table 3.14 describes each property used for both mother wavelets and the sample numbers that were used to select the feature vectors.

Table 3.14: The types of features extracted using image recognition techniques from MATLAB are described here, along with the sample number indices resulting from the feature selection process described above from fingerprints generated using each mother wavelet (db3, sym5).

		Sample numbers	
Feature	Description	db3	sym5
Inclination	Angle between fingerprint and horizontal axis, measured by binary image recognition techniques	1463, 2300, 2661	1200, 2400
Orientation	Angle between fingerprint and horizontal axis, measured by ellipse matching second moments	1040, 2300	2215
Eccentricity	Eccentricity of ellipse matching the second moments of the fingerprint	2850	1150, 1500, 2540
Area	Number of pixels in the fingerprint region	1200, 2300, 2600	2270, 2800
Y-center	Value of y-axis (scale) of the binary image associated with its center of mass	1470, 2600	1400, 2258, 2550, 2650
Euler Number	Number of objects in the fingerprint minus the number of holes in the object	1572	-
Deg2fit, p1	First coefficient of 2nd degree polynomial fitting the boundary of the fingerprint	1200, 2672	1166, 1600, 2266, 2566
Deg2fit, p2	Second coefficient of 2nd degree polynomial	-	2275
Deg2fit, p3	Third coefficient of 2nd degree polynomial	-	1146, 2533
Deg4fit, p1	First coefficient of 4th degree polynomial fitting the boundary of the fingerprint	2833	2042, 2250, 2550, 2650
Deg4fit, p2	Second coefficient of 4th degree polynomial	2831	1500, 2275, 2575
Deg4fit, p3	Third coefficient of 4th degree polynomial	1200	1495, 2275, 2575
Deg4fit, p4	Forth coefficient of 4th degree polynomial	2850	2565
Deg4fit, p5	Fifth coefficient of 4th degree polynomial	2865	-

Bibliography

[1] S. R. Ghorayeb, C. A. Bertoncini, and M. K. Hinders, "Ultrasonography in dentistry," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 6, pp. 1256–1266, 2008.

[2] M. K. Hinders and J. A. Companion, "Ultrasonic periodontal probe," in *25th Review of Progress in Quantitative Nondestructive Evaluation 18b*, D. O. Thompson and D. E. Chimenti, Eds. New York: Plenum Press, 1998, pp. 1609–1615.

[3] M. K. Hinders, A. Guan, and J. A. Companion, "Ultrasonic periodontal probe," *Journal of the Acoustical Society of America*, vol. 104, no. 3, p. 1844, 1998, abstract.

[4] S. K. Hartman, "Goodbye gingivitis," in *Virginia Business*, 1997, p. 9.

[5] J. A. Companion, "Differential measurement periodontal structures mapping system," US Patent 5,755,571, 1998.

[6] C. Farr, "Ultrasonic probing: the wave of the future in dentistry," *Dentistry Today*, vol. 19, no. 3, pp. 86–91, 2000.

[7] M. K. Hinders, J. E. Lynch, and G. B. McCombs, "Clinical tests of an ultrasonic periodontal probe," in 28^{th} *Review of Progress in Quantitative Nondestructive*

Evaluation 21b, D. O. Thompson and D. E. Chimenti, Eds. Melville, New York: AIP Conference Proceedings, 2001, pp. 1880–1890.

[8] J. E. Lynch, "Ultrasonographic measurement of periodontal attachment levels," Ph.D. dissertation, Department of Applied Science, College of William and Mary, Williamsburg, VA, 2001.

[9] J. E. Lynch and M. K. Hinders, "Ultrasonic device for measuring periodontal attachment levels," *Review of Scientific Instruments*, vol. 73, no. 7, pp. 2686–2693, 2002.

[10] J. Hou, "Ultrasonic signal detection and characterization using dynamic wavelet fingerprints," Ph.D. dissertation, Department of Applied Science, College of William and Mary, Williamsburg, VA, 2004.

[11] F. A. Carranza and M. G. Newman, *Clinical Periodontology*, 8th ed. Philadelphia, PA: W B Saunders Co, 1996.

[12] P. E. Peterson and H. Ogawa, "Strengthening the prevention of periodontal disease: The who approach," *Journal of Periodontology*, vol. 76, no. 12, pp. 2187–2193, 2005.

[13] N. P. Lang and E. F. Corbet, "Periodontal diagnosis in daily practice," *International Dental Journal*, vol. 45, pp. 5–15, 1995.

[14] G. Baum, I. Greenwood, S. Slawski, and R. Smirnow, "Observation of internal structures of teeth by ultrasonography," *Science*, vol. 139, no. 3554, pp. 495–496, 1963.

[15] S. Lees and F. E. Barber, "Looking into teeth with ultrasound," *Science*, vol. 161, no. 3840, pp. 477–478, 1968.

[16] W. L. Kydd, C. H. Daly, and J. B. Wheeler 3rd, "The thickness measurement of masticatory mucosa *in vivo*," *International Dental Journal*, vol. 21, no. 4, pp. 430–441, 1971.

[17] H. Spranger, "Ultra-sonic diagnosis of marginal periodontal diseases," *International Dental Journal*, vol. 21, no. 4, pp. 442–455, 1971.

[18] M. H. Palou, M. J. McQuade, and J. A. Rossmann, "The use of ultrasound for the determination of periodontal bone morphology," *Journal of Periodontology*, vol. 58, no. 4, pp. 262–265, 1987.

[19] C. Löst, K. M. Irion, and W. Nüssle, "Periodontal ultrasonic diagnosis: experiments on thin bony platelets and on a simulated periodontal ligament space," *Journal of Periodontal Research*, vol. 23, no. 6, pp. 347–351, 1988.

[20] ——, "Ultrasonic B-scans of the facial/oral periodontium in pigs," *Journal of Clinical Periodontology*, vol. 16, no. 8, pp. 534–538, 1989.

[21] ——, "Determination of the facial/oral alveolar crest using RF-echograms. an *in vitro* study on the periodontium of pigs," *Journal of Clinical Periodontology*, vol. 16, no. 8, pp. 539–544, 1989.

[22] U. Keller, "High frequency sonography in periodontal diagnosis," *Deutsche Zahnärztliche Zeitschrift*, vol. 16, no. 8, pp. 539–544, 1989, article in German.

[23] K. Oshima, "Application of ultrasonography to periodontal diagnosis," *Nippon Shishubyo Gakkai Kaishi*, vol. 31, no. 1, pp. 235–240, 1989, article in Japanese.

[24] K. S. Oikarinen, T. M. Nieminen, H. Mäkäräinen, and J. Pyhtinen, "Visibility of foreign bodies in soft tissue in plain radiographs, computed tomography,

magnetic resonance imaging, and ultrasound," *International Journal of Oral and Maxillofacial Surgery*, vol. 22, pp. 119–124, 1993.

[25] T. Eger, H. Müller, and A. Heinecke, "Ultrasonic determination of gingival thickness," *Journal of Clinical Periodontology*, vol. 23, no. 9, pp. 839–845, 1996.

[26] M. Berson, J. M. Grégoire, F. Gens, J. Rateau, F. Jamet, L. Vaillant, F. Tranquart, and L. Pourcelot, "High frequency (20 MHz) ultrasonic devices: advantages and applications," *European Journal of Ultrasound*, vol. 10, no. 1, pp. 53 – 63, 1999.

[27] F. I. Tsiolis, I. G. Needleman, and G. S. Griffiths, "Periodontal ultrasonography," *Journal of Clinical Periodontology*, vol. 30, pp. 849–854, 2003.

[28] L. D.R., "Ultrasonic periodontal diagnostic instrumentation system with clinical results," *Measurement*, vol. 23, pp. 125–129, 1998.

[29] S. M. Demyun and K. M. Hagenbuch, "Ultrasonic method and apparatus for measuring the periodontal pocket," U. S. Patent 5100318, Shavertown, PA, April 1990.

[30] J. E. Lynch, M. K. Hinders, and G. B. McCombs, "Clinical comparison of an ultrasonographic periodontal probe to manual and controlled-force probing," *Measurement*, vol. 39, no. 5, pp. 429–439, 2006.

[31] M. K. Hinders and G. B. McCombs, "The potential of the ultrasonic probe," *Dimensions of Dental Hygiene*, vol. 4, no. 4, pp. 16–18, 2006.

[32] J. Hou and M. K. Hinders, "Dynamic Wavelet Fingerprint identification of ultrasound signals," *Materials Evaluation*, vol. 60, no. 9, pp. 1089–1093, 2002.

[33] M. K. Hinders and J. R. Hou, "Ultrasonic periodontal probing based on the dynamic wavelet fingerprint," in 31^{st} *Review of Progress in Quantitative Nondestructive Evaluation 24b*, D. O. Thompson and D. E. Chimenti, Eds. Melville, New York: AIP Conference Proceedings, 2005, pp. 1549–1556.

[34] J. R. Hou, S. T. Rose, and M. K. Hinders, "Ultrasonic periodontal probing based on the dynamic wavelet fingerprint," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 1137–1146, 2005.

[35] K. E. Rudd, "Parallel 3d acoustic and elastic wave simulation methods with applications in nondestructive evaluation," Ph.D. dissertation, The College of William and Mary, June 2007.

[36] K. Rudd, C. Bertoncini, and M. Hinders, "Simulations of ultrasonographic periodontal probe using the finite integration technique," *Open Acoustics*, vol. 2, pp. 1–19, 2009.

[37] D. H. Schoen and M. Dean, *Contemporary Periodontal Instrumentation.* Philadelphia: W B Saunders, 1996.

[38] K. H. Rateitschak, E. M. Rateitschak, H. F. Wolf, and T. M. Hassell, "Periodontology," in *Color Atlas of Dental Medicine*, K. H. Rateitschak, Ed., Thieme Medical Publishers, 1989.

[39] P. M. Morse and K. U. Ingard, *Theoretical Acoustics.* Princeton, NJ: Princeton University Press, 1968.

[40] M. O. Culjat, R. S. Singh, E. R. Brown, R. R. Neurgaonkar, D. C. Yoon, and S. N. White, "Ultrasound crack detection in a simulated tooth," *Dentomaxillofacial Radiology*, vol. 34, pp. 80–85, 2005.

[41] F. A. Duck, *Physical properties of tissue.* London: Academic Press, 1990.

[42] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision.* Boston, MA: Addison-Wesley, 1992, vol. I.

[43] B. K. P. Horn, *Robot Vision.* Cambridge, MA: MIT Press, 1986.

[44] R. P. W. Duin, *PRTools Version 3.0: A MATLAB Toolbox for Pattern Recognition*, Delft University of Technology, 2000.

[45] J. C. Platt, N. Cristianini, and J. Shawe-taylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems.* MIT Press, 2000, pp. 547–553.

[46] L. I. Kuncheva, *Combining pattern classifiers methods.* New York: Wiley, 2004.

[47] D. G. Bassani, L. A. Miranda, and A. Gustafsson, "Use of the limits of agreement approach in periodontology," *Oral Health & Preventive Dentistry*, vol. 5, no. 2, pp. 119–24, 2007.

[48] N. Ahmed, T. L. P. Watts, and R. F. Wilson, "An investigation of the validity of attachment level measurements with an automated periodontal probe," *Journal of Clinical Periodontology*, vol. 23, pp. 452–455, 1996.

[49] M. C. K. Yang, R. G. Marks, I. Magnusson, B. Clouser, and W. B. Clark, "Reproducibility of an electronic probe in relative attachment level measurements," *Journal of Clinical Periodontology*, vol. 19, pp. 306–311, 1992.

[50] U. Velden, "Probing force and the relationship of the probe tip to the periodontal tissues," *Journal of Clinical Periodontology*, vol. 6, no. 2, pp. 106–114, 1979.

[51] T. E. Rams and J. Slots, "Comparison of two pressure-sensitive periodontal probes and a manual probe in shallow and deep pockets," *International Journal of Periodontics & Restorative Dentistry*, vol. 13, pp. 521–529.

[52] L. Mayfield, G. Bratthall, and R. Attström, "Periodontal probe precision using 4 different periodontal probes," *Journal of Clinical Periodontology*, vol. 23, pp. 76–82, 1996.

[53] L. Tupta-Veselicky, P. Famili, F. J. Ceravolo, and T. Zullo, "A clinical study of an electronic constant force periodontal probe," *Journal of Periodontology*, vol. 65, pp. 616–622, 1994.

Chapter 4

Classification of Lamb Wave Tomographic Ray Paths in Pipes

Lamb waves are guided plate waves and are often used in structural health monitoring because the unique modal behavior of Lamb waves results in different arrival times for modes traveling through flaws. One application of Lamb waves for structural health monitoring is a motorized scan over the surface of the sample resulting in a tomographic image that accurately depicts the presence of flaws in plates and plate-like objects like pipes. However, these images cannot predict the type or severity of the flaw, which are useful in order to decide a course of correction for the structure. The research in this paper describes the incorporation of pattern classification techniques to Lamb wave tomographic scans of pipes in order to predict flaw type. The features used were extracted using image recognition techniques on Dynamic Wavelet Fingerprint (DWFP) transforms of the unfiltered ray paths. Features were selected at the anticipated mode arrival times for tomographic scans of a pipe with two different flaw types. The application of Support Vector Machines (SVM) in a multi-class one-versus-one method resulted in a direct comparison of predicted and known flaw types.

4.1 Introduction

Structural health monitoring involves the application of sensor technologies to identify and locate defects that develop over time. The addition of pattern recognition techniques to structural health monitoring may help to minimize false positives and false negatives [1]. Recently, several attempts have been made to uncover features that may be able to distinguish between damaged and undamaged samples for specific applications. These features include time series analysis [2–4], energy [5, 6], Fourier transforms [7], wavelet energy [8, 9], novelty detection [10, 11], and principal component expansion of plasma spectra [12]. Many of these are *in situ* experiments using classifiers that include discriminants, k-nearest-neighbor, support vector machines, and neural networks. The classifiers are usually able to discriminate between specific flaws in the structure. However, these sensing techniques are one-dimensional, so that a single interrogation of a region of the structure is associated with a single decision of whether or not a flaw exists in that region.

Tomography, on the other hand, is two-dimensional, which means the sensors gather data from multiple directions over an area of the structure. Lamb wave tomography has previously been used to detect flaws in plates and locally plate-like structures such as pipes [13–17]. In the process of Helical Ultrasound Tomography (HUT), two parallel transducer arrays are installed around the perimeter of the pipe and guided waves travel between every pair of transducers in the arrays [18]. In the laboratory, the two arrays of transducers are approximated by two single transducers that are moved by motors around the perimeter of the area being scanned. The result of the HUT scan is an array of recorded ultrasonic waveforms for each position of the pair of transducers. For all the possible positions of the transmitting and receiving transducers, the result of the tomographic reconstruction is a 2D image where each pixel relates to the local slowness of that ray path. In this way, flaws in the sample under study are localized. However, tomographic reconstructions are not always possible because producing accurate images requires many ray paths and access to the structure in question may be limited. In addition, the tomographic reconstructions cannot always predict the cause or the type of flaw.

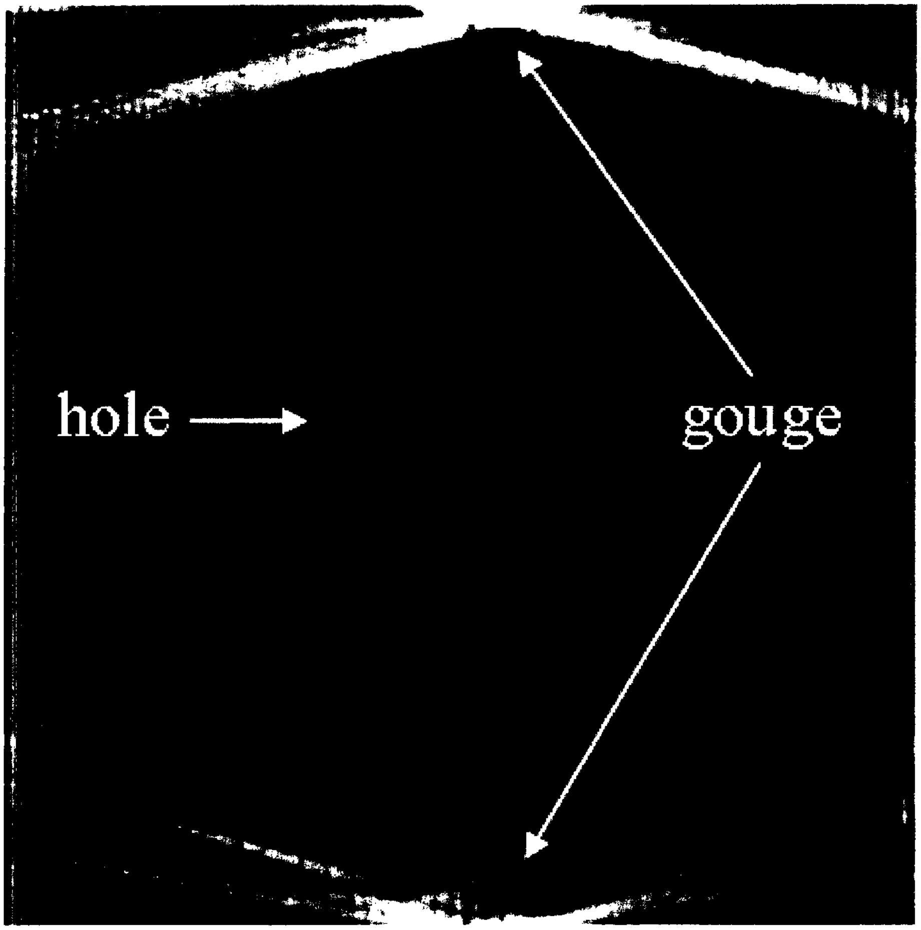


Figure 4.1: A tomographic reconstruction of the pipe under study. The gouge flaw shows at the top of the 2D image, while the through-hole is visible in the middle of the image.

The research in this chapter involves applying pattern recognition techniques to Lamb waves generated from a tomographic scan of an aluminum pipe. The application of pattern classification to tomography will be able to identify the source of flaws as well as their location using a limited number of ray paths. Figure 4.1 shows a tomographic reconstruction of the pipe used in this study. This pipe had two flaws, an internal gouge flaw where the transducers began scanning and appears at the top of the image, as well as a through-hole that appears in the middle of the image. The hole flaw could be misinterpreted as a circular artefact due to the tomographic reconstruction process. These artifacts are common [19]. The addition of pattern classification to this technology will be able to correctly identify the hole flaw, reducing the risk of identifying artifacts as flaws.

There are several instances of pattern recognition studies of pipes in the literature [20–22]. Ravanbod [20] used ultrasonic pipeline inspection gauges (PIGs) commonly used for inspection of oil pipelines. Neural network decision was made using fuzzy logic to soften the decision threshold in simulated and real data with features selected to accurately detect the location and type of external and internal flaws, in addition to producing an image of the predicted flaw. Zhao *et al.* [21] used electromagnetic transducers mounted on a PIG to inspect several steel pipes with artifical flaws introduced. Features were extracted through time domain correlation analysis, selected with principal component analysis, and classified using discriminant analysis. Flaws of two different types and at least 2.5 mm deep were successfully identified with this method. Lastly, Sun *et al.* [22] applied fuzzy pattern recognition to detect weld defects in steel pipes using x-rays. However, to our knowledge there are no instances of applying pattern classification to helical ultrasound tomography in pipes.

4.2 Theory

The structural acoustics of a pipe is commonly regarded as corresponding to Lamb waves in an unrolled plate [23]. Therefore, the theory of Lamb waves propagating in a pipe can analogously be developed by examining Lamb waves in plates.

Lamb waves occur in a plate or other solid layer with free boundaries in which the elastic waves propagate both perpendicularly and within the plane of the plate [24]. The solution to the free plate problem yields the Rayleigh-Lamb frequency equations, which can be written [25]

$$\frac{\tan(qh)}{\tan(ph)} = -\frac{4k^2pq}{(q^2-k^2)^2} \tag{4.1}$$

$$\frac{\tan(qh)}{\tan(ph)} = -\frac{(q^2-k^2)^2}{4k^2pq} \tag{4.2}$$

where $k = \omega/c_p$ is the wavenumber for the phase velocity of the Lamb wave mode c_p, and ω is the circular frequency. The variables p and q are shorthand for

$$p^2 = (\tfrac{\omega}{c_L})^2 - k^2 \quad \text{and} \quad q^2 = (\tfrac{\omega}{c_T})^2 - k^2$$

The Rayleigh-Lamb equations give solutions for the symmetric (Eqn. 4.1) and antisymmetric (Eqn. 4.2) modes of the propagating wave. The physical interpretation of the symmetric and antisymmetric modes (Fig. 4.2) implies that, as the Lamb wave travels along the plate, the top and bottom of the plate either move in concert (antisymmetric mode), or in opposition to each other (symmetric mode). Numerically, the Rayleigh-Lamb equations are solved for a given material to yield the phase velocity c_p, and the group velocity can be found by

$$c_g = c_p^2\{c_p - (fd)\frac{\mathrm{d}c_p}{\mathrm{d}(fd)}\} \tag{4.3}$$

Plots of c_p and c_g for the first two symmetric ($S0, S1$) and antisymmetric ($A0, A1$) modes are shown in Fig. 4.3. The material used in this plot was aluminum, in which $c_T = 6.32$ mm/μs and $c_L = 3.13$ mm/μs were the transverse and longitudinal wave speeds used.

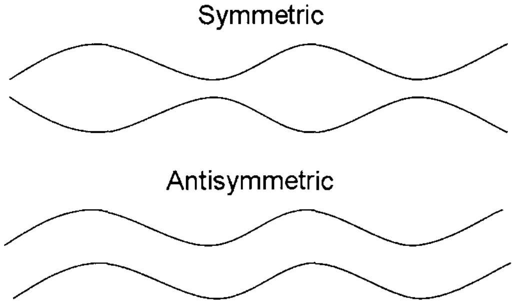


Figure 4.2: The symmetric and antisymmetric modes of the Lamb wave are shown in this sketch.

Lamb waves are commonly used in nondestructive evaluation (NDE) inspections. As the Lamb waves propagate along the plate, if the plate has a flaw, such as a gouge or corrosion, then the thickness of the plate is reduced at that point. Since the thickness changes, the frequency-thickness product $f \cdot d$ in Fig. 4.3 changes, resulting in a different arrival time of the mode. Depending on the frequency of the ultrasonic transducer used as well as the material and thickness of the plate, it is often sufficient to detect flaws by measuring the arrival time of the mode. If the experimentally-detected arrival time of the mode differs from its anticipated value, a flaw can be detected. Changing the transducer frequency can allow for improved detection capabilities since different frequency-thickness products $f \cdot d$ change the expected arrival times at different rates.

The application of pattern classification to tomographic Lamb wave scans of a pipe includes the following steps:

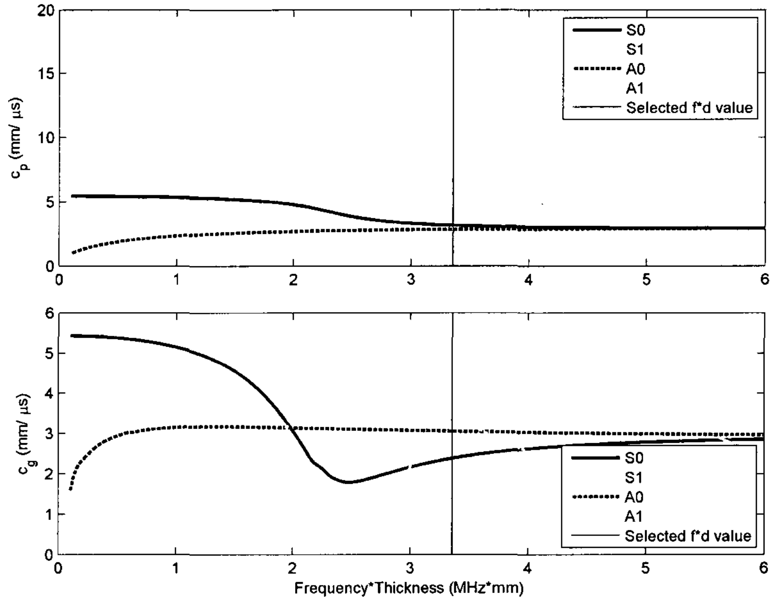


Figure 4.3: The solution to the Rayleigh-Lamb equations for the first two symmetric and antisymmetric modes in an aluminum plate are plotted here. Frequency-thickness product $(f \cdot d)$ is plotted on the abscissa versus phase and group velocity, respectively. The value of $f \cdot d$ used in the experiments is indicated with a line.

1. Sensing: Helical ultrasound tomography is applied to an aluminum pipe with known flaws, resulting in a 2D array of recorded Lamb waves.

2. Feature extraction: An integral transformation, the Dynamic Wavelet Fingerprint (DWFP), is computed on the waveforms. The transformation maps the 1D waveform to a 2D binary image. Image recognition techniques measure properties of those fingerprints over time.

3. Feature selection: Fingerprint properties are chosen at the anticipated mode arrival times predicted by Fig. 4.10 for tomographic scans performed at different transducer frequencies.

4. Classification: The data set is split via a resampling algorithm into training and testing subsets. The classifier is trained on the training set and tested with the testing set.

5. Decision: the predicted class of the testing set is finalized and the performance of the classifier is evaluated.

Formally [26], consider a two-class problem with labels ω_1, ω_2 with probabilities of each class occuring given by $p(\omega_1), p(\omega_2)$. Now consider a feature vector $\mathbf{x}$, which is the vector of measurements made by the sensing apparatus for one ray path. Then $\mathbf{x}$ is assigned to class ω_j whenever

$$p(\omega_j|\mathbf{x}) > p(\omega_k|\mathbf{x}), k \neq j \tag{4.4}$$

By using Bayes theorem, we can rewrite Eq. 4.4 as

$$p(\mathbf{x}|\omega_j)p(\omega_j) > p(\mathbf{x}|\omega_k)p(\omega_k), k \neq j \tag{4.5}$$

In this way, the sensed object associated with the feature vector $\mathbf{x}$ is assigned to the class ω_j with the highest likelihood. In practice, we have several feature vectors $\mathbf{x}_i$, each with an associated class label w_i taking on the value of one of the ω_j.

Classification generally involves calculating those posterior probabilities $p(\omega_i|\mathbf{x})$ using some mapping. If N represents the number of objects to be classified and M is the number of features in the feature vector $\mathbf{x}$, then we will perform pattern classification on the features $\mathbf{x}_i$ that have an associated array of class labels w_i that take on values $\omega_1, \omega_2, \omega_3$ for $i = 1, \ldots, N$. The most useful classifier for this data set was Support Vector Machines (SVM) [27].

4.3 Method

4.3.1 Apparatus

In this chapter, we perform pattern classification of the ray paths of a Lamb wave tomographic scan of a pipe with known flaws. The experimental apparatus is shown in Fig 4.4. In operating the scanner, the transmitting transducer remains fixed while the receiving transducer steps by motor along the circumference of the pipe until it returns to the starting position. The transmitting transducer indexes by one unit step, and the process repeats. This process is illustrated in Figure 4.5 and gives a data set with helical criss-cross propagation paths which allows for mode-slowness reconstruction via helical ultrasound tomography.

In this study, tomographic scans of a single aluminum pipe were used (Fig. 4.1). The pipe was 4 mm thick with a circumference of 19 inches, and the transducers were placed 12.25 in apart. The sampling frequency used was 25 MHz. Tapered delay lines were used between the transducer face and the pipe surface. Two different kinds of flaws were introduced into the pipe: a shallow, interior-surface gouge flaw approx-

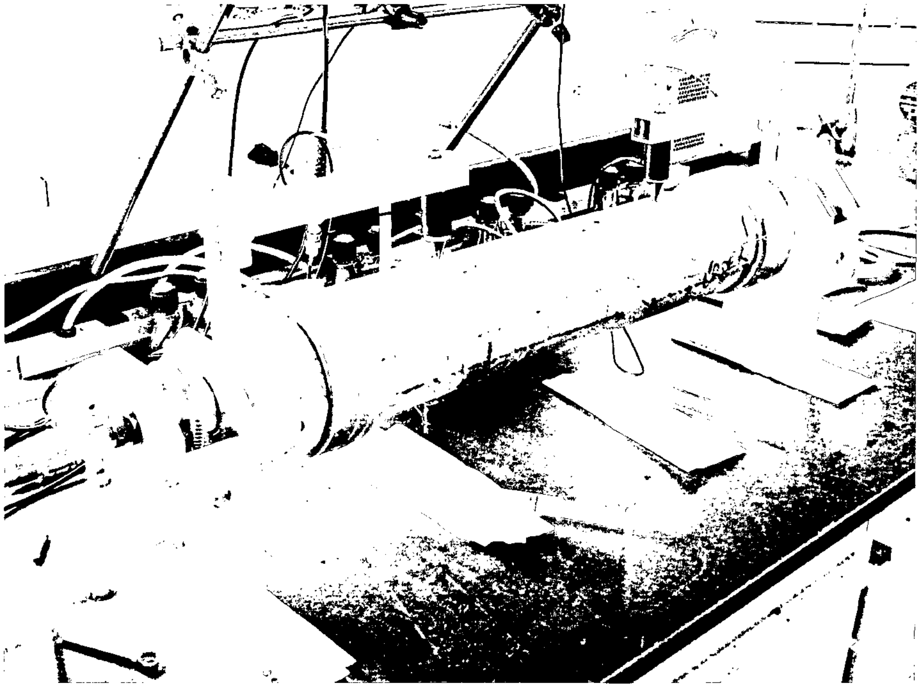


Figure 4.4: The experimental apparatus, in which motors drive a transmitting and receiving transducer around the perimeter of the pipe.

imately 3.5 cm in diameter, and a through-hole 3/8 in in diameter. The positions of the flaws introduced to the pipe can be seen in Figure 4.8, where the geometry of the figure corresponds to the unrolled pipe. There were 180 steps of each of the transducers used. Multiple scans of the pipe were performed while the frequency of the transducer ranged from 0.8 – 0.89 MHz in units of 0.1 Mhz. For classification, only three of these frequencies were selected: 0.8 Mhz, 0.84 MHz, and 0.89 MHz. The pipe was scanned under these conditions, then the hole was increased to 1/2 in diameter, and the process was repeated.

4.3.2 Ray Path Selection

In order to perform classification on the ray paths involved, we first calculated which transducer and receiver index positions, called i_1 and i_2 respectively, correspond to

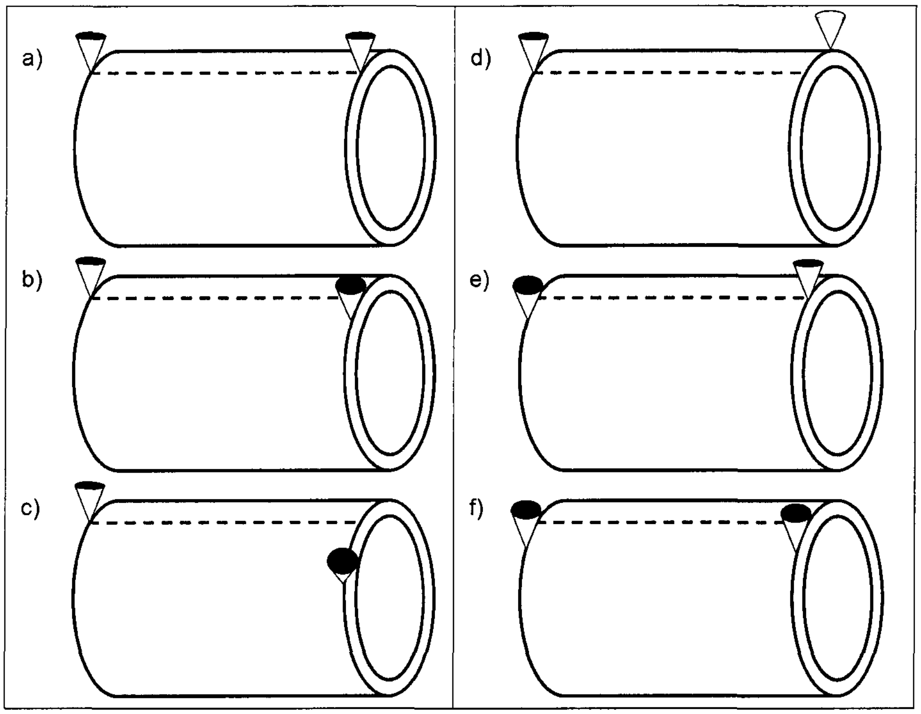


Figure 4.5: The scanning procedure for pipe tomography, in which a) the transducers begin at the same point, and b)-c) the receiver transducer indexes around the pipe until d) it almost returns to the origin. Then d) the transmitting transducer indexes by one as the receiving transducer reaches the origin and e) the receiving transducer continues to step around the perimeter of the pipe.

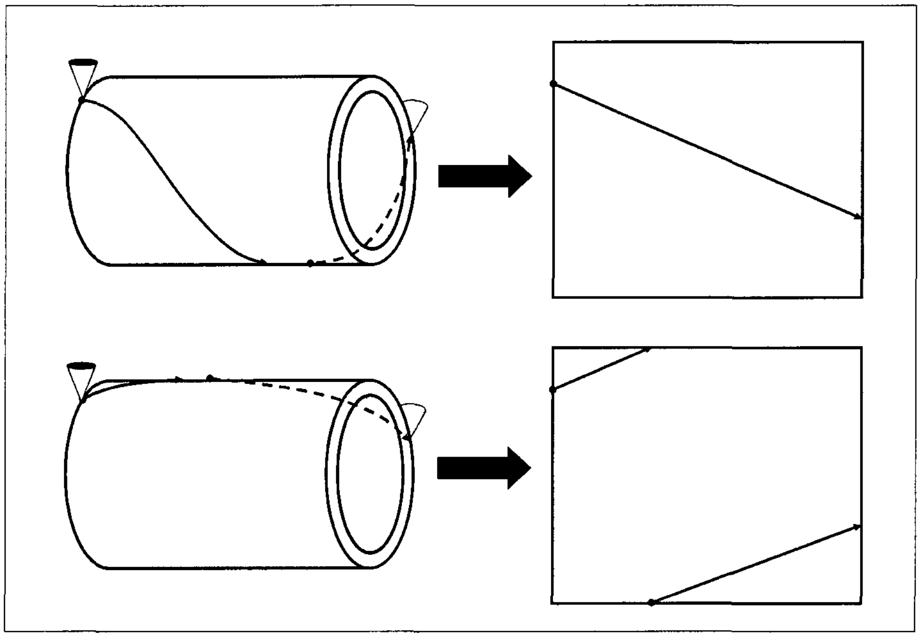


Figure 4.6: The figure demonstrates how different transmitting and receiving transducer positions can wrap around the 2D geometry. In a), the transducers are close enough so that no wrapping occurs, whereas in b) the ray path wraps around the 2D boundary.

a ray path intersecting one of the flaws. To this end, we mapped the physical specifications of the pipe to that of a plate, with the positions of the transmitting and receiving transducers on the left- and right-hand sides. Figure 4.6 demonstrates the unwrapping of the pipe into 2D geometry and shows that some transmitting and receiving transducers result in a ray path that wraps around the boundary of the 2D space.

The equation for the distance the Lamb wave travels on the surface of the pipe can be derived from the geometry shown in Figure 4.7 and is given by [16]

$$\begin{aligned} s &= \sqrt{L^2 + a^2} \\ &= \sqrt{L^2 + \gamma^2 r^2} \end{aligned} \tag{4.6}$$

where L is the axial distance between the transducers, a is the arc length distended by the axial and actual distance between the transducers, and r is the radius of the pipe. The variable γ is the smallest angle between the transducers,

$$\gamma = \min\{(\phi_1 - \phi_2 + 2\pi), |\phi_1 - \phi_2|, (\phi_1 - \phi_2 - 2\pi)\}$$

where ϕ_1 and ϕ_2 are the respective angles of the transducers.

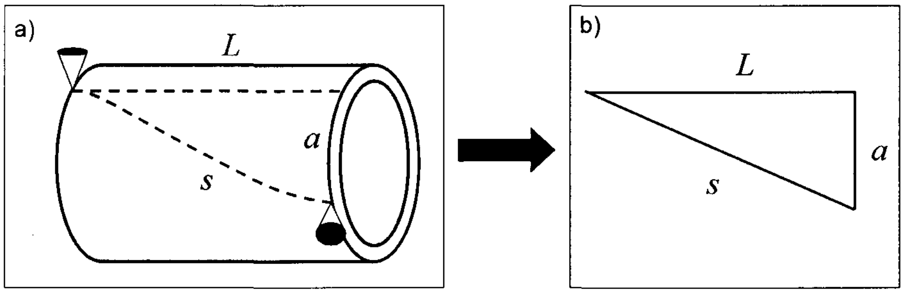


Figure 4.7: The derivation of the equation for the distance a ray path travels on the surface of a pipe is shown, where L is the axial distance between the transducer, s is the distance traveled by the ray path, and a is the arc length distended by the axial and actual distance between the transducers.

The transmitting and receiving transducers have indices represented by i_1 and i_2 respectively, so that $i_1, i_2 = 1, \ldots, 180$. Then if L is the distance between the transducers, and Y is the circumference of the pipe, both measured in centimeters, then the abscissa position of the transducers are $x_1 = 0$ and $x_2 = L$, and the radius of the pipe is given by $r = Y/(2\pi)$. We can convert the indices to angles using the fact that there are 180 transducer positions in one full rotation around the pipe. This gives

$$\phi_1 = i_1 \left(\frac{2\pi}{180} \right)$$

and similarly for ϕ_2. Substituting these into the expression for γ, the minimum angle is

$$\gamma = \left(\frac{2\pi}{180} \right) \min \{ (i_1 - i_2 + 180), |i_1 - i_2|, (i_1 - i_2 - 180) \} \tag{4.7}$$

and the axial distance between the transducers is already given as $f_z = L$. Then substitution into Eqn. 4.6 yields the helical ray path distance between the transducers.

The three different cases considered in finding the minimum angle γ (Eqn. 4.7) are also important in order to determine which ray paths wrap across the 2D simulation space, as shown in Figure 4.6, so that the ordinate coordinates y_1 and y_2 can be calculated. Table 4.1 delineates the different cases that determine γ as the minimum angle and therefore also determine whether or not the ray path wraps across the 2D boundary.

Table 4.1: The formulas for calculating the ordinate coordinates y_1 and y_2 for different transducer indices i_1 and i_2 are given below

$\min(\gamma)$	y_1	y_2
$i_1 - i_2 + 180$	$(i_1 + 180)(Y/180)$	$i_2(Y/180)$
$i_1 - i_2 - 180$	$i_1(Y/180)$	$(i_2 + 180)(Y/180)$
$\|i_1 - i_2\|$	$i_1(Y/180)$	$i_2(Y/180)$

The positions of the flaws were added to the simulation space, and ray paths were drawn between the transducers using coordinates (x_1, y_1) and (x_2, y_2) that depend on i_1 and i_2, as described above. If the ray path intersected any of the flaws, that ray path was recorded to have a class label corresponding to that type of flaw. The labels included no flaw encountered $(\omega_i = 1)$, gouge flaw $(\omega_i = 2)$, and hole flaw $(\omega_i = 3)$. An example of a ray path intersection is shown in Figure 4.8. This example shows the ray path at indices $(i_1, i_2) = (45, 135)$ intersecting the hole flaw, so $w_i = \omega_3$ here. As the figure shows, the flaws were roughly octagonal in shape. In addition, the ray paths were approximated as lines with a width determined by the smallest pixel size, which is $0.1 \cdot (Y/180) = 0.2681$ mm. In reality, Lamb waves have some horizontal spread. These aspects of the ray path simulation may result in some mislabeled ray paths.

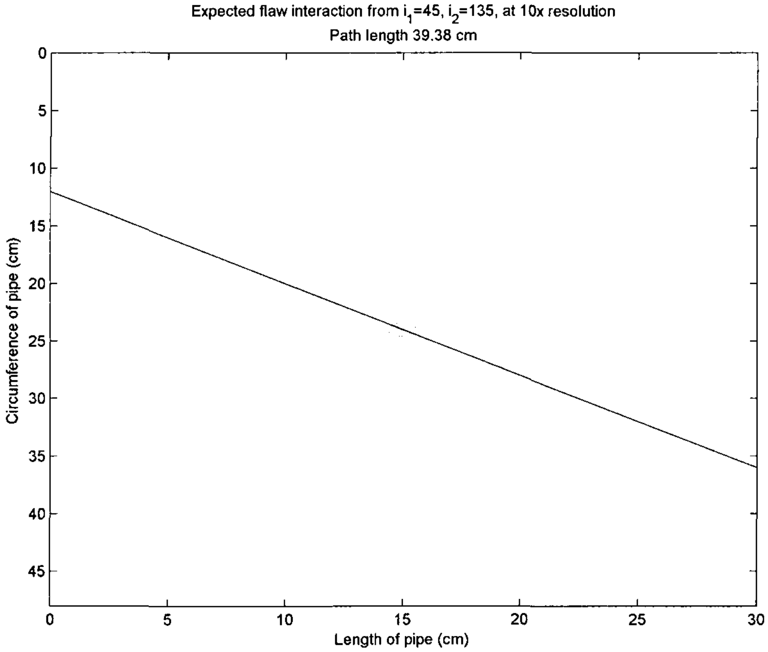


Figure 4.8: An example of the determination of which positions of the transducer (i_1 and i_2) would intersect one of the known flaws in the pipe.

4.4 Classification

As already mentioned, classification will be performed on ray paths with a limitation on the distance between the transducers. In preliminary tests, it was noted that when all the distances were considered, many of the ray paths that actually intersected the hole flaw were labeled as intersecting no flaws. The explanation is that Lamb waves tend to scatter from hole flaws, which means no features will indicate a reduction in thickness of the plate, but the distance traveled will be longer. Therefore, limiting the classification on ray paths of a certain distance improves the classification by reducing the influence of scattering effects.

4.4.1 Feature Extraction

Let D represent the distance limit. The next step of classification is to find M-many features for feature vectors $\mathbf{x}_i$ such that $s_i \leq D$, $i = 1, .., N$. Feature extraction used Dynamic Wavelet Fingerprinting (DWFP), while feature selection involved either selecting points at the relevant mode arrival times for a tomographic scan using a single transducer frequency, or selecting points at only one or two mode arrival times for three different transducer frequencies at once.

DWFP

The DWFP technique (Fig. 4.9) applies a wavelet transform on the original time domain waveform, which results in "loop" features that resemble fingerprints. It has shown promise for a variety of applications.

The Lamb wave tomographic waveforms were fingerprinted using the DWFP algorithm without any preprocessing or filtering. Let $\phi_i(t)$ represent a waveform selected from a Lamb wave tomographic scan ($i = 1, \ldots, N$). The first step of the DWFP

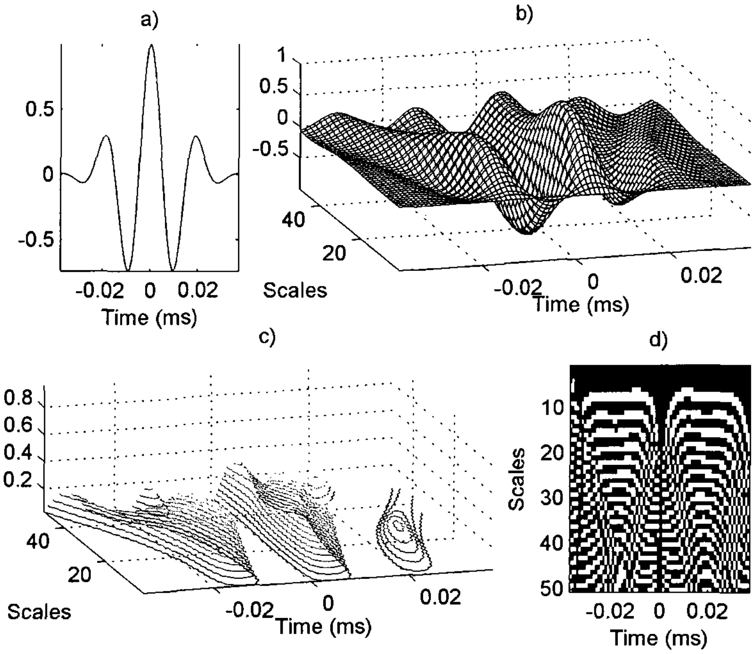


Figure 4.9: The DWFP technique [28] begins with a) the ultrasonic signal, where it generates b) wavelet coefficients indexed by time and scale, where scale is related to frequency. Then c) the coefficients are sliced and projected onto the time-scale plane (d). The final binary image is used to select features for the pattern classification algorithm.

(Fig. 4.9a-b) involves applying a wavelet transform on each of the waveforms. The continuous wavelet transform can be written

$$C(a,b) = \int_{-\infty}^{+\infty} \phi(t)\overline{\psi_{a,b}(t)}dt \tag{4.8}$$

Here, $\phi(t)$ represents a square-integrable 1D function, and $\psi(t)$ represents the mother wavelet. The mother wavelet is scaled in frequency (f) and transformed in time (t) using $a, b \in \mathbb{R}$, respectively, in order to form the $\psi_{a,b}(t)$ in Equation 4.8. The wavelet transform on a single waveform (Fig. 4.9a) results in wavelet coefficients (Fig. 4.9b). Then, a slicing algorithm is applied to create an image analogous to the gradient of the wavelet coefficients in the time-scale plane, resulting in a binary image, $I(a,b)$:

$$\phi(t) \overset{DWFP(\psi_{a,b})}{\longrightarrow} I(a,b) \tag{4.9}$$

We used mother wavelets that previously showed promise for other applications, including Debauchies 3 (db3) and Symelet 5 (sym5). The resulting image I contains fingerprint-like binary contours of the initial waveform $\phi_i(t)$.

The next step is to apply image processing routines to collect properties from each fingerprint object in each waveform. First, the binary image I is labeled with the 8-connected objects, allowing each individual fingerprint in I to be recognized as a separate object using the procedure in Haralick and Shapiro [29]. Next, properties are measured from each fingerprint. Some of these properties include counting the on- and off-pixels in the region, but many involve finding an ellipse matching the second moments of the fingerprint and measuring properties of that ellipse, such as eccentricity. In addition to the orientation measure provided by the ellipse, another measurement of inclination relative to the horizontal axis was determined by Horn's method for a continuous 2D object [30]. Lastly, further properties were measured by determining the boundary of the fingerprint and fitting 2nd or 4th order polynomials.

The image processing routines result in fingerprint properties $F_{i,\nu}[t]$ relative to the original waveform $\phi_i(t)$, where ν represents an index of the image processing-extracted fingerprint properties ($\nu = 1, \ldots, 17$). These properties are discrete in time because the values of the properties are matched to the time value of the fingerprint's center of mass. Linear interpolation yields a smoothed array of property values, $F_{i,\nu}(t)$.

4.4.2 Feature Selection

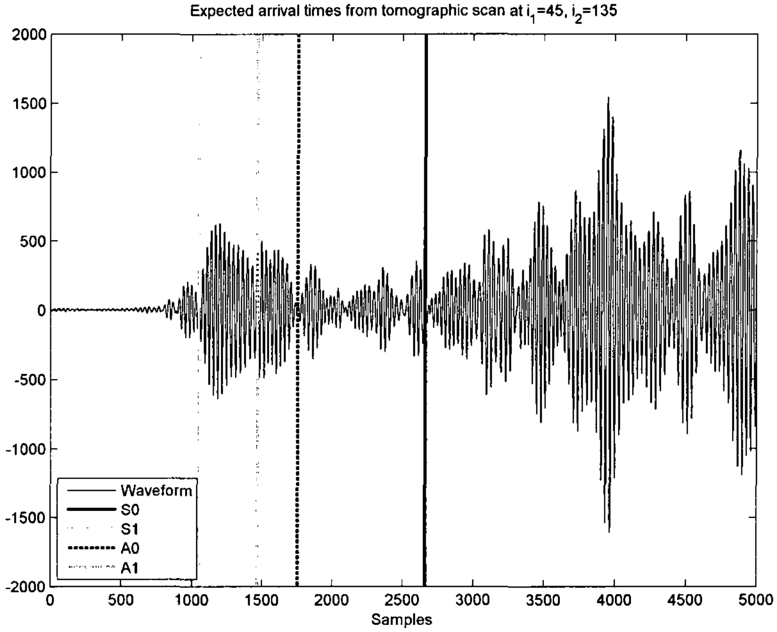


Figure 4.10: A sample waveform for a single pair of transducer positions $(i_1, i_2) = (45, 135)$ is shown here along with the predicted mode arrival times.

The DWFP algorithm results in a 2D array of fingerprint features for each waveform, while only a 1D array of features can be used for classification. For each waveform $\phi_i(t)$, we need to find M-many features $x_{i,j}$, $j \leq M$ from the 2D array of wavelet fingerprint features $F_{i,\nu}(t)$. In this case, the DWFP features selected were wavelet fingerprint features that occurred at the predicted mode arrival times for all fingerprint features under study. At the frequency-thickness product used, there were four modes available: $S0$, $S1$, $A0$, and $A1$. However, as Fig. 4.10 shows, the $S1$ arrival time often occurs early in the signal, and there may not always be DWFP fingerprint feature available. In addition, because there were several different transducer frequencies studied, in order that the number of features remained manageable, there were two different feature selection schemes used:

1. All modes: All four mode arrival times for all 17 fingerprint properties from both mother wavelets are used, but only one transducer frequency is studied from the range $\{0.8, 0.84, 0.89\}$ MHz. There are $M = 136$ features selected.

2. All frequencies: One or two mode arrival times for all 17 fingerprint properties from both mother wavelets are used for all three frequencies at once. The modes used include $S0, A0, A1$, in which case there are $M = 102$ features used. There were also combinations of $S0\&A0$, $S0\&A1$, and $A0\&A1$ mode arrival times used for all properties, frequencies, and mother wavelets, in which case there were $M = 204$ features selected.

The class imbalance problem must also be considered for this data set. Figure 4.11a shows the natural class distribution for the full tomographic scan. Note that the great majority of ray paths intersect with no flaws. In this case, only 3% of the data intersects with the hole flaw, and 10.5% intersects with the gouge flaw. These are poor statistics to build a classifier. Instead, ray paths were randomly selected from the no flaw cases to be included for classification so that

$$\frac{|\omega_1|}{|\omega_2|} = 2$$

Figure 4.11b shows the resulting class distributions used for classification. In this case, now 9% of the ray paths intersect with the hole flaw and 30% intersect with the gouge flaw, so that the number of ray paths used for classification is reduced from $N = 32,400$ to $N = 11,274$ for all ray path distances. One advantage of limiting the ω_1 cases is so that classification can proceed more rapidly. Randomly selecting the ω_1 cases to be used does not adversely affect the results, and later, the ω_1 ray paths not chosen for classification will be used to test the pipe flaw detection algorithm.

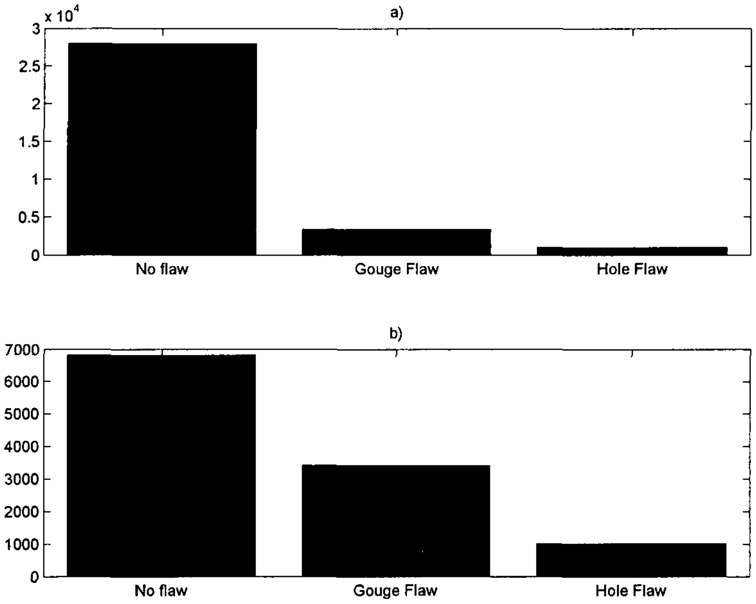


Figure 4.11: a) The class distribution for the full 180 × 180 transduer indices, and b) the class distribution used for classification.

4.4.3 Summary of classification variables

At this point, it may become clear that there were a number of variables involved in the classification design. The list below provides a summary.

1. The same pipe was scanned twice, once when the hole was 3/8 in in diameter,

and again when the hole was enlarged to 1/2 in diameter.

2. The pipe had two different flaws, a gouge flaw (ω_2) and a hole flaw (ω_3). The ray paths that intersected no flaw (ω_1) were also noted.
3. A range of transducer frequencies was used, including $0.8 - 0.89$MHz. For classification, only three of these frequencies were selected: 0.8MHz, 0.84MHz, and 0.89MHz.
4. Classification was restricted by ray paths that had a maximum path length D such that $s \leq D$. There were 91 different path lengths for all transducer positions. Three different values of D were selected to limit the ray paths selected for classification. These correspond to the 10^{th}, 20^{th}, and 91^{st} path length distances, or D_{10} = 31.21 cm, D_{20} = 31.53 cm, and D_{91} = 39.38 cm. The latter case considers all the ray path distances.
5. For the feature extraction, two different wavelets were used (db3 and sym5), and 17 different wavelet fingerprint properties were extracted.
6. Two different feature selection schemes were attempted, varying either the modes selected or the frequencies used.
7. One classifier was selected here (SVM). Other classifiers (such as quadratic discriminant) had lower accuracy.
8. Classification will be performed on training and testing data sets drawn from each individual tomographic scan using the reduced class distribution shown in Fig. 4.11b. The resulting flaw detection algorithm will be tested with the ω_1 ray paths that were excluded from the distribution used for classification.
9. The classification tests on the 1/2 in hole tomographic scan will use the tomographic scan of the 3/8 in hole solely for training the classifier and the 1/2 in hole raypaths solely for testing the classifier. This does not substantially alter the expression of the classification routine in Program 2

4.4.4 Sampling

As already mentioned, the SVM classifier was used on the classifier configured by the options listed above in Section 4.4.3. However, SVM is a binary classifier, and three classes were considered in this study. Therefore, the one-versus-one approach was used, in which pairs of classes are compared at one time for classification, and the remaining class is ignored. The process is repeated until all permutations of the available classes are considered. In this case, classification compared ω_1 vs ω_2, ω_1 vs ω_3, and ω_2 vs ω_3. For each pair of binary classes, the data was randomly sampled via bagging by randomly selecting roughly twice as many samples from the more highly-populated class as the less populated class. The process is repeated until each ray path has been selected several times for training. The results are collapsed by majority rule normalized by the number of samples drawn in the bagging process. Program 2 displays pseudocode representing the sampling algorithm that splits the data into training and testing sets and the means by which the majority vote is decided.

As an example of the majority vote selection process, consider the votes shown in Table 4.2. This shows the confusion matrix as described in Program 2. For example, coordinate $(1, 3) = 0.14$, so 14% of the labels for object $\mathbf{x}_k$ were ω_1 when ω_1 and ω_3 were selected for the training set. Likewise, coordinate $(3, 1) = 0.86$, so 86% of the labels were ω_3 when ω_1 and ω_3 were selected for training. Because there were three unique binary pairs of possible labels, the actual label is found by adding over the columns of $\mathcal{L}$. The predicted label is the value of ω_i (the rows) at which the maximum of the sum over columns occurs. In this case, the predicted label is ω_2. Indeed, this corresponds exactly to the actual class of the ray path in this example. Through this method, we get a one-dimensional array of labels λ_k, $k = 1, \ldots, N$ so that each ray path corresponds to a scalar predicted class.

Program 2 The sampling algorithm used to split the data into training and testing sets for SVM classification is described here. It is similar to bagging.

```
input $\{\mathbf{x}_i\}$, $i = 1, \ldots, N$ feature vectors
      $w_i$ class labels corresponding to each $\mathbf{x}_i$
         which take on values $\{\omega_1, \omega_2, \omega_3\}$
      $\mathcal{C}$ a classifier mapping
FOR EACH unique binary pair $(\omega_i, \omega_j)$ {
      FIND $S_1 = \{\mathbf{x}_l | w_l = \omega_i\}$
           $S_2 = \{\mathbf{x}_l | w_l = \omega_j\}$
           $S_3 = \{\mathbf{x}_l | w_l \neq \omega_{i,j}\}$
      WHILE any $\mathbf{x}_l \in \{S_1, S_2\}$ remains unselected {
           CHOOSE randomly $s_1 \subset S_1$, $s_2 \subset S_2$, $s_3 \subset S_3$
                  such that
                  $|s_1| = 2|s_2|$ whenever $|S_1| > |S_2|$
                  and $|s_3| \leq |s_2|$
           DEFINE $T_S = \{\frac{1}{2}s_1, \frac{1}{2}s_2, s_3\}$ (the testing set)
                  $T_R = \{s1, s2\} - Ts$ (the training set)
           TRAIN $\mathcal{C}$ on $T_R$
           TEST $\mathcal{C}$ on $T_S$
                resulting in labels $\{y_k\}$, $k = T_S$
      }
      SAVE results in confusion matrix for each $\mathbf{x}_k$
           $\mathcal{L}_k(\omega_i, \omega_j) = \frac{|y_k = \omega_i|}{|y_k|}$
           $\mathcal{L}_k(\omega_j, \omega_i) = \frac{|y_k = \omega_j|}{|y_k|}$
}
VOTE on the actual label by summing over the columns
     of the confusion matrix
     $\mathcal{P}_k = \sum_{\omega_j} \mathcal{L}_k(\omega_i, \omega_j)$
The predicted label is the maximum
     $\lambda_k = \omega_i | \max(\mathcal{P}_k(\omega_i))$
```

Table 4.2: An example of the majority voting process through which predicted labels are decided from the confusion matrix $\mathcal{L}$.

	ω_j			
ω_i	1	2	3	sum
1	0	0	0.14	0.14
2	1.00	0	1.00	2.00
3	0.86	0	0	0.86

4.5 Decision

As Program 2 shows, a single configuration of the classifier variables C (described in Section 4.4.3) takes an array of feature vectors $\mathbf{x}_i$, $i = 1, \ldots, N$ and their corresponding class labels $w_i \in \{\omega_1, \omega_2, \omega_3\}$ and produces an array of predicted class labels λ_i after dimensionality reduction. Each index i corresponds to a single ray path between the two transducer indices. We can conclude how well the classifier performs by evaluating its accuracy, which is defined as

$$A(\omega_k) = \frac{|(w_i = \omega_k)\&(\lambda_i = \omega_k)|}{|(w_i = \omega_k)|}, i = 1, \ldots, N; k = 1, 2, 3. \tag{4.10}$$

However, a further step is required in order to determine whether or not there are in fact any flaws present in the pipe scanned by Lamb wave tomography. The predicted labels of the ray paths is not sufficient to decide whether or not there are flaws. Therefore, we will use the ray path drawing algorithm described in Section 4.3.2 to superimpose the ray paths that receive predicted labels λ_i in each class $\omega_1, \omega_2, \omega_3$. If several ray paths intersect on a pixel, their value is added, so the higher the value of the pixel, the more ray path intersections occur at that point. This technique averages out the misclassifications that occur in the predicted class labels. The ray paths for the predicted class labels associated with each flaw type are drawn separately, so that the more ray paths that have been predicted to be associated with a particular flaw intersect in the same region, the more likely a flaw exists at that point.

Figure 4.12 and 4.13 show the method of drawing ray paths for each predicted class label. Both of these examples were configured for the 3/8 in hole and selected the $A0$ mode from all three transducer frequencies. However, Fig. 4.12 classified all ray paths regardless of length, and Fig. 4.13 restricted the ray path distances to D_{20}. The ray path intersections were originally drawn at 10× resolution but the images were later smoothed by averaging over adjacent cells to 1× resolution. The larger the pixel value, the more ray paths intersected in that region, and the more likely a flaw exists at that point. Note that in Fig. 4.13b and 4.13c, the pixel values are higher at the locations where their respective flaws actually exist. Fig. 4.13a also shows more intersections at that point, because of the geometry of the pipe and the way the transducers scanned, those regions of the pipe actually did have more ray path intersections than elsewhere. That's why those regions were selected to introduce flaws into the pipe. But the largest pixel value when the ray paths predicted to intersect no flaws is smaller than the ray paths predicted to intersect either of the other flaws. Also, Fig. 4.13a shows a higher average pixel value, showing that more intersections occur throughout the pipe rather than focused on a smaller region, such as Fig. 4.13b and Fig. 4.13c. However, due to the scattering of the Lamb waves, Fig. 4.12 is not as precise. Fig. 4.12a does show a higher average pixel value, but 4.12b and 4.12c seem to both indicate hole flaws in the middle of the pipe, while neither seem to indicate the correct location of the gouge flaw.

In order to automate the flaw detection process from these ray intersection plots, image recognition routines and thresholds were applied. The process of automatically detecting flaws in the image of ray path intersections (U) include:

1. Apply a threshold h_I to the pixel values of the image. If $|U > h_I| = \emptyset$, then no flaws are detected. Otherwise, define $U' = U(U > h_I)$

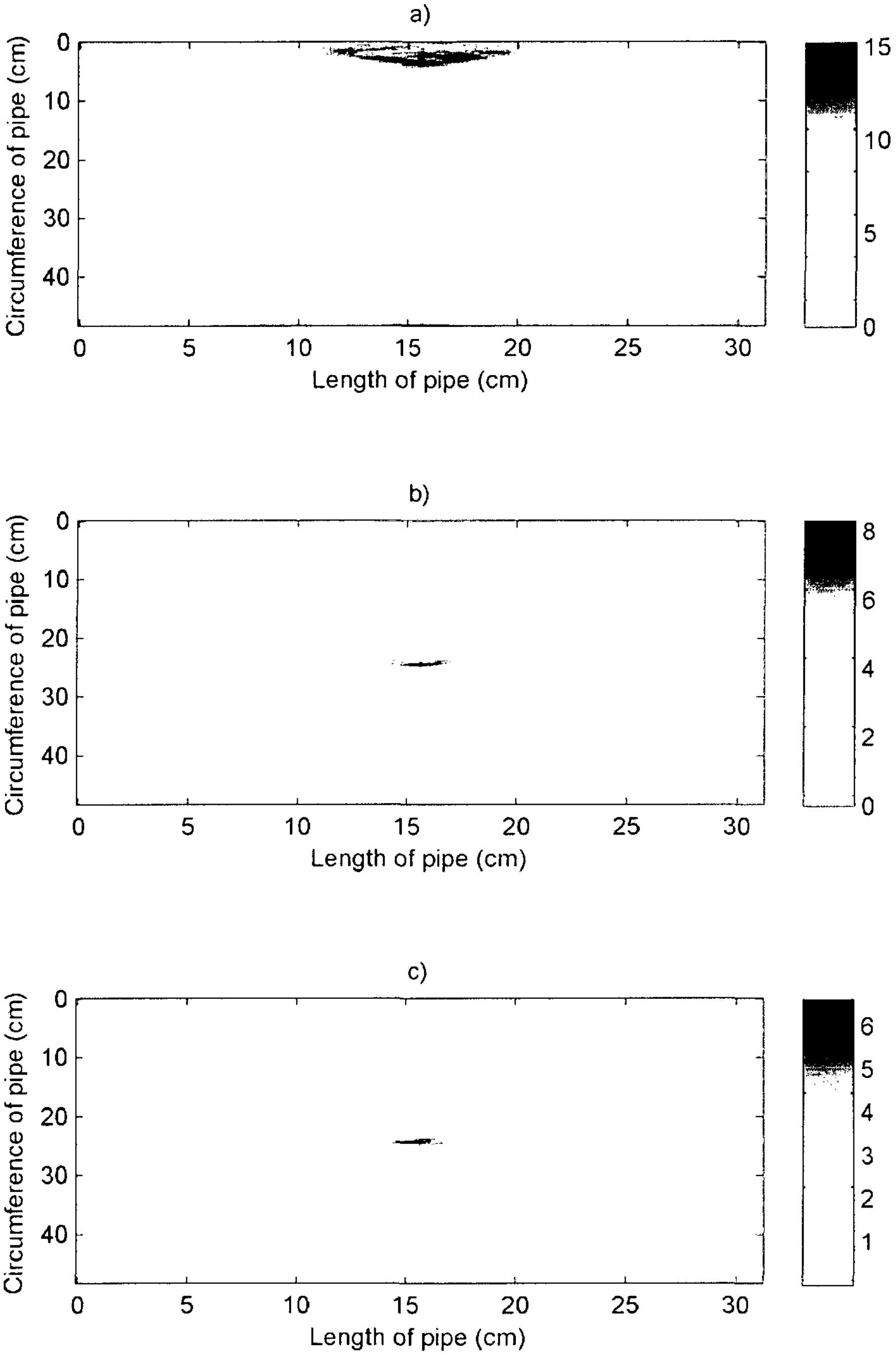


Figure 4.12: The ray paths that have received a predicted class label of a) no flaw, b) gouge flaw, and c) hole flaw are drawn here. The classifier configured here used all the available ray path distances ($D = D_{91}$).

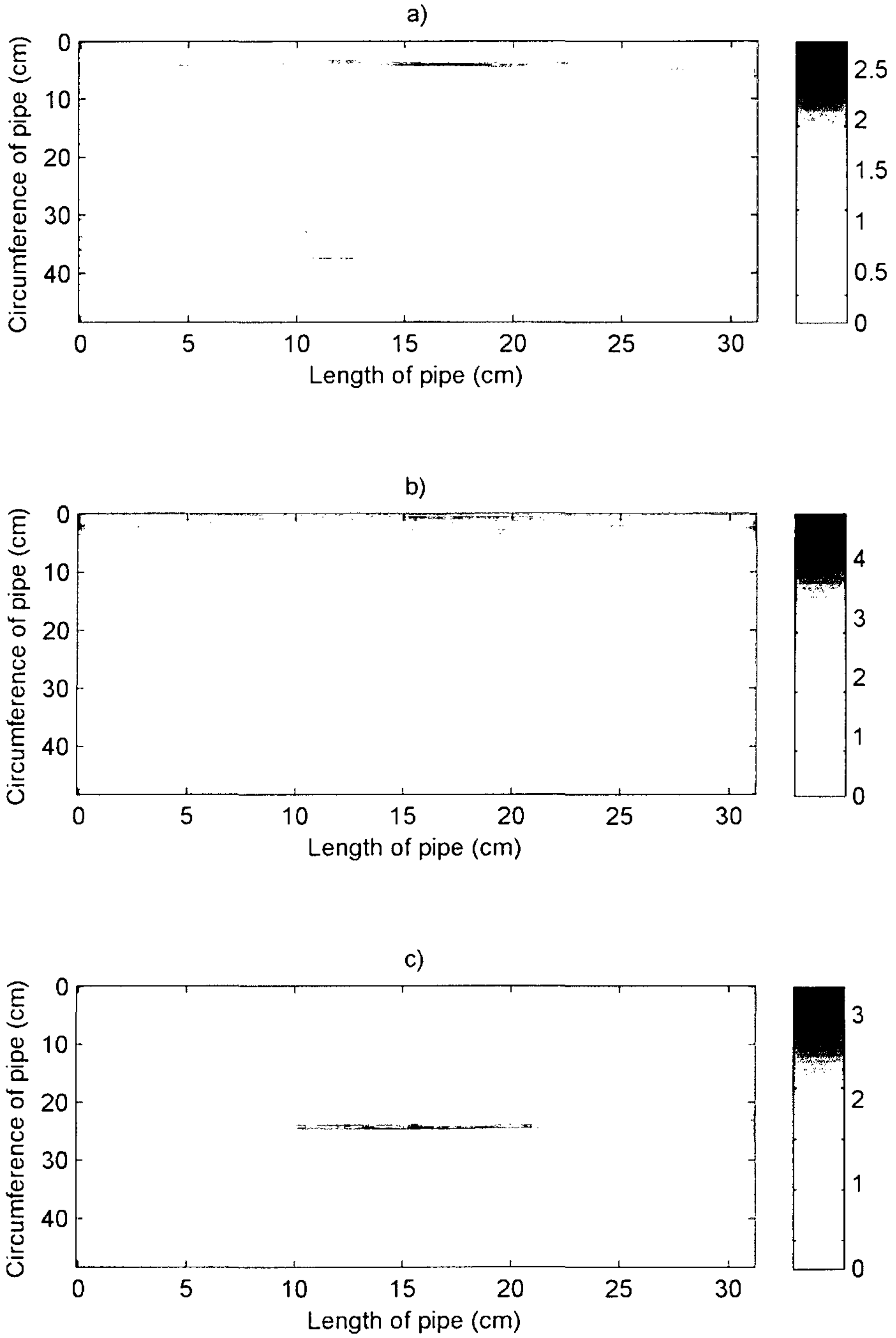


Figure 4.13: The ray paths that have received a predicted class label of a) no flaw, b) gouge flaw, and c) hole flaw are drawn here. Unlike Figure 4.12, the ray path lengths limit used was D_{20}.

2. Check that the size of the nonzero elements of U' are smaller than half its total area

$$\sum_{i1}\sum_{i2} U'(i1,i2) < \frac{1}{2}|U'|$$

3. Apply a threshold h_a to the area of U'. If $\sum_{i1}\sum_{i2} U'(i1,i2) \leq h_a$, then no flaws are detected. Otherwise, decide that U' accurately represents the flaws in the region. Return an image of U'.

This algorithm is only intended to be performed on the ray paths that predicted a flaw location. It does not tend to work well on the ray path intersections that predicted no flaw, since those depend on the geometry of the object being scanned.

Figure 4.14 shows predicted flaw locations relative to the sample ray path intersection images given in Figs. 4.12 and 4.13. Specifically, Figs. 4.14a) and b) give the predicted flaw locations from Figs. 4.12b) and 4.12c), while Figs. 4.14c) and d) give the predicted flaw locations from Figs. 4.13b) and 4.13c). Note that the images produced by the classifier designed to accept all ray path distances (Fig. 4.14a,b) shows a flaw location that is more discrete and closer to the size of the actual flaw, but it is not as accurate at predicting whether or not flaws exist as the classifier designed to accept ray paths for classification restricted by path length (Fig. 4.14c,d).

4.6 Results and Discussion

4.6.1 Accuracy

The classifier variables described in Section 4.4.3 were explored using the classification routine sketched out in Program 2 with predicted labels collapsed by the procedure illustrated in Table 4.2. Before examining the predictions of the pipe flaw detector, we will first examine the accuracy of the classifier itself. Each table of the accuracy

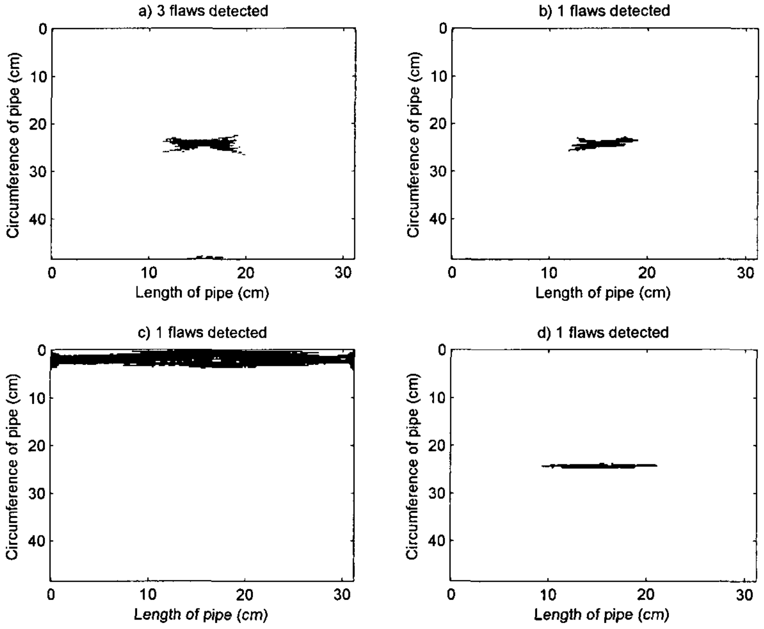


Figure 4.14: The results of the automatic pipe flaw detector routine on the ray path intersection images in Fig 4.12 and 4.13 that predicted flaws are shown here. The approximate location of the flaw is shown in the image. Here, subplots a) and b) show the predicted flaw locations for the gouge and hole flaws respectively when all distances were used in classification. Similarly, subplots c) and d) show the predicted flaw locations for the gouge and hole flaws when the ray path distance was limited to D_{20}

results shows the classifier configuration, including two different tomographic scans of the aluminum pipe with different hole flaw diameters, as well as the frequencies and modes selected for classification. The classifiers were also limited by the maximum distance between the transducers. For ease of comparison, the average accuracy over each class type ω_k is computed in the last column.

Table 4.3 shows the accuracy $A(\omega_k)$ for both tomographic scans of the pipe when only a single transducer frequency was used at a time, and therefore all the available Lamb wave modes were selected. The table is grouped by different maximum distance lengths used in the classifier. Meanwhile, Tables 4.4-4.6 show the classifier accuracy for classifiers in which features were selected for only certain modes but for three different values of the transducer frequency at once.

These classification results show that no classifier configuration had more than 80% average accuracy per class. Many classifiers scored lower than 50% average accuracy. However, the detection routine does not require high accuracy classifiers, and as already mentioned, the diffraction effect of Lamb waves around the hole flaw could certainly explain the lower accuracy of the classifiers for that hole type. Other patterns that emerge in the data show that the smaller values of the maximum path length distance yield higher average accuracy by as much as 20%. In addition, the feature selection strategy that utilizes only select modes for all three transducer frequencies seems to work slightly better than the feature selection routine that selects all modes for only one transducer frequency. Lastly, the tomographic scan of the aluminum pipe with the 3/8 in hole seems to have higher accuracy than the scan of the pipe with the 1/2 in hole. These results make sense given the fact that the 3/8 in hole was trained and tested on unique subsets of the ray paths drawn from the 3/8 in hole tomographic scan, which yields somewhat optimistic results. Meanwhile the results for the 1/2 in hole scan were tested with ray paths drawn from the 1/2 in hole scan but trained with ray paths selected from the 3/8 in hole scan.

4.6.2 Flaw Detection Algorithm

As mentioned in Section 4.5, a more promising way of determining whether or not flaws exist in the pipe is to use the ray path simulation routine described in Section 4.3.2. Because the result of the pipe flaw detection routine is an image (Fig. 4.14), and an exhaustive depiction of the resulting flaw prediction images would take up an unnecessary amount of space, the results will be presented in qualitative format. The pipe detection routine was tested on all of the classifier configurations and a judgment was made about which flaw the routine was predicting.

Figure 4.15 shows some of the different ways that the flaw detector routine can present results. Fig. 4.15a) shows a correctly identified flaw location for the gouge flaw, despite the fact that the flaw detector claims the flaws drawn at the bottom of the plot to be separate. That's because the pipe in reality wraps the 2D plot into a cylinder, so the regions identified as flaws at the bottom of the plot actually can be associated with the correct location of the gouge flaw at the top of the plot. However, Fig. 4.15b) shows an incorrect assignment of an indentation flaw in the center of the plot, where the hole flaw actually resides. Similarly, Fig. 4.15c) shows a correct identification of the hole flaw, despite the fact that the two identified flaws in the middle of the plot do not seem connected. This is an artifact of the drawing algorithm. But Fig. 4.15d) shows a falsely-positive-identified region at the top of the plot predicted to be a hole flaw where the gouge flaw actually exists. This is a failure of specificity rather than sensitivity.

Tables 4.7-4.9 show the performance of the pipe flaw detector routine. All of the different frequencies, modes, and hole diameters are displayed in each table, but Table 4.7 shows the results for a maximum ray path length of $D = D_{10}$ used in the

Table 4.3: The classifier accuracy when the feature selection utilized only one transducer frequency and all modes are shown below. The results are grouped by the maximum ray path distance selected. The average accuracy per class is shown in the last column.

Hole	Frequencies	Modes	Max Path	$A(\omega)$ [%]			
Diameter [in]	Used [MHz]	Used	Distance [cm]	No Flaw (ω_1)	Gouge Flaw (ω_2)	Hole Flaw (ω_3)	Average
3/8	0.80	S0, A0, A1	D_{10}	77.7	62.4	72.4	70.8
3/8	0.84	S0, A0, A1	D_{10}	78.5	67.3	42.9	62.9
3/8	0.89	S0, A0, A1	D_{10}	70.8	77.6	69.5	72.6
1/2	0.80	S0, A0, A1	D_{10}	78.5	43.2	22.8	48.2
1/2	0.84	S0, A0, A1	D_{10}	84.0	10.9	30.9	41.9
1/2	0.89	S0, A0, A1	D_{10}	52.3	62.4	39.8	51.5
3/8	0.80	S0, A0, A1	D_{20}	70.5	50.6	62.8	61.3
3/8	0.84	S0, A0, A1	D_{20}	73.9	59.9	41.9	58.5
3/8	0.89	S0, A0, A1	D_{20}	68.3	70.0	58.1	65.5
1/2	0.80	S0, A0, A1	D_{20}	75.8	47.7	23.3	48.9
1/2	0.84	S0, A0, A1	D_{20}	93.0	21.2	14.6	42.9
1/2	0.89	S0, A0, A1	D_{20}	52.6	64.9	34.4	50.6
3/8	0.80	S0, A0, A1	D_{91}	93.8	14.1	27.4	45.1
3/8	0.84	S0, A0, A1	D_{91}	93.8	14.6	19.8	42.7
3/8	0.89	S0, A0, A1	D_{91}	91.7	17.0	28.2	45.6
1/2	0.80	S0, A0, A1	D_{91}	73.7	42.7	16.1	44.1
1/2	0.84	S0, A0, A1	D_{91}	82.9	35.7	7.6	42.0
1/2	0.89	S0, A0, A1	D_{91}	41.9	61.1	15.4	39.5

Table 4.4: The classifier accuracy when the feature selection used only one or two modes at once but all transducer frequencies are shown here. The maximum ray path distance used here was D_{10}

Hole	Frequencies	Modes	Max Path	$A(\omega)$ [%]			
Diameter [in]	Used [MHz]	Used	Distance [cm]	No Flaw (ω_1)	Gouge Flaw (ω_2)	Hole Flaw (ω_3)	Average
3/8	0.80, 0.84, 0.89	S0	D_{10}	71.5	65.0	61.9	66.1
3/8	0.80, 0.84, 0.89	A0	D_{10}	84.2	72.6	64.8	73.8
3/8	0.80, 0.84, 0.89	A1	D_{10}	75.7	68.0	52.4	65.4
3/8	0.80, 0.84, 0.89	S0, A0	D_{10}	87.5	79.2	73.3	80.0
3/8	0.80, 0.84, 0.89	S0, A1	D_{10}	81.0	74.9	74.3	76.7
3/8	0.80, 0.84, 0.89	A0, A1	D_{10}	85.3	80.9	72.4	79.5
1/2	0.80, 0.84, 0.89	S0	D_{10}	82.8	34.0	16.3	44.4
1/2	0.80, 0.84, 0.89	A0	D_{10}	67.5	46.9	34.1	49.5
1/2	0.80, 0.84, 0.89	A1	D_{10}	67.7	34.7	40.7	47.7
1/2	0.80, 0.84, 0.89	S0, A0	D_{10}	82.3	40.6	29.3	50.7
1/2	0.80, 0.84, 0.89	S0, A1	D_{10}	81.8	38.6	23.6	48.0
1/2	0.80, 0.84, 0.89	A0, A1	D_{10}	74.6	49.8	39.8	54.8

Table 4.5: The classifier accuracy when the feature selection used only one or two modes at once but all transducer frequencies are shown here. The maximum ray path distance used here was D_{20}

Hole	Frequencies	Modes	Max Path	$A(\omega)$ [%]			
Diameter [in]	Used [MHz]	Used	Distance [cm]	No Flaw (ω_1)	Gouge Flaw (ω_2)	Hole Flaw (ω_3)	Average
3/8	0.80, 0.84, 0.89	S0	D_{20}	69.4	57.8	60.5	63.6
3/8	0.80, 0.84, 0.89	A0	D_{20}	76.5	64.6	54.9	70.5
3/8	0.80, 0.84, 0.89	A1	D_{20}	74.2	64.1	54.9	69.2
3/8	0.80, 0.84, 0.89	S0, A0	D_{20}	81.8	66.5	66.0	74.2
3/8	0.80, 0.84, 0.89	S0, A1	D_{20}	79.9	70.3	64.7	75.1
3/8	0.80, 0.84, 0.89	A0, A1	D_{20}	81.1	71.7	60.9	76.4
1/2	0.80, 0.84, 0.89	S0	D_{20}	82.6	33.2	21.3	57.9
1/2	0.80, 0.84, 0.89	A0	D_{20}	65.4	51.0	31.6	58.2
1/2	0.80, 0.84, 0.89	A1	D_{20}	67.8	46.0	30.4	56.9
1/2	0.80, 0.84, 0.89	S0, A0	D_{20}	80.1	44.5	26.9	62.3
1/2	0.80, 0.84, 0.89	S0, A1	D_{20}	79.5	43.4	25.7	61.5
1/2	0.80, 0.84, 0.89	A0, A1	D_{20}	71.0	56.2	30.0	63.6

Table 4.6: The classifier accuracy when the feature selection used only one or two modes at once but all transducer frequencies are shown here. All ray paths were used in this configuration ($D = D_{91}$)

Hole	Frequencies	Modes	Max Path	$A(\omega)$ [%]			
Diameter [in]	Used [MHz]	Used	Distance [cm]	No Flaw (ω_1)	Gouge Flaw (ω_2)	Hole Flaw (ω_3)	Average
3/8	0.80, 0.84, 0.89	S0	D_{91}	92.6	14.3	24.1	43.7
3/8	0.80, 0.84, 0.89	A0	D_{91}	93.1	14.6	25.9	44.5
3/8	0.80, 0.84, 0.89	A1	D_{91}	92.6	15.2	29.7	45.8
3/8	0.80, 0.84, 0.89	S0, A0	D_{91}	93.8	16.1	32.1	47.3
3/8	0.80, 0.84, 0.89	S0, A1	D_{91}	93.2	16.7	31.7	47.2
3/8	0.80, 0.84, 0.89	A0, A1	D_{91}	93.5	16.9	32.7	47.7
1/2	0.80, 0.84, 0.89	S0	D_{91}	67.4	40.8	17.0	41.7
1/2	0.80, 0.84, 0.89	A0	D_{91}	59.6	44.8	20.9	41.8
1/2	0.80, 0.84, 0.89	A1	D_{91}	57.8	48.8	19.9	42.2
1/2	0.80, 0.84, 0.89	S0, A0	D_{91}	64.1	46.0	23.5	44.6
1/2	0.80, 0.84, 0.89	S0, A1	D_{91}	64.6	48.5	19.6	44.2
1/2	0.80, 0.84, 0.89	A0, A1	D_{91}	59.4	53.0	22.5	44.9

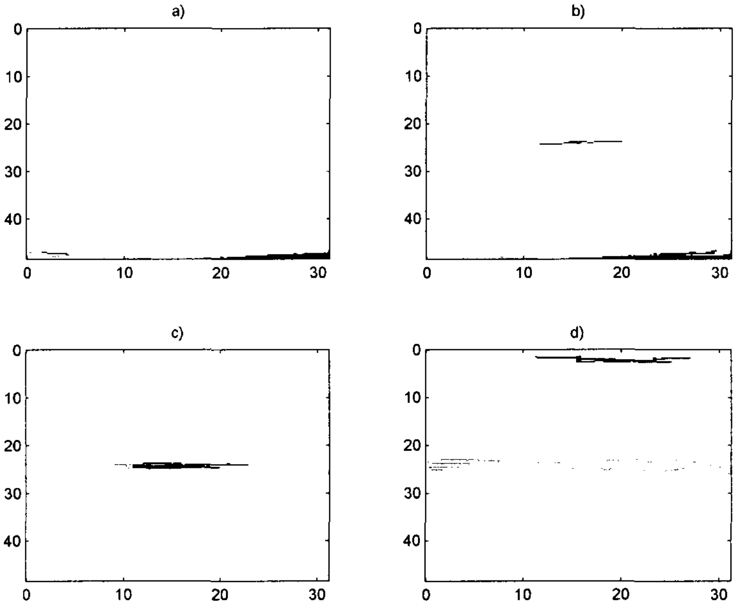


Figure 4.15: The qualitative evaluation of different flaw images produced by the flaw detector routine for the a)-b) gouge and c)-d) hole flaws. False positives are shown in subplots b) and d), while subplots a) and c) show true positives.

Table 4.7: The predicted flaw types using the pipe flaw detection routine are shown here. The columns show the predicted ray path types whose intersection forms the figures as shown in Fig. 4.14 and 4.15. The rows show the different variables used to configure the classifier. All of the classifiers shown here used $D = D_{10}$, $h_I = 1$, $h_a = 10$.

Classifier Configuration			Flaw Type Tested	
Hole Diameter [cm]	Frequencies Used [MHz]	Modes Used	ω_2	ω_3
3/8	0.8	S0, A0, A1	ω_2	ω_3
3/8	0.84	S0, A0, A1	ω_2	ω_3
3/8	0.89	S0, A0, A1	ω_2	ω_3
3/8	0.8, 0.84, 0.89	S0	ω_2	ω_3
3/8	0.8, 0.84, 0.89	A0	ω_2	ω_3
3/8	0.8, 0.84, 0.89	A1	ω_2	ω_3
3/8	0.8, 0.84, 0.89	S0, A0	ω_2	ω_3
3/8	0.8, 0.84, 0.89	S0, A1	ω_2	ω_3
3/8	0.8, 0.84, 0.89	A0, A1	ω_2	ω_3
1/2	0.8	S0, A0, A1	ω_2	ω_1
1/2	0.84	S0, A0, A1	ω_1	ω_3
1/2	0.89	S0, A0, A1	ω_2, ω_3	ω_3
1/2	0.8, 0.84, 0.89	S0	ω_2	ω_1
1/2	0.8, 0.84, 0.89	A0	ω_2, ω_3	ω_3
1/2	0.8, 0.84, 0.89	A1	ω_2	ω_3
1/2	0.8, 0.84, 0.89	S0, A0	ω_2	ω_1
1/2	0.8, 0.84, 0.89	S0, A1	ω_2	ω_1
1/2	0.8, 0.84, 0.89	A0, A1	ω_2, ω_3	ω_3

Table 4.8: Similarly to Table 4.4, the qualitative performance of the pipe flaw detector routine is shown here using $D = D_{20}$, $h_I = 2$, $h_a = 5$.

Classifier Configuration			Flaw Type Tested	
Hole	Frequencies	Modes		
Diameter [cm]	Used [MHz]	Used	ω_2	ω_3
3/8	0.8	S0, A0, A1	ω_2	ω_3
3/8	0.84	S0, A0, A1	ω_2	ω_3
3/8	0.89	S0, A0, A1	ω_2	ω_3
3/8	0.8, 0.84, 0.89	S0	ω_2	ω_3
3/8	0.8, 0.84, 0.89	A0	ω_2	ω_3
3/8	0.8, 0.84, 0.89	A1	ω_2	ω_3
3/8	0.8, 0.84, 0.89	S0, A0	ω_2	ω_3
3/8	0.8, 0.84, 0.89	S0, A1	ω_2	ω_3
3/8	0.8, 0.84, 0.89	A0, A1	ω_2	ω_3
1/2	0.8	S0, A0, A1	ω_2	ω_3
1/2	0.84	S0, A0, A1	ω_2	ω_1
1/2	0.89	S0, A0, A1	ω_2, ω_3	ω_1
1/2	0.8, 0.84, 0.89	S0	ω_2	ω_1
1/2	0.8, 0.84, 0.89	A0	ω_2, ω_3	ω_3
1/2	0.8, 0.84, 0.89	A1	ω_2	ω_1
1/2	0.8, 0.84, 0.89	S0, A0	ω_2	ω_3
1/2	0.8, 0.84, 0.89	S0, A1	ω_2	ω_1
1/2	0.8, 0.84, 0.89	A0, A1	ω_2	ω_1

Table 4.9: Similarly to Tables 4.4 and 4.6, the qualitative performance of the pipe flaw detector routine is shown here using $D = D_{91}$ (so all ray paths were used), $h_I = 3$, $h_a = 20$.

Classifier Configuration			Flaw Type Tested	
Hole Diameter [cm]	Frequencies Used [MHz]	Modes Used	ω_2	ω_3
3/8	0.8	S0, A0, A1	ω_2	ω_3
3/8	0.84	S0, A0, A1	ω_2, ω_3	ω_3
3/8	0.89	S0, A0, A1	ω_2, ω_3	ω_3
3/8	0.8, 0.84, 0.89	S0	ω_2, ω_3	ω_3
3/8	0.8, 0.84, 0.89	A0	ω_2, ω_3	ω_3
3/8	0.8, 0.84, 0.89	A1	ω_2, ω_3	ω_3
3/8	0.8, 0.84, 0.89	S0, A0	ω_2, ω_3	ω_3
3/8	0.8, 0.84, 0.89	S0, A1	ω_2, ω_3	ω_3
3/8	0.8, 0.84, 0.89	A0, A1	ω_2, ω_3	ω_3
1/2	0.8	S0, A0, A1	ω_2	ω_2, ω_3
1/2	0.84	S0, A0, A1	ω_2, ω_3	ω_1
1/2	0.89	S0, A0, A1	ω_2, ω_3	ω_2, ω_3
1/2	0.8, 0.84, 0.89	S0	ω_2, ω_3	ω_2, ω_3
1/2	0.8, 0.84, 0.89	A0	ω_2, ω_3	ω_2, ω_3
1/2	0.8, 0.84, 0.89	A1	ω_2, ω_3	ω_2, ω_3
1/2	0.8, 0.84, 0.89	S0, A0	ω_2, ω_3	ω_2, ω_3
1/2	0.8, 0.84, 0.89	S0, A1	ω_2, ω_3	ω_2, ω_3
1/2	0.8, 0.84, 0.89	A0, A1	ω_2, ω_3	ω_2, ω_3

classification, and likewise Tables 4.8 and 4.9 use a maximum ray path distance of D_{20} and D_{91} respectively. The latter case considers all possible ray path lengths. The results are grouped in this way because the threshold values of h_I and h_a had to be adjusted for each value of D. Obviously, the smaller the value of D, the fewer number of ray paths included in the classification and the smaller the intersection between the predicted ray paths. The last two columns show the pipe flaw detector routine applied to ray paths with a predicted class of either gouge flaw (ω_2) or hole flaw (ω_3) respectively. The type of flaw predicted by the detection routine is displayed under those columns, including the possibility of no flaws displayed (ω_1) or both the gouge flaw and the indentation flaw displayed (ω_2, ω_3). These judgments are made according to the guidelines for false positives and true positives shown in Fig. 4.15.

These results show that Table 4.7, in which $D = D_{10} = 31.21$ cm is the maximum permitted path length used for classification, has the highest number of true positives and the fewest number of false negatives for all the configurations used. Table 4.9 shows the worst discriminative ability when all the possible path lengths ($D = D_{91}$) were applied. However, only one of these classifier configurations needs to perform well in order to select a classifier to be applied in further applications. It is not necessary (or possible) that all combinations of the classifier variables should perform well. Clearly it is possible to find at least one classifier configuration that accurately discriminates between the types of flaws.

In addition, the results for the second tomographic scan of the pipe, when the hole diameter was increased to 1/2 in, tends to have more false positives and false negatives than the original scan of the pipe at a hole diameter of 3/8 in. As described above, the classification for the 1/2 in hole pipe was performed using the 3/8 in hole pipe as a training set, so the accuracy of the classifier tends to be lower than that of the 3/8 in pipe when mutually exclusive sets of ray paths drawn from the same tomographic scan were used for training and testing. Since the same threshold values h_I, h_a were used for all the classifier variables that used the same maximum distance D, it might be possible to adjust these threshold values in the future to optimize for a training set based on a wider variety of data and a testing set drawn from a new tomographic scan.

Lastly, recall that the distribution of ray paths drawn from the three classes $\omega_1, \omega_2, \omega_3$ was adjusted so that a smaller number of ω_1 cases were randomly selected to be used in classification (Fig. 4.11). We also mentioned that the ω_1 ray paths not chosen for classification would later be used to test the pipe flaw detection algorithm. We did in fact perform this test using the first tomographic scan. The training set used for classification was the same as before but it was tested using the ω_1 ray paths originally excluded from classification in the results above. As expected, some of these ray paths were falsely identified by the classifier as ω_2 or ω_3. The pipe flaw detection algorithm was performed on these predicted classes to see if the intersection of these misclassified ray paths would yield a falsely identified flaw. In fact, for all of the classifier variables described in Section 4.4.3, all concluded with an assignment of ω_1 (no flaw identified).

4.7 Conclusion

In this chapter, we have described an artificial intelligence technique that operates on the ray paths of a Lamb wave tomographic scan to produce an image of predicted flaw types and locations. The results shown were necessarily qualitative evaluations of the predicted flaw image but were successfully able to demonstrate classifiers that were configured to be able to distinguish between gouge flaws, hole flaws, and no hole present in an aluminum pipe. The type of image produced by the flaw detection routine is similar to the result of the Lamb wave tomographic scan itself, which is a 2D color plot of the changes in lamb wave velocity over the surface of the pipe. However, because it utilizes pattern classification and image recognition techniques, the method described here can be more specific, identifying the flaw type, and it may be able to work on smaller flaws. The higher accuracy rate of the classifier on limited ray path distances demonstrate that pattern classification aides the successful detection of flaws in geometries where full scans cannot be completed due to limited access. The method does require building a training data set of different types of flaws, and it can take long computer times to classify and test an entire scan. It may be best used on isolated locations where a flaw is suspected rather than in an exhaustive search over a large area. In order to be most accurate, the training data set should include flaws from different individual tomographic scans but of a structure similar to the intended application of the intelligent flaw detector. The application of pattern classification techniques to Lamb wave tomographic scans may be able to improve the detection capability of structural health monitoring.

Table 4.10: A list of mathematical notation

N	=	number of measurements
M	=	number of features
i, j, k, l	=	indices
$\mathbf{x}$	=	feature vector
$\mathbf{x}_i$	=	feature vector of the ith sample
$x_{i,j}$	=	the jth feature corresponding to the ith measurement
p	=	probability
ω	=	class labels
ω_1	=	no flaw
ω_2	=	gouge flaw
ω_3	=	hole flaw
w_i	=	class label corresponding to feature vector $\mathbf{x}$ ($w_i \in \{\omega_1, \omega_2, \omega_3\}$)
λ	=	predicted classes
c_p, c_g	=	phase and group velocity of lamb wave
c_L, c_T	=	longitudinal and transverse wave speeds (relative to the material)
$S0, S1$	=	first and second symmetric lamb wave modes
$A0, A1$	=	first and second antisymmetric lamb wave modes
t	=	time
f	=	frequency
d	=	thickness of the material being scanned
$f \cdot d$	=	frequency-thickness product
r	=	radius of the pipe
L	=	axial transducer separation
Y	=	circumference of the pipe
i_1, i_2	=	indices of the transducer position $(i_1, i_2) \in [1, 180]$
ϕ_1, ϕ_2	=	angles of the transducer
γ	=	minimum angle between the transducers
s	=	helical path length
f_z	=	axial path length
$\phi(t)$	=	a 1D square-integrable function
a, b	=	DWFP parameters ($a \propto f, b \propto t$)
$\psi_{a,b}$	=	mother wavelet
$C(a, b)$	=	wavelet coefficient
$I(a, b)$	=	binary DWFP fingerprint image
$F_{j,v}$	=	DWFP feature, with $\nu = 1 \ldots, 16$
D	=	maximum path length used in classification $s \leq D$
		D_{10}=31.21 cm, D_{20}=31.53 cm,D_{91}=39.38 cm
$\mathcal{C}$	=	classifier map
$\mathcal{L}$	=	confusion matrix
$A(\omega_k)$	=	the accuracy of the classifier at classifying class ω_k
U	=	image of intersection of ray paths with a particular predicted class label
h_a	=	threshold on the area of an object in U
h_I	=	threshold on the pixel value of an object in U

Bibliography

[1] L. Mallet, B. C. Lee, W. J. Staszewski, and F. Scarpa, "Structural health monitoring using scanning laser vibrometry: II. Lamb waves for damage detection," *Smart Materials and Structures*, vol. 13, no. 2, pp. 261–269, 2004.

[2] C. R. Farrar, D. A. Nix, T. A. Duffey, P. J. Cornwell, and G. C. Pardoen, "Damage identification with linear discriminant operators," in 17^{th} *International Modal Analysis Conference*, vol. 3727, no. 2, 1999, pp. 599–607.

[3] J. P. Lynch, "Linear classification of system poles for structural damage detection using piezoelectric active sensors," in *Proceedings of SPIE*, vol. 5391, no. 1, 2004, pp. 9–20.

[4] K. K. Nair, A. S. Kiremidjian, and K. H. Law, "Time series-based damage detection and localization algorithm with application to the asce benchmark structure," *Journal of Sound and Vibration*, vol. 291, no. 1-2, pp. 349 – 368, 2006.

[5] S. W. Kercel, M. B. Klein, and B. F. Pouet, "Bayesian separation of lamb wave signatures in laser ultrasonics," in *Proceedings of SPIE*, K. L. Priddy, P. E. Keller, and D. B. Fogel, Eds., vol. 4055, no. 1, 2000, pp. 350–361.

[6] S. S. Kessler and P. Agrawal, Metis Design Corporation, Tech. Rep., 2007, presented to the American Society for Composites. [Online]. Available: http://www.metisdesign.com/docs/ppt/ASC07.pdf

[7] A. Mita and A. Fujimoto, "Active detection of loosened bolts using ultrasonic waves and support vector machines," in *Proceedings of the 5th International Workshop on Structural Health Monitoring*, Stanford University, Sept 2005, pp. 1017–1024.

[8] S. Park, C.-B. Yun, Y. Roh, and J.-J. Lee, "Pzt-based active damage detection techniques for steel bridge components," *Smart Materials and Structures*, vol. 15, no. 4, pp. 957–966, 2006.

[9] S. Park, J. Lee, C. Yun, and D. J. Inman, "A built-in active sensing system-based structural health monitoring technique using statistical pattern recognition," *Journal of Mechanical Science and Technology*, vol. 21, pp. 896–902, 2007.

[10] K. Worden, S. G. Pierce, G. Manson, W. R. Philp, W. J. Staszewski, and B. Culshaw, "Detection of defects in composite plates using lamb waves and novelty detection." *International Journal of Systems Science*, vol. 31, no. 11, pp. 1397 – 1409, 2000.

[11] H. Sohn, D. W. Allen, K. Worden, and C. R. Farrar, "Structural damage classification using extreme value statistics," *Journal of Dynamic Systems, Measurement, and Control*, vol. 127, no. 1, pp. 125–132, 2005.

[12] J. Mirapeix, P. García-Allende, A. Cobo, O. Conde, and J. López-Higuera, "Real-time arc-welding defect detection and classification with principal component analysis and artificial neural networks," *NDT&E International*, vol. 40, no. 4, pp. 315 – 323, 2007.

[13] J. E. Lynch, "Ultrasonographic measurement of periodontal attachment levels," Ph.D. dissertation, Department of Applied Science, College of William and Mary, Williamsburg, VA, 2001.

[14] J. C. P. McKeon and M. K. Hinders, "Parallel projection and crosshole lamb wave contact scanning tomography," *The Journal of the Acoustical Society of America*, vol. 106, no. 5, pp. 2568–2577, 1999.

[15] J. Hou, K. R. Leonard, and M. K. Hinders, "Multi-mode lamb wave arrival time extraction of improved tomographic reconstruction," in *Review of Progress in Quantitative Nondestructive Evaluation*, D. O. Thompson and D. E. Chimenti, Eds., vol. 24a, Melville, NY, 2005, pp. 736–743.

[16] K. E. Rudd, K. R. Leonard, J. P. Bingham, and M. K. Hinders, "Simulation of guided waves in complex piping geometries using the elastodynamic finite integration technique," *The Journal of the Acoustical Society of America*, vol. 121, no. 3, pp. 1449–1458, 2007.

[17] J. P. Bingham and M. K. Hinders, "Lamb wave detection of delaminations in large diameter pipe coatings," *Open Acoustics*, vol. *in press*, 2009.

[18] K. R. Leonard and M. K. Hinders, "Guided wave helical ultrasonic tomography of pipes," *Journal of the Acoustical Society of America*, vol. 114, pp. 767–774, 2003.

[19] E. V. Malyarenko and M. K. Hinders, "Ultrasonic lamb wave diffraction tomography," *Ultrasonics*, vol. 39, no. 4, pp. 269 – 281, 2001.

[20] H. Ravanbod, "Application of neuro-fuzzy techniques in oil pipeline ultrasonic nondestructive testing," *NDT&E International*, vol. 38, pp. 643–653, 2005.

[21] X. Zhao, V. K. Varma, G. Mei, B. Ayhan, and C. Kwan, "In-line nondestructive inspection of mechanical dents on pipelines with guided shear horizontal wave electromagnetic acoustic transducers," *Journal of Pressure Vessel Technology*, vol. 127, no. 3, pp. 304–309, 2005.

[22] Y. Sun, P. Bai, H. Sun, and P. Zhou, "Real-time automatic detection of weld defects in steel pipe," *NDT&E International*, vol. 38, pp. 522–528, 2005.

[23] A. D. Pierce and H. Kil, "Elastic wave propagation from point excitations on thin-walled cylindrical shells," *Journal of vibration and acoustics*, vol. 112, pp. 399–406, 1990.

[24] I. A. Viktorov, *Rayleigh and Lamb Waves: Physical Theory and Applications.* New York: Plenum Press, 1967.

[25] J. L. Rose, *Ultrasonic Waves in Solid Media.* New York: Cambridge University Press, 1999.

[26] A. R. Webb, *Statistical Pattern Recognition*, 2^{nd} ed. Hoboken, NJ: Wiley, 2002.

[27] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines.* New York: Cambridge University Press, 2000.

[28] J. Hou and M. K. Hinders, "Dynamic Wavelet Fingerprint identification of ultrasound signals," *Materials Evaluation*, vol. 60, no. 9, pp. 1089–1093, 2002.

[29] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision.* Boston, MA: Addison-Wesley, 1992, vol. I.

[30] B. K. P. Horn, *Robot Vision.* Cambridge, MA: MIT Press, 1986.

Chapter 5

Fuzzy Classification of Roof Fall Predictors in Microseismic Monitoring

Microseismic monitoring involves placing geophones on the rock surfaces of a mine to record seismic activity. Classification of microseismic data can be used to predict seismic events in a mine to mitigate mining hazards, such as roof falls, where properly bolting and bracing the roof is often an insufficient method of preventing weak roofs from destabilizing. In this study, six months of recorded acoustic waveforms from microseismic monitoring in a Pennsylvania limestone mine were analyzed using classification techniques to predict roof falls. Fuzzy classification using features selected for computational ease was applied on the mine data. Both large roof fall events could be predicted using a Roof Fall Index (RFI) metric calculated from the results of the fuzzy classification. RFI was successfully used to resolve the two significant roof fall events and predicted both events by at least 15 hours before visual signs of the roof falls were evident.

5.1 Introduction

Roof falls are a hazard in mining that can lead to injuries, damaged equipment, blocked escape routes, and disrupted resource production in the mine [1]. For example, studies by the Mine Safety and Health Administration (MSHA) show that mines producing only 20% of coal result in 60% of roof falls, mostly located in Illinois and the Appalachian coal basins [2]. These types of mining hazards are of large concern in room-and-pillar mines, in which properly supporting the roof does not always prevent roof falls. Other techniques, such as roof screening, are also insufficient in preventing roof falls. Predicting roof falls can improve mine safety by providing enough forewarning of a roof fall that equipment and personnel can be moved out of the unstable area before the fall occurs.

Supervising and evaluating mine activity often occurs by microseismic monitoring because it can detect ground motion changes due to seismic activity during mining [3]. Seismic activity caused by mining is due to stress field changes in the rock material from mining excavations [4]. Geophones operating in the audible range are installed in the rock face or in boreholes to measure velocity at the rockmass. Microseismic monitoring has the advantage of continuous deflection measurements and a relatively low number of instruments required to cover the mine [5], although an array of geophones must be installed throughout the mine in order to precisely locate the source of seismic activity. The geophones passively monitor the mine and can record events ranging from creaking due to tensile strain to normal mine activities such as blasts and scaling.

The approach of much of the geophysical community toward microseismic monitoring requires measuring the source location and various parameters. The value of those source parameters then indicates the severity of the seismic activity in the same way that the Richter scale indicates the severity of an earthquake. Seismic source parameters, especially stress drop, are often used to measure seismicity and have been shown to be useful in correlating rock failure with microseismic events [6]. Seismic source parameters include scalar seismic moment (M_0), stress drop ($\Delta\sigma$), and apparent stress (σ_a). They can be written [4]:

$$M_0 = \mu_s \vec{u} A \tag{5.1}$$

$$\Delta\sigma = \frac{7}{16}\frac{M_0}{r_0^3} \tag{5.2}$$

$$\sigma_a = \eta\vec{\sigma} \tag{5.3}$$

where the other parameters include area A, average displacement $\vec{u}$, shear modulus μ_s, seismic efficiency η, and radius r_0 (usually $r_0 = 2.34\beta_0/2\pi f_s$). Other seismic parameters calculated from the spectra are described in Section 5.3.2, including corner frequency (f_c) and low frequency spectral level (Ω_0), which can be used to determine M_0 directly, as well as the seismic energy.

The type of rock failure from the mining-induced stresses can vary. Some types are sudden, such as rock bursts and gas outbursts. Roof failures, on the other hand, are gradual and are largely caused by tensile failure. Roof falls occur in mines when horizontal stress causes roof strata to buckle, rupture, and fall [7]. Merwe *et al.* [8] examined the causes of roof falls in South African collieries and found that 64% of roof falls were caused by wide bolt spacing and poorly-supported joints. Sometimes roof falls can be predicted by visual inspection. Warning signs of roof falls can include sounds like rock popping and creaking and visual identifiers such as bolt heads popping off rock bolts, small pieces of rock falling from the roof, and sagging of the roof because the roof bolts have been sheared or broken [9]. However, automatic methods of predicting roof falls could improve mine safety in a way that eliminates observer hazards.

There are some studies of microseismic monitoring to predict roof falls in the literature, many involving the Springfield Pike mine data that is examined in this chapter. In Iannacchione *et al.*'s 2000 study of the Springfield Pike mine, they noted that the background rate of seismic activity at $0.14 - 0.35$ events/hour increased significanty to $0.5 - 12.3$ events/hour during periods of elevated microseismic activity, showing that monitoring seismic activity over time can predict roof falls [10]. In 2005, Iannacchione *et al.* compared seismic activity as measured by geophones with roof-to-floor convergence measurement devices, in which a voltage change indicates that the roof and floor have moved with respect to each other [11]. In some cases, they found that the microseismic activity preceded the roof-to-floor convergence and in general verified a connection between roof movement and microseismic activity in the Springfield Pike mine. That same year, Iannacchione *et al.* used source parameters such as seismic moment to examine rock drop tests and microseismic data from the Springfield Pike mine during 2002-2003 [12]. On a log-based scale, they showed an ability to distinguish between rock fractures, roof falls, drop tests, and blasts when apparent stress was plotted versus seismic moment. In order to get more data, Iannacchione's group examined data from the Moonee Colliery Mine, an Australian longwall coal mine, from the years 1998-2002 in order to predict roof falls [5]. After an initial monitoring period, they developed a roof fall alarm based on frequency of seismic activity, magnitude, and trend, which was successful in 73% of the roof falls. Of the false alarms, the authors suggest that cessation of mining due to the alarm might have halted the failure process. Most recently, Ge *et al.* [13] examined 2 years of data from the Springfield Pike mine in order to develop an accurate location algorithm. Their method, however, is time consuming and requires interactive user input.

Many of the methods listed above rely upon accurate measurements of seismic source parameters in conjunction with monitoring seismic event activity over time to predict roof falls. It seems logical to suggest circumventing the difficulty of obtaining accurate measurements of source parameters by implementing pattern recognition methods and artificial intelligence techniques to identify hazardous seismic events. Pattern classification techniques have previously been applied to seismic records [14]. A number of the earlier works focused on texture analysis methods in order to, for example, detect mineral rich areas [15, 16] or to detect bright spots in seismograms [17, 18]. Some researchers used pattern recognition to distinguish earthquakes from underground explosions [19, 20] or to measure deformation [21]. Most research has focused on pattern recognition to improve source parameter detection, for example, applying pattern recognition to laboratory acoustic emission records in order to distinguish stress levels of rock plates under loading [22] or using pattern recognition techniques to improve hypocenter location calculations for earthquakes [23]. Detecting optimal feature sets for classification has also been performed using synthetic training data [24, 25]. However, such applications were not intended for prediction and not necessarily applicable to real-time analysis.

Instead of attempting to accurately measure source parameters, we focus on fuzzy classification techniques to cluster the recorded seismic data into groups that indicate the severity of seismic activity with the aim of predicting roof falls in real time.

5.2 Data Collection

The microseismic data considered here was collected by the National Institute for Occupational Safety and Hazards (NIOSH) at the Springfield Pike limestone mine in

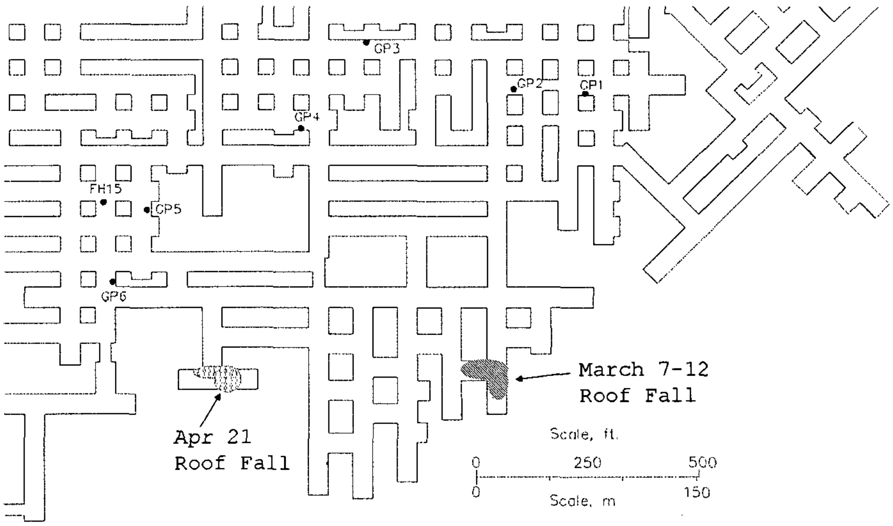


Figure 5.1: The location of the geophones (GP1-GP6, FH15) near the event are shown on the mine map along with the roof fall locations. This use of this figure is with permission from NIOSH.

Pennsylvania. A map of the mine is shown in Figure 5.1. This mine accesses the Loyalhanna Limestone structure, which is prone to roof falls that have led to mine worker injuries in the past [26]. The mine is of the room-and-pillar type, in which pockets of limestone are extracted while pillars of rock are left standing for support. The study lasted from November 2001-April 2002 with 15 4.5 Hz uniaxial geophones installed in the mine throughout the study. During the period of microseismic monitoring, 13,228 different microseismic events were triggered, but only two significant roof falls occurred. The events were triggered for recording whenever a particular threshold of velocity amplitude was reached for several geophones at once. In the case of the two recorded roof fall events, the first recognition of roof failure occurred at 3am on March 7 and 8am on April 21, respectively [7]. Figure 5.2 shows the number of events per day recorded in the Springfield Pike mine. The roof fall dates are indicated by dotted lines. The event density varies widely over the period of study and is not sufficient to predict roof falls, since large number of events occur at times when roof falls were not recorded to have occurred, while the number of events leading up to the April 21 roof fall is small.

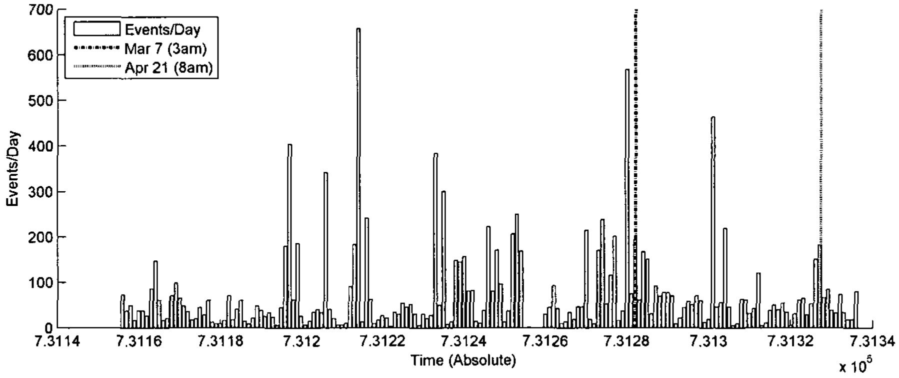


Figure 5.2: The number of events per day is plotted along with the dates roof falls occurred in the Springfield Pike mine during the period under study. The event density is not sufficient to predict roof falls in this case.

Figure 5.3 compares four different sample waveforms. The first plot, labeled as a fracture event, shows the kind of waveforms that indicate shear stresses leading to roof falls. The second plot shows no visible event, whether seismic or noise. The third plot demonstrates the geophone response to electrical noise perhaps caused by a lightening storm, and the last waveform shows a recorded blast event in the mine. The event shown in the first plot demonstrates the kind of waveforms that the fuzzy classification algorithm needs to identify and separate from the other three sample waveforms, which are still triggered by the system as significant but do not reflect seismic activity. Previous studies have shown that rock fracture events produce a signal that is sharply defined, with large amplitude and short duration [12, 27], such as the fracture event waveform shown in Fig. 5.3.

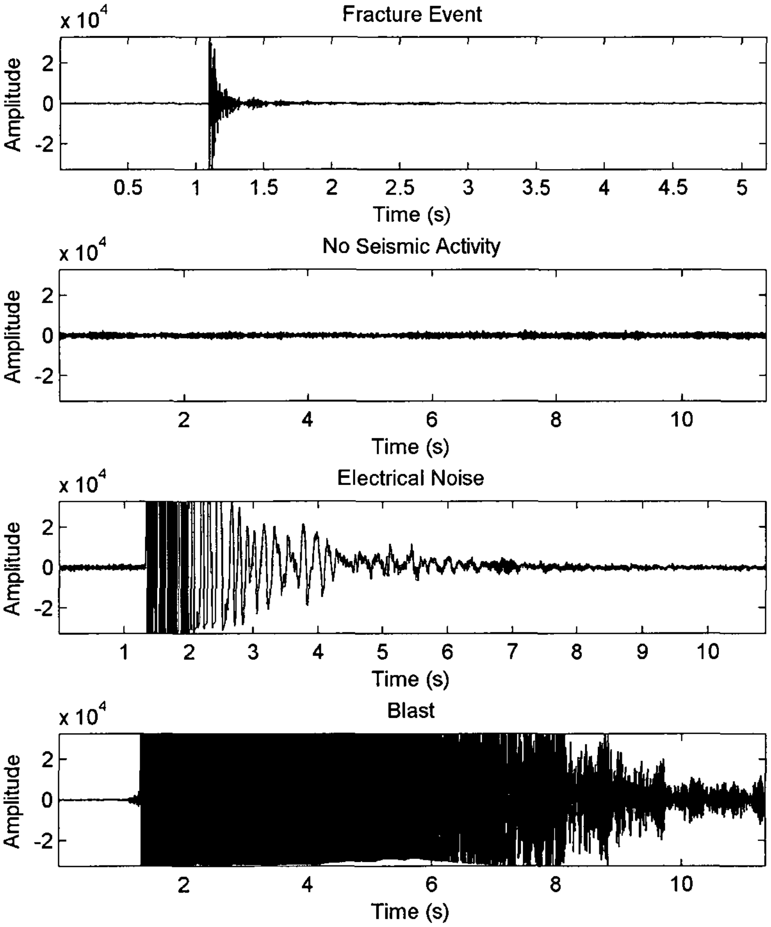


Figure 5.3: Four different waveforms have been selected from the recorded seismic data. The amplitude of the events is normalized to each individual event, so a direct comparison of the amplitude of waveforms recorded from different events cannot be performed.

5.3 Feature Selection

Pattern classification proceeds by finding a set of p features from each of the $n = 13,228$ observations yielding a classification matrix of size $n \times p$. We will note the jth feature of the ith recorded event as w_j^i, where $j = 1, \dots, p$ and $i = 1, \dots, n$. In the microseismic data set described above, there is also the additional dimension of 15 geophones. However, the fuzzy classification approach described here will classify each of the $n \times p$ data sets for each of the 15 geophones separately. Whenever several geophones simultaneously record a significant event, the roof fall event can be roughly located in the mine by the region covered by those geophones.

The goal was to select features for classification that are associated with rock fracture events, such as shown in Fig. 5.3. There were two main techniques used for feature selection. The first technique uses the Dynamic Wavelet Fingerprint (DWFP), and the second measures spectral parameters.

5.3.1 DWFP

To extract features from the microseismic waveform, the DWFP technique (Fig. 5.4) applies a wavelet transform on the original time domain waveform, which results in "loop" features that resemble fingerprints. It has previously shown promise for a variety of applications.

The first step of the DWFP (Fig. 5.4a-b) involves applying a wavelet transform on each of the microseismic waveforms. The continuous wavelet transform can be written

$$C(a,b) = \int_{-\infty}^{+\infty} f(t)\overline{\psi_{a,b}(t)}dt \tag{5.4}$$

Here, $f(t)$ represents a square-integrable 1D function, and $\psi(t)$ represents the mother wavelet. The mother wavelet is scaled in frequency (f) and transformed in time (t)

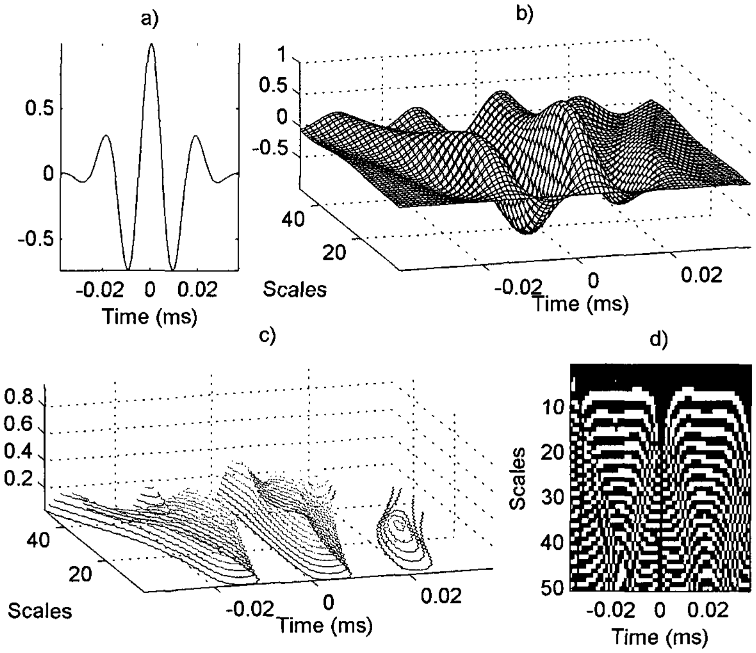


Figure 5.4: The DWFP technique [28] begins with a) the ultrasonic signal, where it geneates b) wavelet coefficients indexed by time and scale, where scale is related to frequency. Then c) the coefficients are sliced and projected onto the time-scale plane (d). The final binary image is used to select features for the pattern classification algorithm.

using $a, b \in \mathbb{R}$, respectively, in order to form the $\psi_{a,b}(t)$ in Equation 5.4. The wavelet transform on a single waveform (Fig. 5.4a) results in wavelet coefficients (Fig. 5.4b). Then, a slicing algorithm is applied to create an image analogous to the gradient of the wavelet coefficients in the time-scale plane, resulting in a binary image, $I(a, b)$:

$$f(t) \overset{DWFP(\psi_{a,b})}{\longrightarrow} I(a, b) \tag{5.5}$$

In this application, the Haar wavelet was selected because of its simplicity. Figure 5.5 shows the wavelet transforms of the waveforms in Figure 5.3. Note that only a small time window is displayed here for clarity. The fingerprint shape is difficult to see here because no filtering has been performed, and small fluctuations in the waveforms have a large effect on the time-scale plane. Despite this clutter, the wavelet fingerprints appear drastically different for the four different types of waveforms. In particular, the fracture event, which is characterized by high amplitude and short duration, has isolated but highly complex fingerprints. Meanwhile, the waveforms that do not display any seismicity tend to have a constant string of nearly identical waveforms over the whole window.

The next step involves extracting a single feature from the image $I(a, b)$. Because of the high amplitude, short duration nature of the fracture events, the resulting fingerprints are isolated and complex. Therefore, to measure the complexity of the fingerprints, the algorithm counts the number of ridges in the fingerprint. For each value of b, it finds the number of times $I(a, b)$ has changed between 1 and 0, resulting in an array of ridge-counts $R(b)$, a vector similar in size to the original waveform $f(t)$ (Figs 2.18 and 2.19). The final feature, w_1, finds the number of times b such that

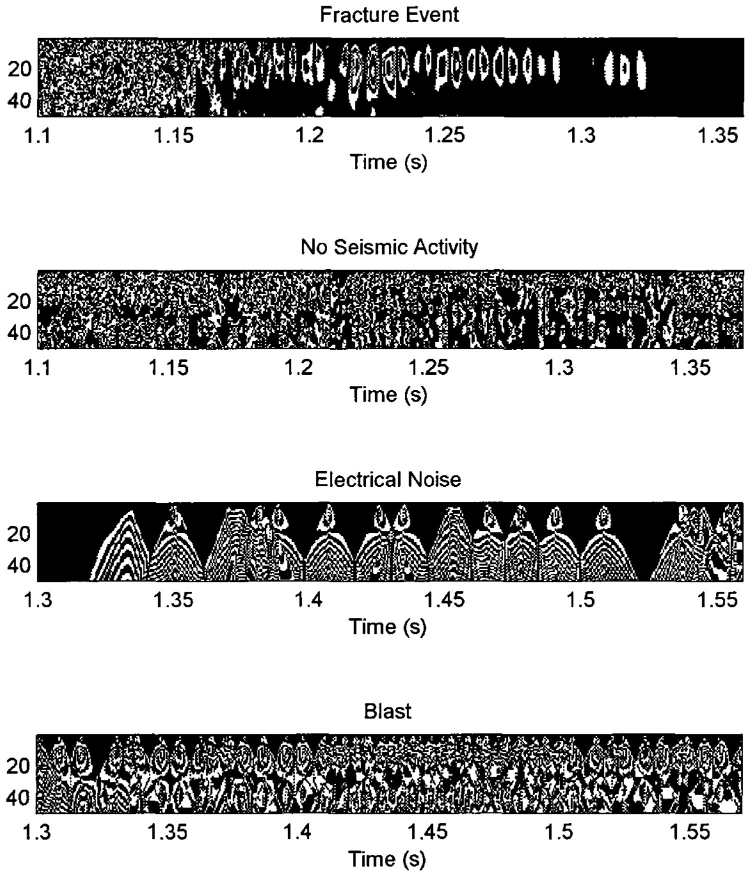


Figure 5.5: The results of applying the DWFP on the four waveforms from Fig. 5.3 are displayed.

$R(b)$ is greater than some threshold:

$$w_1 = \#b \mid \{R(b) > 0.3 \cdot \max R\} \tag{5.6}$$

Figure 5.6 shows the resulting ridge-count measurements $R(b)$ and the resulting calculated feature measure w_1, which is the number of ridges greater than 30% of the maximum value. The fracture event has the lowest value of w_1, as desired.

5.3.2 Spectral Parameters

Previous work has established that seismic data can be modeled by two parameters: low level frequency response (Ω_0), and corner frequency (f_c) [29, 30]. That is, the displacement spectral amplitude $|D(\omega)|$ can be modeled by

$$|D(\omega)| = \frac{\Omega_0}{(1 + \frac{\omega}{f_c}^{\gamma m})^{1/\gamma}} \tag{5.7}$$

where ω is frequency, m is the falloff rate of the log-log plot, and γ is a constant (usually $m = 2$, $\gamma = 1, 2$). In this form, the displacement and velocity spectral amplitudes should appear as in Figure 5.7 on a log scale, with the sharpness of the corners determined by γ and slopes determined by m [31]. So in general, below the corner frequency, the displacement spectral amplitude is a constant, and above the corner frequency, the displacement spectral amplitude is a line with negative slope. Similarly, the velocity displacement spectrum is a line with negative slope below the corner frequency and a line with positive slope above the corner frequency.

It is possible to calculate Ω_0 and f_c from these spectral images. Another way, however, is to calculate the source parameters from spectral integrals [32]. Given the velocity power spectrum $V^2(f)$ and the displacement power spectrum $D^2(f) =$

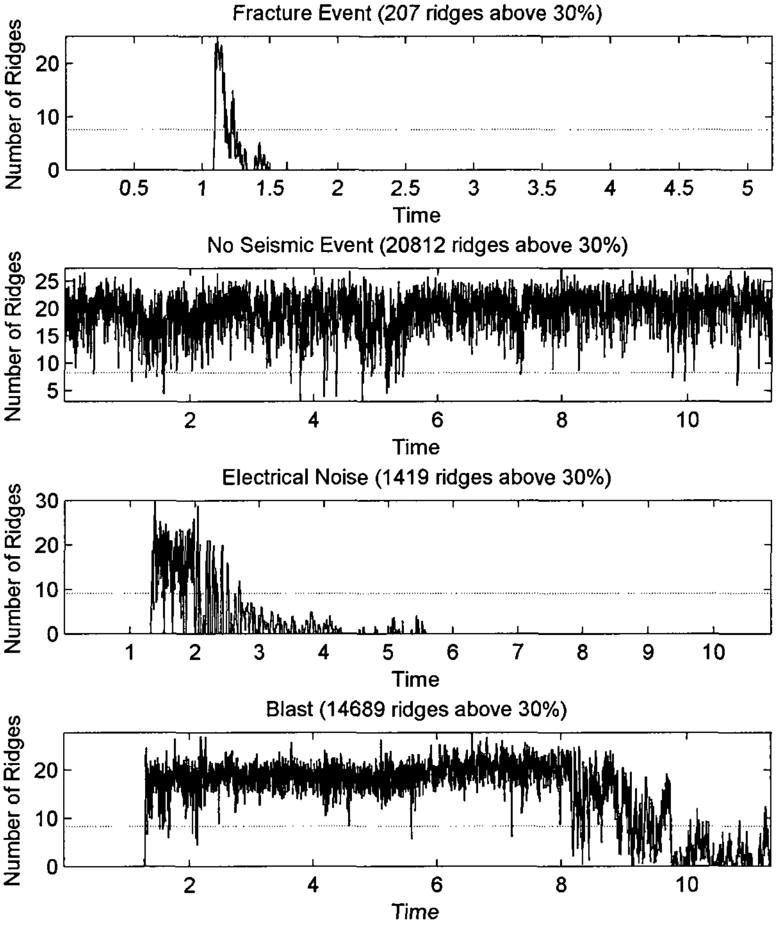


Figure 5.6: The number of ridges in Fig. 5.5 have been counted here, with the 30% threshold marked as a dotted line.

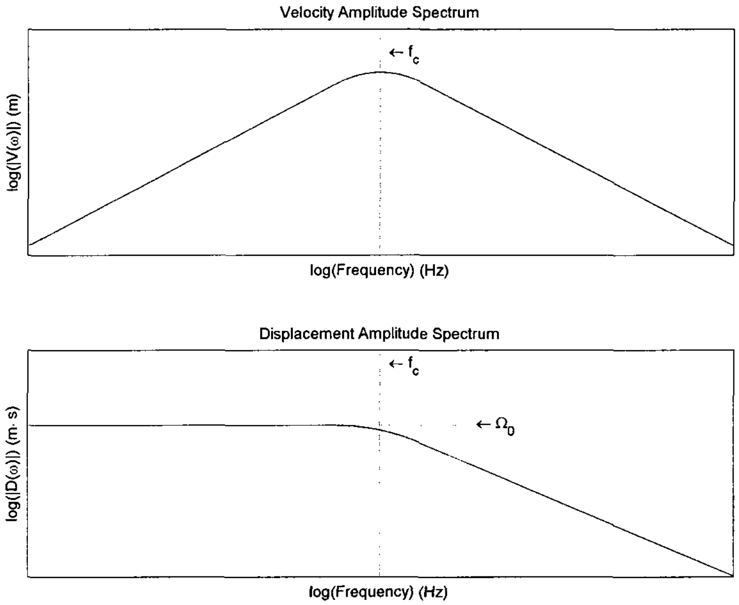


Figure 5.7: A cartoon of the predicted velocity and displacement spectral amplitude shapes as predicted by Brune. The figure is adapted from [31].

$V^2(f)/(2\pi f)^2$ can be used to calculate the following integrals:

$$S_{D2} = 2\int_0^\infty D^2(f)\mathrm{d}f \tag{5.8}$$

$$S_{V2} = 2\int_0^\infty V^2(f)\mathrm{d}f \tag{5.9}$$

Then, assuming a Brune spectral shape (Eq. 5.7) with parameters $m = 2$, $\gamma = 1$ gives values of corner frequency and low-frequency spectral level:

$$f_c = \frac{1}{2\pi}\sqrt{\frac{S_{V2}}{S_{D2}}} \tag{5.10}$$

$$\Omega_0 = \sqrt{4S_{D2}^{3/2}S_{V2}^{-1/2}} \tag{5.11}$$

However, the source parameters of seismic waveforms cannot usually be estimated by merely plotting the spectral shapes (Fig. 5.7) or by applying Eqs. (5.10)-(5.11). Certain corrections need to be made first. Corrections for geometrical spreading usually involves multiplying by the square of the hypocentral distance [32], which is the distance from the source of the seismic activity to the receiver. This can be calculated in a number of ways, including seismic ray tracing methods and matrix inversion methods [4]. All methods require knowing the precise arrival of the longitudinal wave (p-wave), but calculating the p-wave arrival is also iterative. One method, such as short-term over long-term averaging, is commonly employed. All methods generally attempt to locate the leading edge of the highest-amplitude event in the signal. That event is assumed to be the p-wave.

However, in order for hypocentral distance calculations to be accurate on the order required for source parameter calculations, the p-wave and s-wave arrival times should also be calculated accurately. However, roof fall events tend to involve complex, emergent waveforms caused by numerous rocks hitting the ground, possibly obscuring p- and s-wave arrivals [12]. Since the p- and s-wave arrival times were difficult to detect, then the hypocentral distance calculations were inaccurate, making corrections for geometrical spreading and instrument response incorrect. Additionally, the data does not fit the Brune spectral shape well [33].

In the end, the spectral source parameters f_c and Ω_0 were calculated without the corrections for geometrical spreading and instrument response that would be required in order to assure an accurate numerical value. Besides the difficulty of resolving s-wave arrivals from the microseismic data, other reasons include the fact that the computations involved are time-consuming, unlikely to be implemented in real-time in a mine, and often require user input for tasks such as selecting p-wave arrivals and estimating an expected location for the seismic event. See, for example, Ge *et al.*'s paper on creating a source location algorithm for the Springfield Pike mine data in order to illustrate the complexity and difficulty of this task [13]. In any case, since our desired application is for pattern classification, the exact numerical value of the source parameters may not be required. As will be shown, uncorrected waveforms provided measurements with enough sensitivity to be successful for fuzzy classification techniques.

Both the spectral method (Fig. 5.7) and the integral method (Eqns. 5.10 and 5.11) were used to approximate f_c and Ω_0. In both cases, the only corrections to the raw data included removing any recorded information under the 4.5Hz frequency of the transducer and subtracting the mean of the waveform to eliminate the DC offset.

5.3.3 Summary of Extracted Features

Both the DWFP features and the source parameter features were used for pattern classification. In total, 8 features were selected. They include the following, in order, where μ represents the arithmetic mean and σ represents the standard deviation:

1. The number of peaks above 30% in the DWFP ridge-counts (Eqn. 5.6).

2. The maximum value of the waveform.

3. The corner frequency as measured by the frequency at which the maximum velocity spectral amplitude occurs (as in Fig. 5.7).

4. The low frequency spectral limit measured from the displacement spectral amplitude. If f_c is the corner frequency from item 3 above, then the low frequency spectral limit is measured by

$$\Omega_0 = \mu\big(D(f)\big) + 2\sigma\big(D(f)\big) \tag{5.12}$$

for $f < f_c$, where $\mu(X)$ and $\sigma(X)$ are respectively the mean and standard deviation functions of the distribution X.

5. The average dispersion of the velocity spectral amplitude, measured. If $\mathrm{d}V$ represents the difference between adjacent values of the velocity spectral amplitude V, then the dispersion is
$$\Delta = \mu(|\mathrm{d}V|). \tag{5.13}$$

6. The variance in the velocity spectral amplitude.

7. The value of f_c calculated using Andrews' integral method (Eqn. 5.10).

8. Similarly, the value of Ω_0 calculated using integrals (Eqn. 5.11).

5.4 Supervised Pattern Recognition

The result of the feature extraction process above provides 8 features for the 13,228 waveforms captured over the period of the study for each of 15 geophones. Because the correspondence between the events recorded by the geophone and the physical interpretation of mine activity is uncertain, supervised pattern recognition is not immediately possible. However, the shape of fracture events is known (Fig. 5.3a) [12, 27]. We can use the known shape of fracture events to label some portion of the data and use that to train a classifier on which the rest of the data set is tested.

The training region selected occurred from March 5-7, 2002, during which 529 events occurred. All 529 events from all 15 geophones were viewed manually and 81 fracture events were selected. Figure 5.8 shows some of these manually-selected waveforms and the time at which they were recorded. Note the similarity of shape between these events. An event was selected as a precursor if even only one of its geophones recorded such a fracture event.

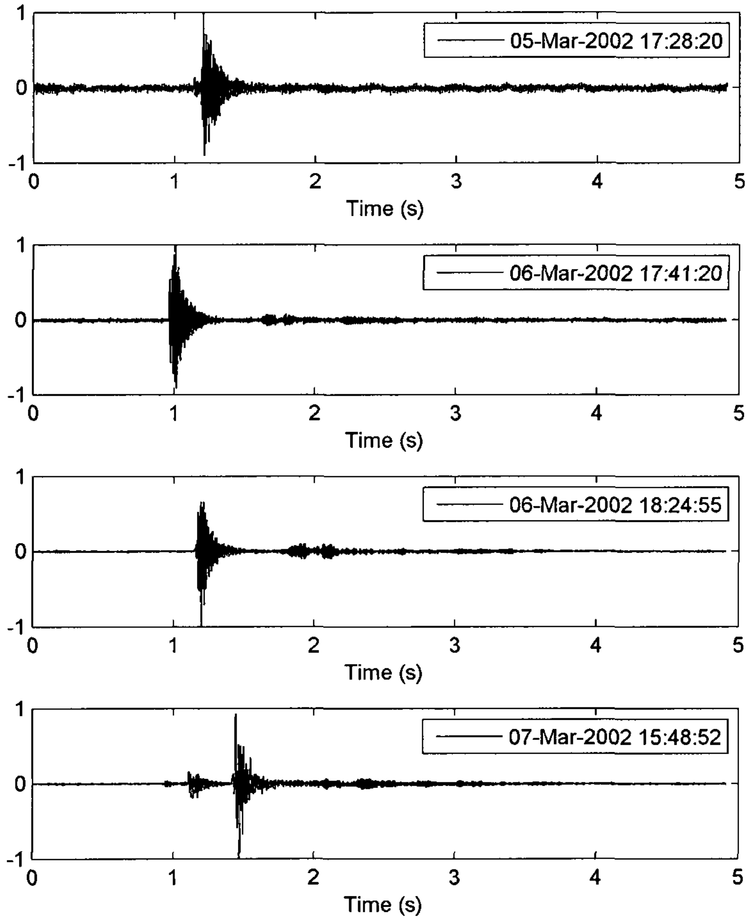


Figure 5.8: Several manually-selected precursor events to the March 7, 2002 roof fall are compared here to show similarity of shape. These are only 4 of the 81 manually-selected precursor events recorded during March 5-7.

Two features were used for classification - the number of DWFP ridges above 30% of the maximum (w_1), and the maximum value of the waveform (w_2). The training data set manually selected from the Mar 5-7 events was then used to classify the remaining seismic data for precursors. Each other event not used for training was classified separately against this training set so that one predicted label was associated with each event. The classification was performed for features gathered from all geophones' recorded events and the number of fracture events per day (e_+) and non-fracture events per day (e_-) were calculated. These were normalized by their totals, so that

$$E_+ = \frac{e_+}{|e_+|}$$

and similarly for the normalized non-fracture events per day, E_-. The relative difference between the two, $E_+ - E_-$ is shown in Figures 5.9-5.11 during the time period January 1, 2002 - April 30, 2002 with the two roof fall dates indicated. Figure 5.9 shows $E_+ - E_-$ for geophones 1 through 5, while Figs. 5.10 and 5.11 and shows the results for geophones 6 through 10 and 11 through 15 respectively.

In order to predict roof falls using these results, we are looking for more fracture-identified events in a single day than non-fracture events. Therefore the difference $E_+ - E_-$ should be positive whenever there are a significant number of fracture events occurring in one day. Figure 5.9 shows that the March roof fall had significant fracture events during March 8-10 that show up in geophones 1 and 2. Figure 5.11 shows the only other significant activity on March 5 in geophone 15 for that roof fall event. Similarly, the April roof fall shows up strongly in several geophones, including April 20-22 in geophones 4 and 5 (Fig. 5.9), geophones 6 and 7 (Fig. 5.10), and geophones 13-15 (Fig. 5.11). As the mine map shows (Fig. 5.1), these were mostly the geophones actually close to the event. However, other events show up on this type of analysis that were not positively identified as roof falls in the mine. Significant activity occurs on April 6 in geophones 1-3 and 9, on January 30 in geophones 9-12, and on February 27 in geophones 8-11, among other smaller events. Table 5.1 summarizes the results, indicating the geophones and consecutive dates in which the events per day difference $E_+ - E_- > 0.015$, with the correct identifications highlighted. If instead the threshold for $E_+ - E_-$ is raised, the number of true positives decreases. In addition, such a method has little predictive capability, since the events per day measure does not give enough warning time to take action in the mine, in addition to providing false positives relative to the roof fall events that were visually recorded in this study.

Table 5.1: The geophones and dates registering $E_+ - E_- > 0.015$ with the events corresponding to the actual roof falls are highlighted. Some false positives occur with this method.

Geophones	Begin Date	End Date
6	1/17/2002	1/17/2002
5	1/19/2002	1/19/2002
10, 11, 12	1/30/2002	1/30/2002
8, 9, 10, 11	2/27/2002	2/27/2002
15	3/5/2002	3/5/2002
1, 14	3/7/2002	3/7/2002
1, 2, 4, 13, 14	3/9/2002	3/10/2002
9, 12	3/26/2002	3/26/2002
7, 9, 12	3/29/2002	3/29/2002
1	4/6/2002	4/6/2002
4, 5, 6, 7, 13, 14, 15	4/20/2002	4/21/2002
7	4/23/2002	4/23/2002
7	4/30/2002	4/30/2002

5.5 Clustering

In general, clustering is an unsupervised learning method, which means that the given data set does not have any known label values. The microseismic data from the Springfield Pike mine is similarly unlabeled, since it cannot be determined exactly

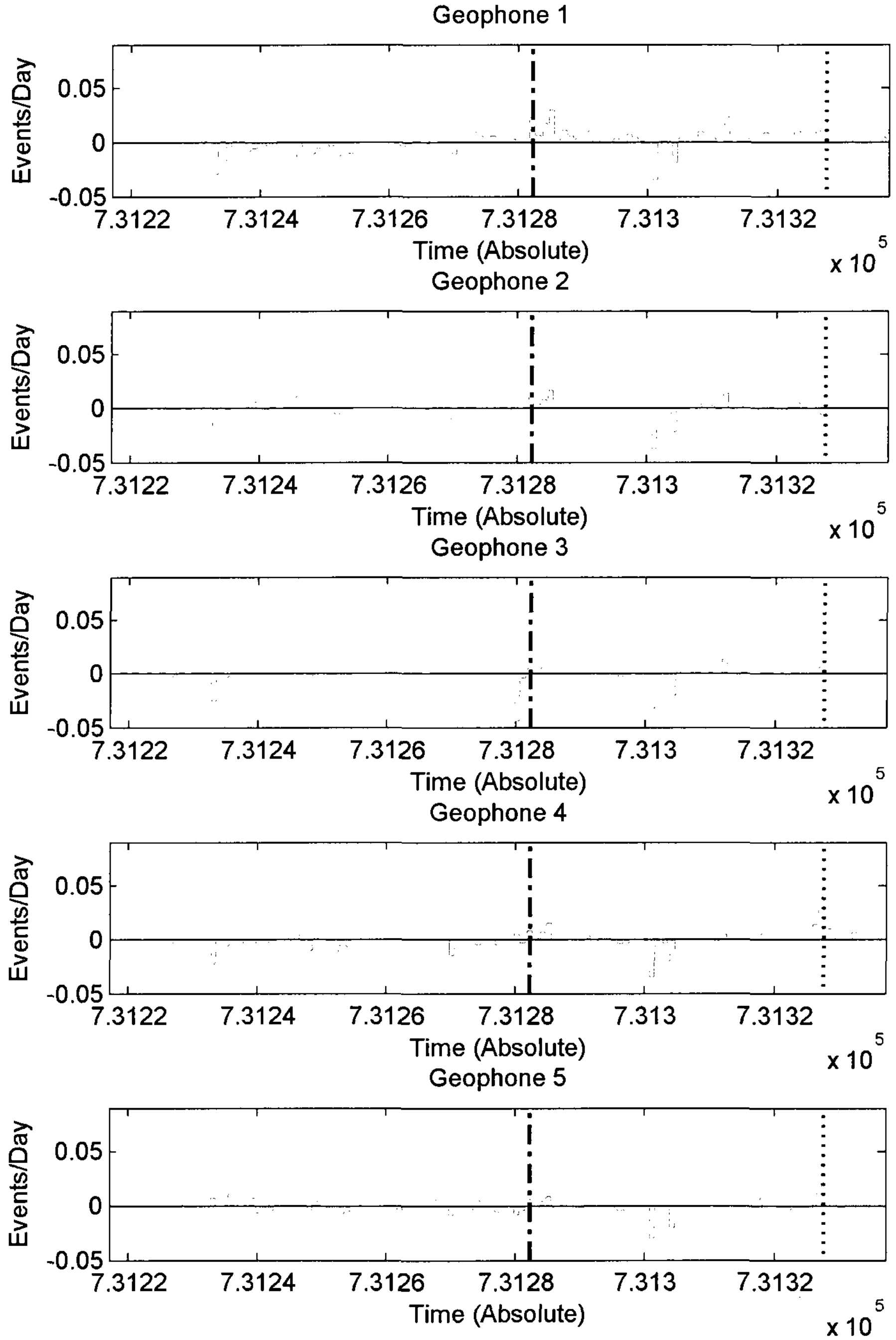


Figure 5.9: For the time range of January 1, 2002 - April 30, 2002, the quantity $E_+ - E_-$ is plotted to show the relative difference between the classified fracture events per day from the non-fracture events per day. The two roof fall dates are indicated by dotted lines. Each plot shows $E_+ - E_-$ for geophones 1 through 5.

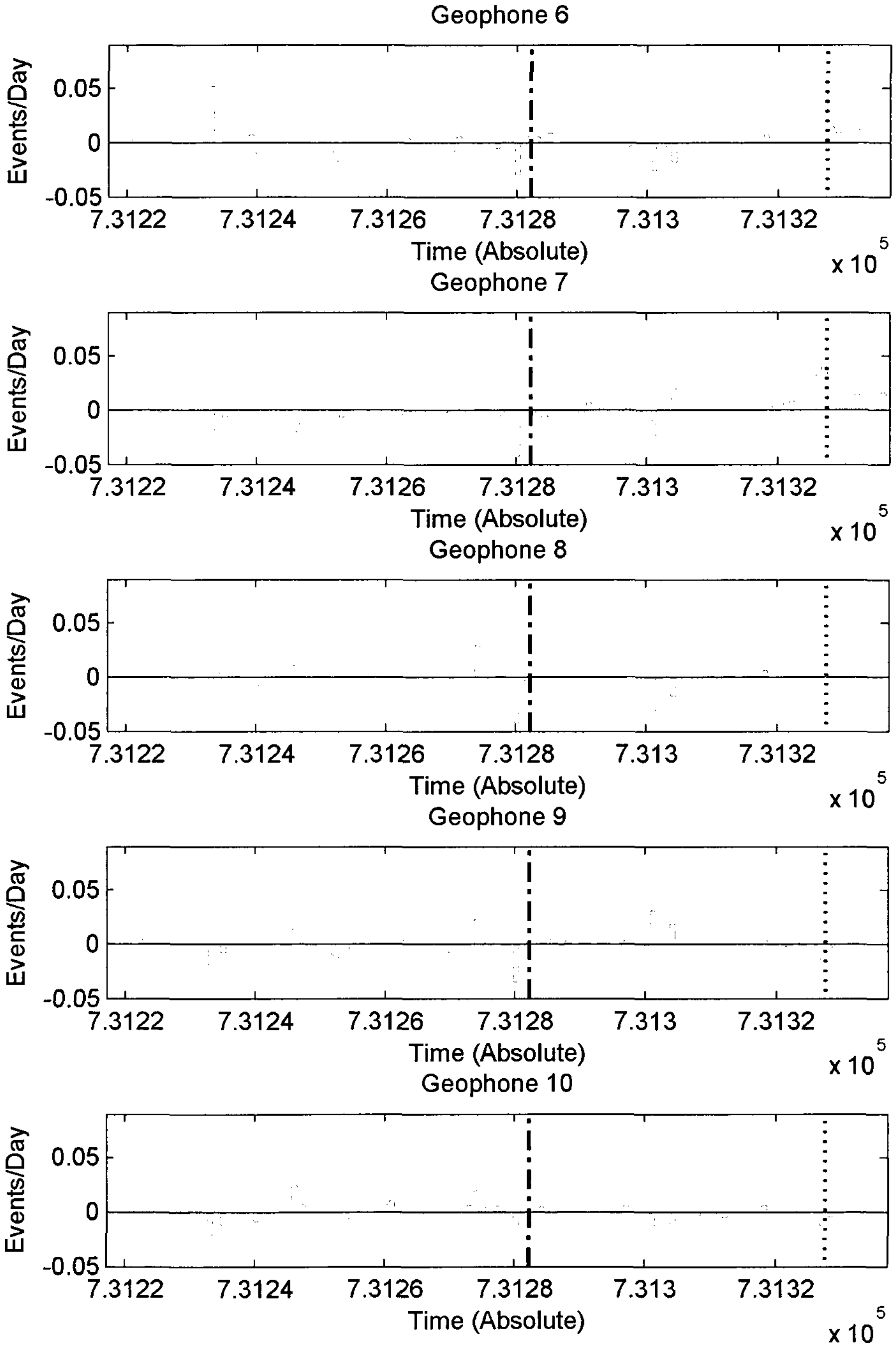


Figure 5.10: Similarly to Fig. 5.9, the quantity $E_+ - E_-$ is plotted to show the relative difference between the classified fracture events per day from the non-fracture events per day for geophones 6 through 10.

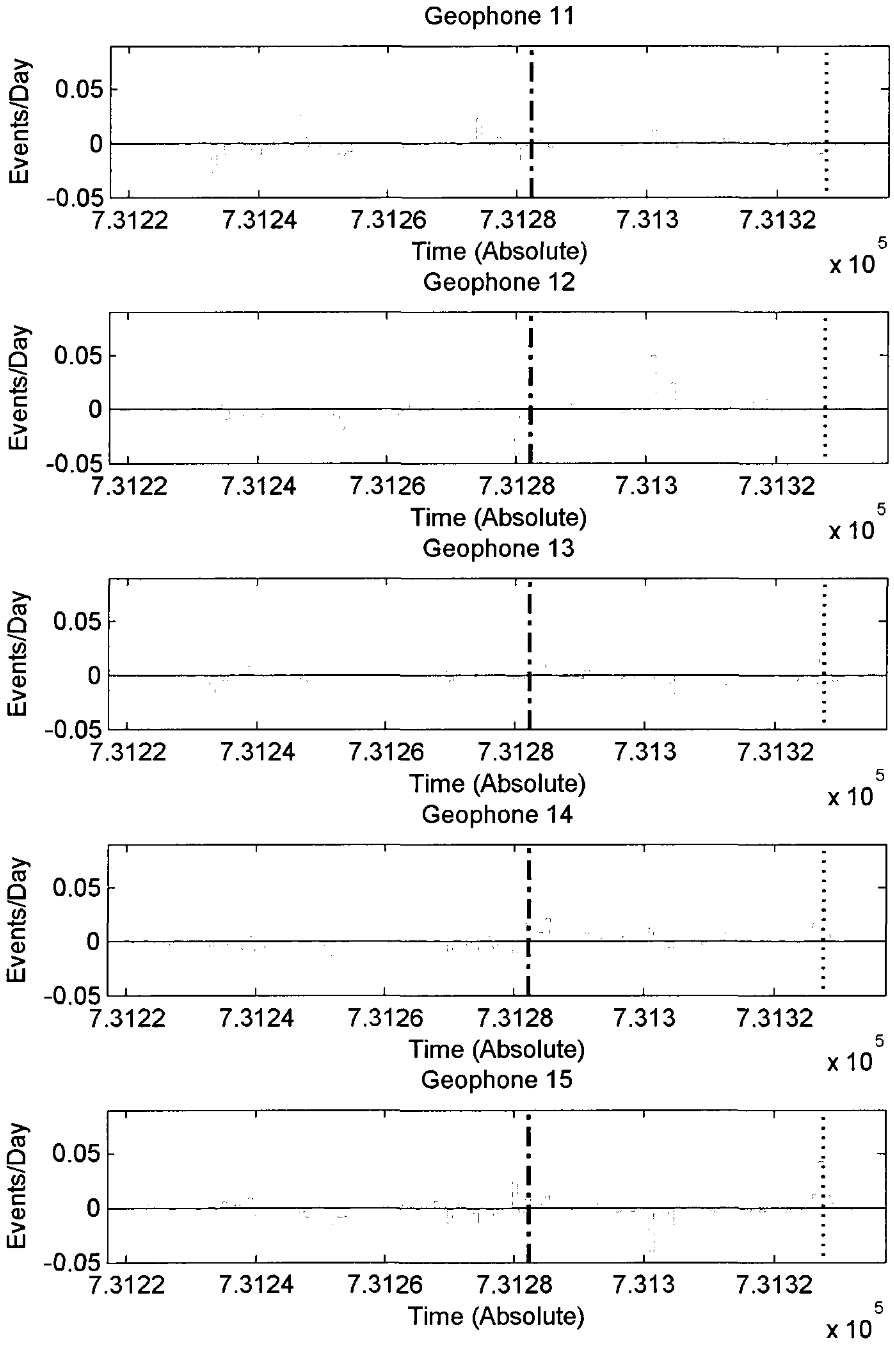


Figure 5.11: Similarly to Figs. 5.9 and 5.10, the quantity $E_+ - E_-$ for geophones 11 through 15 is shown.

which recorded events correspond to particular types of seismic activity, such as roof falls, or to electrical noise, or to background mining noise. Clustering methods assign the members of such unlabeled data sets to fuzzy (or probabalistic) classes [34] and have been particularly useful for partitioning time series data [35]. There are a variety of clustering methods available. The method used in the work described below uses the well-known k-means clustering algorithm [36, 37].

Given a set of n objects w_j

$$w_j \mid j = 1, \ldots, n$$

the k-means algorithm attempts to cluster the w_j into k paritions

$$S^i \mid (i = 1, \ldots, k) \& (k < n)$$

that minimizes the squared error function

$$E = \sum_{i=1}^{k} \sum_{w_j \in S^i} (w_j - \mu_i)^2$$

where μ_i is the mean value of $w_j \in S^i$.

The k-means algorithm is iterative. As an example, consider the k-means procedure provided by Martinez *et al.* [38]:

1. Begin with a partition of k groups, possibly chosen at random or user-specified.

2. Calculate the distance between each point w^i and every cluster centroid.

3. Move the observation w^i from group r to group s if

$$\frac{n_r}{n_r - 1}(d^i_r)^2 > \frac{n_s}{n_s + 1}(d^i_s)^2$$

where n_r, n_s are the number of observations in sets r and s respectively, and d_r^i, d_s^i are the Euclidean distance between w^i and the centroid cluster in sets r,s.

4. If there are several sets that satisfy the inequality above, then choose the set s that has the smallest value of

$$\frac{n_s}{n_s+1}(d_s^i)^2$$

.

5. Repeat 2 through 4 until no observations are moved.

In this example, the Euclidean distance is the distance measure used. However, other distance measures can be used [39].

The actual k-means algorithm employed in MATLAB proceeds in two phases [40, 41]. First, batch updates are performed to move entire groups of observations closer to their centroid, followed by an updating of the centroid location. Then, individual points are reassigned to reduce the squared error, with centroid locations updated after each individual reassignment. The algorithm ends when a local minimum is achieved.

5.6 Results and Discussion

As discussed in Section 5.5, the clustering must be initialized with a number of clusters and the initial centroid locations. Different values of number of clusters and distance metrics with random starting cluster centroids were tested on the features selected from the $13,228$ different observations from the Springfield Pike mine. One

Table 5.2: Starting k-means cluster matrix containing centroid locations for each cluster (rows) and feature (columns)

Cluster	Features							
1	0.9089	0.0783	-0.143	-0.1708	-0.1707	-0.1707	-0.1612	-0.1708
2	0.9331	-0.08	-0.1258	-0.1463	-0.1463	-0.1462	-0.1422	-0.1463
3	0.5847	0.6252	-0.1559	-0.2142	-0.2141	-0.2141	-0.1975	-0.2141
4	0.2289	0.857	-0.149	-0.1899	-0.19	-0.1899	-0.1772	-0.1899
5	0.8121	0.3307	-0.1524	-0.2009	-0.2009	-0.2008	-0.1867	-0.2009
6	-0.0508	0.9287	-0.1292	-0.1516	-0.1517	-0.1516	-0.1422	-0.1516

that appeared particularly promising used a non-Euclidean distance metric, which measures one minus the sample correlation of the points in the cluster, and 6 different clusters were used. The resulting cluster centroids are shown in Table 5.2. This cluster configuration was chosen because the behavior of the features near the roof fall dates is significantly different from earlier times. Figure 5.12 shows a comparison of those observations labeled as cluster 6 by the k-means algorithm using centroid positions in Table 5.2. The results from two different geophones are shown in terms of the maximum amplitude of that waveform versus the time it occurred. Geophone 2 was observed to have many high amplitude fracture events during the March 7 roof fall, and similarly for geophone 6 relative to the April 21 roof fall. Those dates are both marked on the figure with dotted lines. As Fig. 5.12 shows, the clustered data identified as class 6 tends to group strongly around the roof fall dates.

From this information, a roof fall index (RFI) was determined to indicate the severity of seismicity within a unit of time. The RFI the number of events ($\#w^i$) in a period of time T registering in one geophone that satisfy the following properties:

$$\mathrm{RFI}(g,T) = \#w^i \mid \{(c^i = 6)\&(w_2^i < 2.1 \times 10^4)\&(w_1^i < 2500)\&(t^i \in T)\} \qquad (5.14)$$

where g represents geophone number, T is a period of time, w^i is the ith seismic

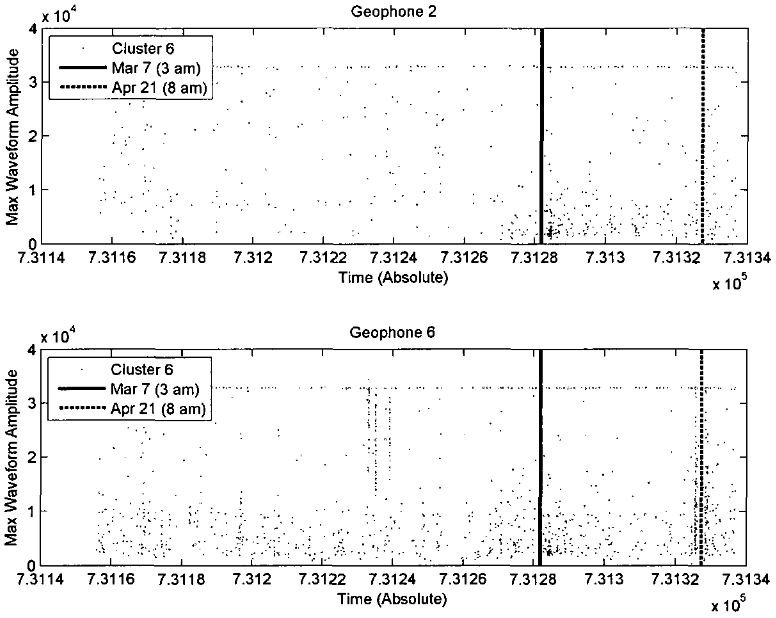


Figure 5.12: The graphs show maximum waveform amplitude versus time for events labeled as cluster 6 in the k-means algorithm. Note the increased activity around the roof fall dates.

event occurring at time t^i, and w^i_j is the jth element of the feature vector for event w^i. Because RFI is calculated for periods of time T, it is a system that is more sensitive to high volume seismic activity occurring within a short time window, and a higher number of events per hour has been shown to be a useful metric in the past [10]. The other parameters taken into account include the amplitude (w_2) and number of ridges (w_1), which relate to the unique shape of a fracture event.

Figure 5.13 shows the RFI index for geophones 2 and 6 calculated using a period of $T = 30$s. The entire 6 month period of the study has been examined here and two large peaks in RFI are visible at the times the roof falls occurred.

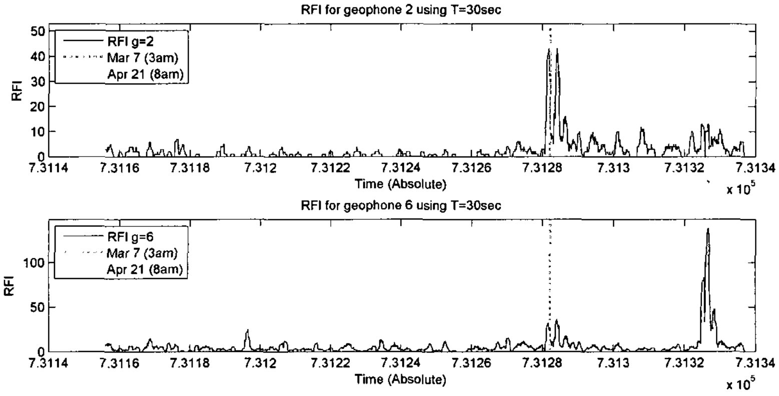


Figure 5.13: The RFI for geophones 2 and 6 is shown here with the roof fall times indicated. A period of 30 seconds was used to generate the RFI.

Figure 5.13 only shows roof fall index for the two geophones that show significant activity at the occurrence of the roof falls. In order to be robust, we must calculate RFI at all the geophones for all time. Then, we find a threshold where RFI is large in order to concentrate on the most likely seismically active areas. Whenever RFI is large for several geophones at once, a roof fall can be predicted. Figure 5.14 shows the number of geophones that recorded RFI > 30 during the period under study. In this figure, the only time more than two geophones have recorded RFI > 30 is during the roof falls, indicated with dotted lines. Table 5.3 compares the times when visual

Table 5.3: The first indication of roof falls is compared. The visual inspection method describes whenever the roof falls were first noticed in the mine. The time the roof falls were first noticed is compared with the first time three geophones register RFI > 30.

Roof fall prediction method	
Visual	RFI
3/7/02 03:00:00	3/6/02 11:41:00
4/21/02 08:00:00	4/18/02 15:20:30

inspection indicated roof falls versus the first time when three geophones registered RFI > 30.

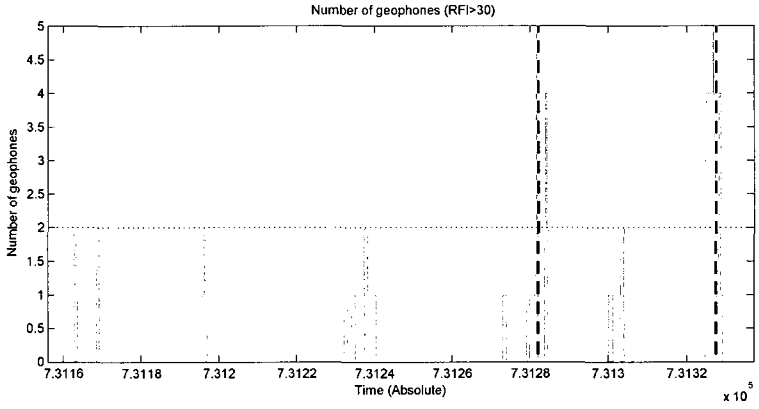


Figure 5.14: The number of geophones with RFI > 30 is plotted versus time.

As shown in Fig. 5.14 and Table 5.3, the Roof Fall Index can successfully be used to predict roof falls. In Fig. 5.14, whenever more than two geophones registered a RFI > 30 indicated by a horizontal green line, visual roof fall signs occurred within a few hours to a few days, indicated by a vertical red line. Table 5.3 shows the amount of warning time between the first time more than two geophones registered RFI > 30 and the first visual signs of roof falls occuring on the March 7, 2002 and April 21, 2002 roof falls. These results show that, for the data used here and the clustering matrix shown in Table 5.2, the RFI prediction method gives more of a warning time before roof falls than even the first visual signs, enough time to vacate the hazardous area or to shore up the roof to prevent the roof fall.

While the RFI index gives sufficient warning time before a roof fall, it can only predict in terms of generalities where the roof fall will occur. The roof fall is predicted whenever more than two geophones pass the RFI $>$ 30 threshold. The geophones that cross that threshold are the ones closest to the seismic activity, and therefore the ones closest to where the roof fall will occur. Figure 5.14 shows that more than two geophones crossed the RFI threshold before the roof fall was first noticed by visual inspection. The specific geophones that first crossed the threshold before the roof falls include geophones 1 through 4 for the March 7 roof fall and geophones 3 through 6 and 15 for the April 21 roof fall. The position of these geophones relative to the roof fall locations is shown in Figure 5.1.

In addition, the RFI method can be implemented in real-time in a mine. Since RFI is a moving average, the time interval involved can be adjusted to a time interval that makes sense for live updates for microseismic mine monitoring. Changing the time interval of RFI does not change the results displayed in Figure 5.14, even when the period used to calculate RFI is reduced to $T = 1$ hour.

5.7 Conclusion and Future Work

Microseismic monitoring has been employed most often to measure seismic activity using source parameters. In this study, we have shown that fuzzy classification techniques can simplify the problem of judging seismic magnitude significantly. In particular, the methods described above can be used not only to measure seismic activity but to predict roof falls. The features selected for classification include features specific to the shape of the waveform as well as features related to spectral source parameters. The k-means clustering technique resolved significant events and motivated the calibration of the Roof Fall Index. Events with a Roof Fall Index above 30 predict roof falls by as much as several hours or even a few days, which can be used to mitigate mining hazards.

Bibliography

[1] C. Mark, G. M. Molinda, and L. M. Burke, "Prevengin falls of ground in coal mines with exceptionally low-strength roof: Two case studies," in *Proceedings of the* 23^{rd} *International Conference on Ground Control in Mining*, Morgantown, WV, Aug. 3-5 2004, pp. 327–333.

[2] National Institute for Occupational Health and Safety, "Reducing ground fall hazards in coal mines with low strength roof," http://www.cdc.gov/niosh/nas/mining/potentialintermediateoutcome62.htm.

[3] J. H Reginald Hardy, *Acoustic Emission/Microseismic Activity.* Exton, PA: A A Balkema, 2003, vol. 1.

[4] S. J. Gibowicz and A. Kijko, *An Introduction to Mining Seismology.* New York: Academic Press, 1994.

[5] A. T. Iannacchione and T. S. Bajpayee, "Forecasting roof falls with monitoring technologies - a look at the Moonee Colliery experience," in *Proceedings of the* 24^{th} *International Conference on Ground Control in Mining*, Morgantown, WV, Aug. 2-4 2005, pp. 44–51.

[6] W. J. Gale, K. A. Heasley, A. T. Iannacchione, P. L. Swanson, P. Hatherly, and A. King, "Rock damage characterisation from microseismic monitoring," in

Proceedings of the 38^{th} *US Rock Mechanics Symposium*, Washington, DC, July 7-10 2001, pp. 1313–1320.

[7] A. Iannacchione, T. Batchler, and T. Marshall, "Mapping hazards with microseismic technology to anticipate roof falls: A case study," in *Proceedings of the* 23^{rd} *International Conference on Ground Control in Mining*, Morgantown, WV, Aug. 3-5 2004, pp. 327–333.

[8] J. van der Merwe, "In-situ investigation into the causes of falls of roof in south africa collieries," in *Proceedings of the* 20^{th} *International Conference on Ground Control in Mining*, Morgantown, WV, 2001, pp. 105–118.

[9] R. Berquist and M. Willis, Private Communication, March 2009, email with Rick Berquist on Mar 26, 2009.

[10] A. T. Iannacchione, T. E. Marshall, and L. J. Prosser, "Failure characteristics of roof falls at an underground stone mine in southwestern pennsylvania," in *Proceedings of the* 20^{th} *International Conference on Ground Control in Mining*, Morgantown, WV, Aug. 7-9 2001, pp. 119–125.

[11] A. T. Iannacchione, P. R. Coyle, L. J. Prosser, T. E. Marshall, and J. Litsenberger, "The relationship of roof movement and strata-induced microseismic emissions to roof falls," *Mining Engineering*, pp. 53–60, 2004.

[12] A. T. Iannacchione, L. M. Burke, and M. C. Chapman, "Characterising roof fall signatures from underground nines," in *Proceedings of the* 6^{th} *International Symposium on Rockburst and Seismicity in Mines, Controlling Seismic Risk*, Australia, Mar. 9-11 2005, pp. 619–629.

[13] M. Ge, M. Mrugala, and A. T. Iannacchione, "Microseismic monitoring at a limestone mine," *Geotechnical and Geological Engineering*, 2008.

[14] J. H. Justice, D. J. Hawkins, and G. Wong, "Multidimensional attribute analysis and pattern recognition for seismic interpretation," *Pattern Recognition*, vol. 18, no. 6, pp. 391–407, 1985.

[15] P. L. Love and M. Simaan, "Segmentation of a seismic section using image processing and artificial intelligence techniques," *Pattern Recognition*, vol. 18, no. 6, pp. 409 – 419, 1985.

[16] Z. Zhang and M. Simaan, "A rule-based interpretation system for segmentation of seismic images," *Pattern Recognition*, vol. 20, no. 1, pp. 45 – 53, 1987.

[17] K. Huang and K. Fu, "Syntactic pattern recognition for the recognition of bright spots," *Pattern Recognition*, vol. 18, no. 6, pp. 421 – 428, 1985.

[18] K. Y. Huang, K. S. Fu, T. H. Sheen, and S. W. Cheng, "Image processing of seismograms: (a) hough transformation for the detection of seismic patterns; (b) thinning processing in the seismogram," *Pattern Recognition*, vol. 18, no. 6, pp. 429 – 440, 1985.

[19] H. Liu, "A rule-based system for automatic seismic discrimination," *Pattern Recognition*, vol. 18, no. 6, pp. 459 – 463, 1985.

[20] R. H. Shumway, "Time-frequency clustering and discriminant analysis," *Statistics & Probability Letters*, vol. 63, no. 3, pp. 307 – 314, 2003.

[21] D. R. Hay and R. W. Y. Chan, "Use of artificial intelligence methods in the analysis of microseismic data," in *Fifth Conference on Acoustic Emission/Microseismic Activity in Geologic Structures and Materials.* Clausthal-Zellerfeld, Germany: Trans Tech Publications, 1995.

[22] N. Feknous, G. Ballivy, and Z. Piasta, "Identification of stress levels in rock with acoustic emission assisted by pattern recognition techniques," in *Proceedings of Conference on Acoustic Emission/Microseismic Activity in Geologic Structures and Materials.* Clausthal-Zellerfeld Germany: Trans Tech Publications, 1995, pp. 135–149.

[23] S. Tsukada and K. Ohtake, "Application of pattern recognition to seismic event discrimination," in *Methods and Applications of Signal Processing in Seismic Network Operations*, ser. Lecture Notes in Earth Sciences. Berlin: Springer-Verlag, 2003, vol. 98, pp. 221–234.

[24] R. F. Kubichek and E. A. Quincy, "Statistical modeling and feature selection for seismic pattern recognition," *Pattern Recognition*, vol. 18, no. 6, pp. 441 – 448, 1985.

[25] ——, "Identification of seismic stratigraphic traps using statistical pattern recognition," *Pattern Recognition*, vol. 18, no. 6, pp. 449 – 458, 1985.

[26] A. T. Iannacchione and P. Coyle, "An examination of the loyalhanna limestone's structural features and their impact on nining and ground control practices," in *Proceedings of the* 21^{st} *International Conference on Ground Control in Mining*, Morgantown, WV, Aug. 6-8 2002, pp. 218–227.

[27] J. L. Ellenberger and T. S. Bajpayee, "An evaluation of microseismic activity associated with major roof falls in a limestone mine: A case study," in *2007 SME Annual Meeting and Exhibit*, Denver, Colorado, 2007, pp. 1–5, preprint 07-103.

[28] J. Hou and M. K. Hinders, "Dynamic Wavelet Fingerprint identification of ultrasound signals," *Materials Evaluation*, vol. 60, no. 9, pp. 1089–1093, 2002.

[29] J. N. Brune, "Tectonic stress and the spectra of seismic shear waves from earthquakes," *Journal of Geophysical Research*, vol. 75, no. 26, pp. 4997–5009, Sept. 1970.

[30] ——, "Correction," *Journal of Geophysical Research*, vol. 76, no. 20, p. 5002, July 1971.

[31] A. J. Mendecki, Ed., *Seismic Monitoring in Mines.* New York: Chapman & Hall, 1997.

[32] D. J. Andrews, "Objective determination of source parameters and similarity of earthquakes of different size," in *Earthquake Source Mechanics*, ser. Geophysical Monograph 37, S. Das, J. Boatwright, and C. H. Scholz, Eds. Washington, DC: American Geophysical Union, 1986, pp. 259–267.

[33] E. B. Grant, "Apparent stress estimates of seismic events associated with underground mining operations in sedimentary rock units: Implications for improving underground roof collapse warning systems," Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, July 2006.

[34] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, ser. The Handbooks of Fuzzy Sets Series, D. Dubois and H. Prade, Eds. Boston: Kluwer Academic Publishers, 1999.

[35] T. W. Liao, "Clustering of time series data–a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857 – 1874, 2005.

[36] J. A. Hartigan, *Clustering Algorithms.* New York: Wiley, 1975.

[37] J. A. Hartigan and M. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[38] W. L. Martinez and A. R. Martinez, *Computational Statistics Handbook with MATLAB.* Chapman & Hall/CRC Press, 2001.

[39] W. Fehlman and M. K. Hinders, *Mobile Robot Navigation with Intelligent Infrared Image Interpretation*, ser. Springer Tracts in Advanced Robotics. New York: Springer, 2009.

[40] G. A. F. Seber, *Multivariate Observations.* Hoboken, NJ: John Wiley & Sons, Inc, 1984.

[41] H. Spath, *Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples.* New York: Halsted Press, 1985, translated by J Goldschmidt.

Chapter 6

Specific Emitter Identification (SEI) for Radio Frequency Identification (RFID) Tags

Originally developed for radar sources, Specific Emitter Identification (SEI) uses unintentional modulations of the electromagnetic signal of RF emitters to identify individual sources of signals as unique from emitters of the same type. In defense fields, SEI allows for the identification and tracking of physical threats, while security applications prevent unauthorized access and cloning of sensitive devices. Artificial intelligence techniques assist SEI by providing automatic recognition of these unique aspects of individual RF emitters. Radio Frequency Identification (RFID) tags are a common RF emitter used in commercial and military applications to track supplies and are also present in credit cards and passports to allow for automatic recognition or monetary transfers. Despite advances in RFID cryptography, RFID tags can still be easily cloned and tracked. Here we implement SEI to identify individual RFID tags. Features are extracted using the Dynamic Wavelet Fingerprint (DWFP) and supervised pattern classification techniques are used to identify unique RFID tags with up to 98% accuracy.

6.1 Introduction

With the ubiquity of RFID tags in modern life, security of RFID tags is a primary concern. In particular, cloning of RFID tags in devices such as car keys or ID badges can lead to serious security breaches. The only components required to crack RFID security are a pair of challenge/response values, which can be obtained by an activity as simple as eavesdropping or as complicated as reverse engineering [1]. For example, researchers at Johns Hopkins University and RSA Laboratories were able to apply reverse engineering to simulate an ignition key for their own car and to clone their own SpeedPass^{TM} token, used to purchase gasoline [2]. Many of the proposed fixes to RFID security involves stronger encryption of the RFID code, employing public key cryptography, or restricting physical access to reading the RFID device, such as passport wallets that block RF fields. However, none of these solutions solve the possibility of sophisticated security breaches such as cloning RFID devices at the factory, and they are only applicable for more expensive devices. Cheaper devices like ID badges would still be vulnerable.

Instead, we apply Specific Emitter Identification (SEI) to the problem of RFID security. SEI refers to the ability to associate a radio frequency waveform with the unique emitter of that signal [3]. The identifiable individual variations are small and unintentional and may arise from the manufacturing and/or tag-reading process. SEI has been investigated for RF fields such as radar [4–7], cell phones [8], wireless internet cards [9], and cognitive radio [10]. The features selected and techniques used vary in each case. Shieh and Lin [4] used basic signal features including frequency, pulse width, and pulse repetition interval in a neural network to improve recognition of RF emitters in radar warning receivers. Kawalec and Owczarek [5] used supervised pattern classification to identify radar sources with similar features, while Dudczyk *et al.* focused on analyzing the unintended emissions of electromagnetic energy from radar emitters to perform SEI. Meanwhile, Matuszewski [7] developed a knowledge-based sequence of decision rules using known facts about radar capabilities to aid radar SEI. A combination of supervised classification and clustering methods was used by Talbot *et al.* [8], in which unspecified features drawn from a variety of RF emitters with commercial applications for cell phones. Remley *et al.* [9] identified a specific transmitter in a wireless network by distinguishing between different wireless card manufacturers. Lastly, Kim *et al.* [10] used the basic radio signal features as well as a second-order measurement comparing spectral and cyclic frequencies and classified individual cognitive radios with hidden Markov models. However, no applications of SEI for RFID tags has been presented in the literature.

Our goal is to collect measurements from several individual RFID tags written with the same Electronic Product Code (EPC), which is like a barcode, to identify the specific source of the RFID signal. The measurements involved include simple properties of the signal as well as sophisticated features measured by wavelet fingerprinting. Pattern classification techniques are applied to differentiate the individual tag.

6.2 Pattern Classification for SEI

The application of pattern classification to specific RFID identification includes the following steps:

1. Sensing: the RFID tags are read and EPC codes from several tag-read events

(TREs) are gathered from each individual tag.

2. Feature extraction: measurements are made from each EPC code.

3. Feature selection: an optional step in which the subset of features is reduced, sometimes to make the algorithm more computationally efficient, and sometimes to remove irrelevant features.

4. Classification: the data set is split via a resampling algorithm into training and testing subsets. The classifier is trained on the training set and tested with the testing set. The actual form of the classifier can vary but for this application it is always discriminative in nature.

5. Decision: the predicted class of the testing set is finalized and the performance of the classifier is evaluated.

Formally [11], consider a two-class problem with labels ω_1, ω_2 with probabilities of each class occurring given by $p(\omega_1), p(\omega_2)$. Now consider a feature vector $\mathbf{x}$, which is the vector of measurements made by the sensing apparatus for one EPC. Then $\mathbf{x}$ is assigned to class ω_j whenever

$$p(\omega_j|\mathbf{x}) > p(\omega_k|\mathbf{x}), k \neq j \tag{6.1}$$

By using Bayes theorem, we can rewrite Eq. 6.1 as

$$p(\mathbf{x}|\omega_j)p(\omega_j) > p(\mathbf{x}|\omega_k)p(\omega_k), k \neq j \tag{6.2}$$

In this way, the sensed object associated with the feature vector $\mathbf{x}$ is assigned to the class with the highest likelihood.

Classification generally involves calculating those posterior probabilities $p(\omega_i|\mathbf{x})$ using some mapping. Four main types of classifiers will be applied to RFID SEI: linear and quadratic discriminant classifiers (LDC and QDC respectively), k-nearest-neighbor (kNN), and Support Vector Machines (SVM). The details of these classifiers can be found in any general pattern classification book, including [11] and [12].

For this application, the features selected for the feature vector will be optimized to detect one individual tag at a time (the classifier tag), and all the other tags (the testing tags) are tested against it one at a time. The class labels are either $\omega = 1$, declaring that the feature vector corresponds to an EPC from the classifier tag, or $\omega = -1$, when the feature vector does not correspond to an EPC from the classifier tag. This is a realistic structure for SEI applications such as an ID challenge, in which the RFID reader must decide if the ID being presented corresponds to the exact same individual RFID tag as is stored in memory instead of a cloned tag with the same EPC. Later sections will describe the feature extraction, feature selection, classification, and decision-making processes involved in the RFID pattern classification configuration.

6.3 Materials and Methods

Two types of passive RFID tags with different types of antennae and chips were examined in this study. Both are configured to fit standards for EPC Class 1 Gen 2 [13]. Their full name and abbreviation are listed below:

1. *AD*: Avery-Dennison AD 612

2. *DN*: Avery-Dennison Runway Gen 2

There were 25 individual tags of each type that were purchased for this study. Figure 6.1 shows the experimental setup in which the tags were read. In this proce-

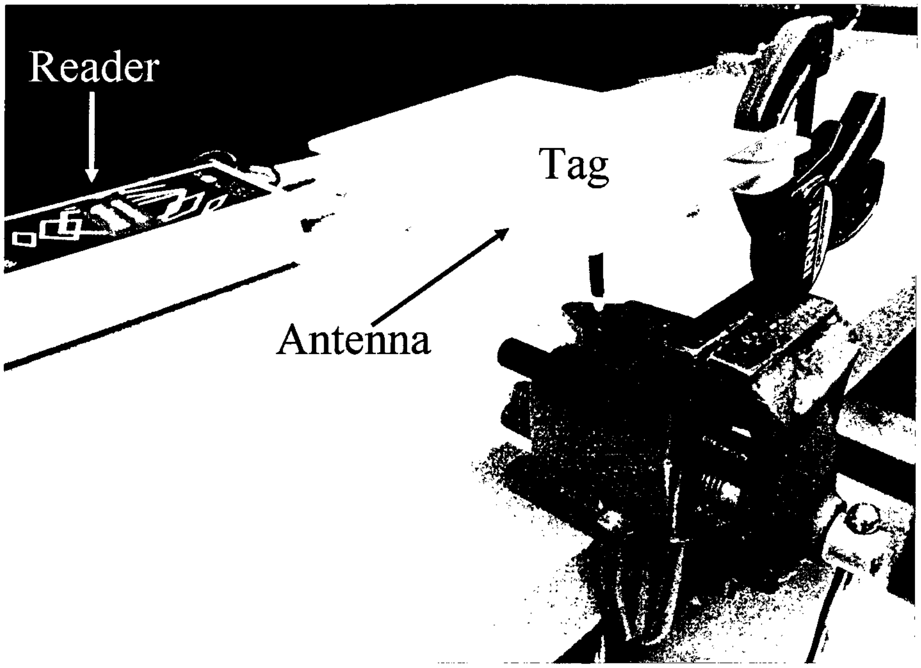


Figure 6.1: The experimental setup used to read the RFID tags is shown, with the tag reader, antenna, and tag displayed. The gap between the antenna and the tag improved the quality of the RFID signal.

dure, the same code was written onto each tag with Thing Magic Mercury 5e RFID Reader[1] and read with an omnidirectional antenna (Laird Technologies[2]) through a vector signal analyzer operating at a 3.84 MHz sampling frequency. There were 8 seconds of data recorded for each tag and stored as a MATLAB format. Each individual tag of each type received a unique label, so that the AD tags were labeled $AD01, AD02, \ldots, AD25$ and the DN tags were likewise labeled $DN01, DN02, \ldots, DN25$. Later, it was noted that the data for 3 of the DN-type tags failed to save properly, so there were only 47 total tags examined.

In most applications using RFID tags, it is the content of the EPC code sent by the reader that is of interest. Here, however, the goal is to identify each individual tag despite the fact that the tags were purchased from the same manufacturer and written with the same EPC code. Therefore, it is the raw RFID data signal itself that is of interest instead of the content of the EPC.

6.4 EPC Signal Extraction

The first step of classifying the RFID tags involves being able to automatically extract the EPC code in software from the 8-second long tag-to-reader communication, or tag-read events (TREs) performed when the data was collected. The query sequence [13] is shown in Figure 6.2, in which the reader repetatively queries for the presence of the RFID tag, the tag eventually responds with a string of 16 random numbers, which is acknolwedged by the reader. Then the EPC code is sent to the reader, and the process repeats, resulting in many repetitions of the EPC code in the received signal.

We describe next a signal processing method to extract the EPC code from the TREs, which works for both tag types, but it does not extract every single EPC from

[1] Cambridge, MA (http://www.thingmagic.com/)

[2] St Louis, MI (http://www.lairdtech.com/)

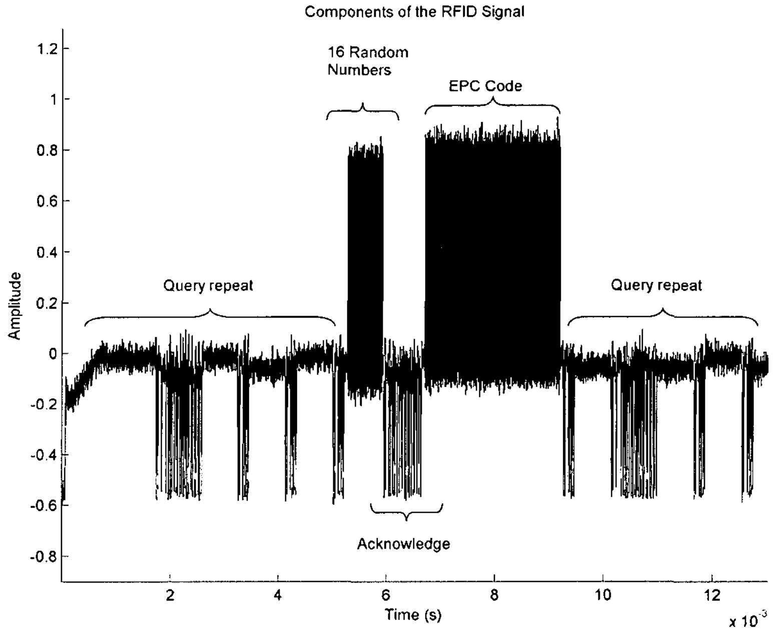


Figure 6.2: The relevant features of the RFID signal have been labeled to show the communication between the tag and the reader.

the many TREs within the recorded period of time. Instead, consecutive windows of the RFID signal are examined for large cross-correlation with a manually-extracted component of a tag from the same manufacturer. The steps of the EPC extraction algorithm are listed below. Unless otherwise mentioned, the preprocessing steps include finding the modulus of the complex-valued RFID signal, removing the DC offset, and normalizing.

1. Manually extract a region of interest from one individual tag of each manufacturer type. The region of interest includes the 16 random digits, acknowledgement, and EPC code, as shown in Fig. 6.2.

2. Starting at the beginning of the signal, examine the first N bits of the signal, where N is the fixed window size. For the experiments in this chapter, $N = 1 \times 10^5$ was used, but the window size should at least be larger than the region

of interest.

3. Calculate the cross correlation of the modulus of the windowed signal with the manually-extracted region of interest. If the cross correlation is below a certain threshold T, then the algorithm declares that no EPC was found in that window. Here, $T = 250$ was found to be sufficient to exclude windows that do not contain an EPC code.

4. If an EPC is declared present in the window, it is extracted by searching for the low amplitude nodes on either side of the EPC. The complex-valued signal corresponding to those nodes is then stored in an array. Later, this signal can be modified by calculating the modulus, the instantaneous frequency, or the phase of the complex-valued signal.

The figures below show results of the EPC extraction for each of the two different manufacturer tag types. Figure 6.3 shows the EPC extracted related to the AD tags, and Figure 6.4 shows the EPC extraction for a DN-type tag. Note that the EPCs are presented differently in each case: the AD tags record the EPC below the baseline activity, and the DN tags are recorded with the EPC above the baseline activity. The EPC extraction routine does not operate differently between the two tag types beyond loading the correct version of the manually-extracted query region. Also, since the windowing is rather arbitrary, a different number of EPCs are recorded for each tag. The extraction algorithm recorded as few as 22 or as many as 114 EPCs collected per tag.

The RFID signal s is a complex-valued envelope composed of real and imaginary parts r and c. We can find the amplitude a, phase θ, and instantaneous frequency f_i using the following formulae [14]:

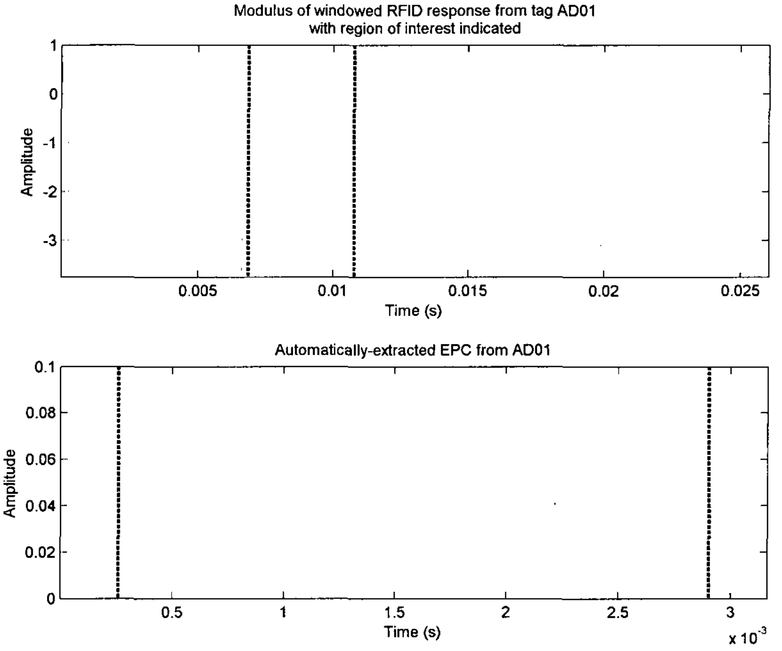


Figure 6.3: The results of extracting EPC codes from RFID interrogation of an AD-type tag are shown. The first plot indicates the region of interest containing the EPC code, and the last plot shows the resulting extracted EPC.

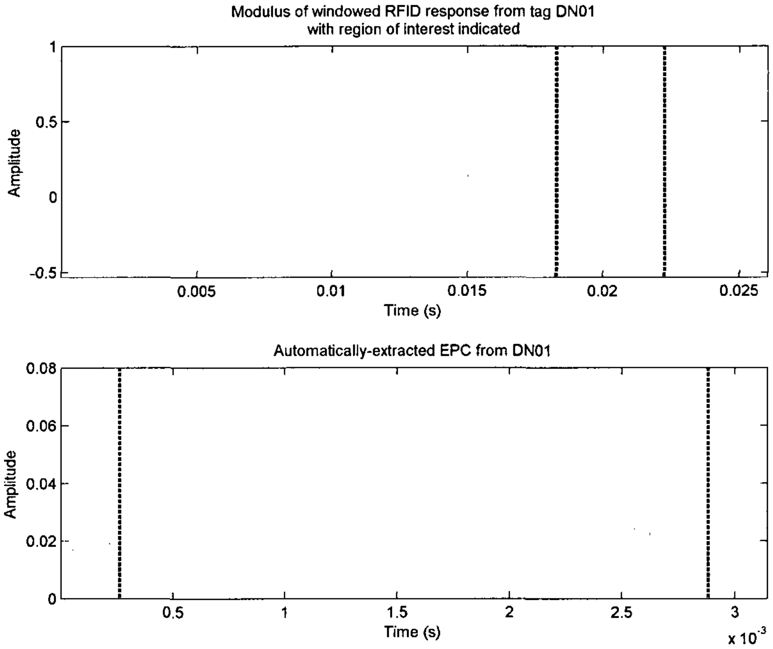


Figure 6.4: Similarly to Fig. 6.3, the extraction of EPC codes from a DN-type tag is displayed.

$$\begin{aligned} s(t) &= r(t) + ic(t) \\ \alpha(t) &= \sqrt{r^2(t) + c^2(t)} \\ \theta(t) &= tan^{-1}\left(\frac{r(t)}{c(t)}\right) \\ f_i(t) &= \frac{1}{2\pi}\frac{d}{dt}\theta_h(t) \end{aligned} \qquad (6.3)$$

where $\theta_h(t)$ is $\theta(t)$ unwrapped whenever the phase passes through multiples of 2π. The signal processing step of reducing complex-valued s to one of α, θ, or f_i will hereafter be referred to as EPC compression. Figure 6.5 compares the different EPC compression results on a complex signal.

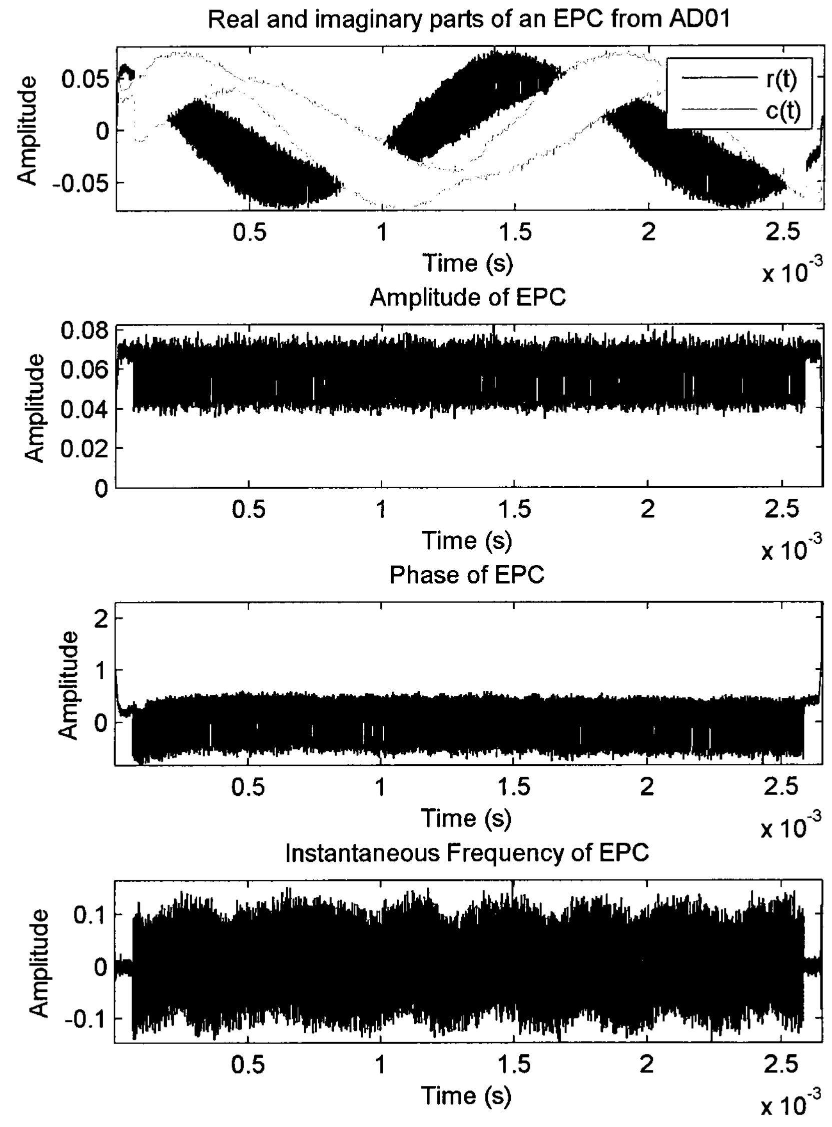


Figure 6.5: The different EPC compression techniques are compared here, displaying the real and imaginary parts $r(t)$ and $c(t)$, the amplitude $\alpha(t)$, the phase $\theta(t)$ (with linear dependence removed), and the instantaneous frequency $f_i(t)$.

6.5 Feature Extraction

For each tag type (AD or DN), the EPC extraction routine results in N-many different EPCs, $s_j(t)$, corresponding to an individual tag $\tau_j = 1, \ldots, 25$, $j = 1, \ldots, N$. The next step is to extract features and select M of them, so that an element of the feature vectors will be denoted $x_{j,k}$, $k = 1, \ldots, M$. There are three different methods employed to extract features: Dynamic Wavelet Fingerprinting (DWFP), Wavelet Packet Decomposition (WPD), and statistical methods. Before the DWFP, the $s_j(t)$ are filtered, but the other two feature extraction techniques are performed on unfiltered waveforms.

6.5.1 DWFP

To extract features from the EPCs, the DWFP technique (Fig. 6.6) applies a wavelet transform on the original time domain waveform, which results in "loop" features that resemble fingerprints. It has previously shown promise for a variety of applications.

Before going through the fingerprinting process, each EPC $s_j(t)$ is first low-pass filtered by applying a stationary wavelet transform and removing the first 5 details. Then, the first step of the DWFP (Fig. 6.6a-b) involves applying a wavelet transform on each of the waveforms.

The continuous wavelet transform can be written

$$C(a, b) = \int_{-\infty}^{+\infty} s(t)\overline{\psi_{a,b}(t)}dt \tag{6.4}$$

Here, $s(t)$ represents a square-integrable 1D function, and $\psi(t)$ represents the mother wavelet. The mother wavelet is scaled in frequency (f) and transformed in time (t) using $a, b \in \mathbb{R}$, respectively, in order to form the $\psi_{a,b}(t)$ in Equation 6.4. The wavelet

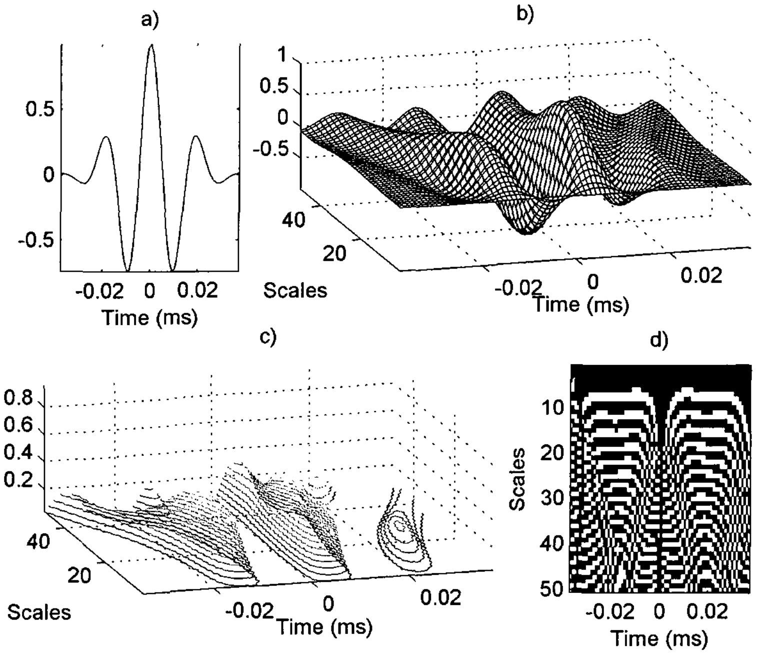


Figure 6.6: The DWFP technique [15] begins with a) the ultrasonic signal, where it geneates b) wavelet coefficients indexed by time and scale, where scale is related to frequency. Then c) the coefficients are sliced and projected onto the time-scale plane (d). The final binary image is used to select features for the pattern classification algorithm.

transform on a single waveform (Fig. 6.6a) results in wavelet coefficients (Fig. 6.6b). Then, a slicing algorithm is applied to create an image analogous to the gradient of the wavelet coefficients in the time-scale plane, resulting in a binary image, $I(a,b)$:

$$s(t) \xrightarrow{DWFP(\psi_{a,b})} I(a,b) \tag{6.5}$$

We used mother wavelets that previously showed promise for other applications, including Debauchies 3 (db3), Symelet 5 (sym5), and Meyer (meyr). The resulting image I contains fingerprint-like binary contours of the initial waveform $s_k(t)$.

Image Processing

The next step is to apply image processing routines to collect properties from each fingerprint object in each waveform. First, the binary image I is labeled with the 8-connected objects, allowing each individual fingerprint in I to be recognized as a separate object using the procedure in Haralick and Shapiro [16]. Next, properties are measured from each fingerprint. Some of these properties include counting the on- and off-pixels in the region, but many involve finding an ellipse matching the second moments of the fingerprint and measuring properties of that ellipse, such as eccentricity. In addition to the orientation measure provided by the ellipse, another measurement of inclination relative to the horizontal axis was determined by Horn's method for a continuous 2D object [17]. Lastly, further properties were measured by determining the boundary of the fingerprint and fitting 2nd or 4th order polynomials. Table 3.14 in the appendix to this chapter summarizes the types of features gathered from the fingerprints.

The image processing routines result in fingerprint properties $D_{j,\nu}[t]$, where ν represents an index of the image processing-extracted fingerprint properties ($\nu =$

$1, \ldots, 16)$. These properties are discrete in time because the values of the properties are matched to the time value of the fingerprint's center of mass. Linear interpolation yields a smoothed array of property values, $D_{j,\nu}(t)$.

Fingerprint Feature Selection

Recall that the goal of applying DWFP to the RFID waveforms $s_j(t)$ is to find M-many features $x_{j,k}$, $k = 1, \ldots, M$, that can be used to successfully classify the tags τ_j, $j = 1, \ldots, N$. Because we are designing our classifier for one-versus-one classification, in which one testing tag will be compared against features designed to identify one classifier tag, we select features from the $D_{j,\nu}$ that have the best chance of correctly identifying the classifier tag. Towards this end, a Euclidean distance metric is used to indicate the most highly-separable interclass distance, and we first normalize $D_{j,\nu}$ so that we can compare the Euclidean distances for different fingerprint properties ν.

In a classifier designed to identify tag τ_L, $L \subset \{1, \ldots, N\}$, then the label of the $s_L(t)$ corresponding to that tag are positive ($\omega_L = 1$) while all the other tags negative label ($\omega_S = -1$), where $\tau_S \neq \tau_L$, $S \subset \{1, \ldots, N\}$, $S \cap L = \emptyset$. Let D_ν represent the maximum value of $D_{j,\nu}$ over all j for a fixed ν. To $x_{j,k}$ for each fingerprint property ν, we calculate the mean (μ) and standard deviations (σ) of the normalized $D_{j,\nu}(t)$ corresponding to the subsets L, S and select times t_m corresponding to a large difference in the mean and small standard deviation. The mean and standard deviation for the subset L are given by Eqns 6.6 and 6.7, and we can obtain $\mu_{S,\nu}$ and $\sigma_{S,\nu}$ by similarly summing over the fingerprint properties $D_{S,\nu}$.

$$\mu_{L,\nu}(t) = \sum_{l \in L} \frac{D_{l,\nu}(t)}{D_\nu} \tag{6.6}$$

$$\sigma_{L,\nu}(t) = \sqrt{\frac{1}{|L|}\sum_{l\in L}\left(\frac{D_{l,\nu}(t)}{D_\nu} - \mu_L(t)\right)^2} \tag{6.7}$$

The distance metric $d_\nu(t)$ is then the difference between the interclass means (Eq 6.8)

$$d_\nu(t) = |\mu_{L,\nu} - \mu_{S,\nu}| \tag{6.8}$$

with the interclass STD given by Eq 6.9

$$\sigma_\nu(t) = \max(\sigma_{L,\nu}(t), \sigma_{S,\nu}(t)) \tag{6.9}$$

Feature selection proceeds by finding t_m such that $d_\nu(t_m)$ is larger than a given threshold, while $\sigma_\nu(t_m)$ is smaller than another threshold. The actual technique used for the RFID applies a varying threshold, as in the pseudocode presented in Program 3.

Program 3 The feature selection scheme used to find promising DWFP features to use for classification.

```
tdetect=0;
WHILE (tdetect==0){
        FOR (sperct=0.3:0.05:0.75){
             FOR(dperct=0.9:-0.05:0.5){
                 IF (FIND (($\sigma_\nu$<sperct*$\max(\sigma_\nu)$) AND
                             ($d_\nu$> dperct*$\max(d_\nu)$))){
                     tdetect=1;
                 }
             }
        }
}
```

This method finds times t_m at which the interclass distance $d_\nu(t_m)$ is large for a given fingerprint property, ν, and is repeated for each ν, resulting in an array of $d_\nu(t_m)$. The largest normalized values of the distance metric can then be chosen between all the fingerprint properties. For each $m \in 1, \ldots, M$, the features selected $x_{j,k}$ correspond to the time values t_m for which $d_\nu(t_m)$ are the largest for all ν.

6.5.2 WPD

What follows is a brief description of Wavelet Packet Decomposition (WPD). For a full description of the procedure, see [18].

Wavelet Packet Decomposition (WPD) requires several parameters: a mother wavelet, ψ, and a number of levels to decompose the waveform, V. We used the same settings as the original authors, $\psi =$ db16 and $V = 6$, because other choices of ψ, V did not vary widely for our data set. WPD begins with a Wavelet Packet Transform (WPT) resulting in a tree of coefficients similarly to the wavelet transform. Next, the wavelet packet energy is calculated, which relates to the percent of energy at the terminal nodes of the WPT tree. For feature selection, the normalized energy for the two classes are inserted into a matrix, e_L and e_S, and singular value decomposition returns the eigenvalues with the highest energy. The WPT elements corresponding to those singular values are used as features. Usually only the highest energy is selected, but if two bins have the same singular values, then both energy bins are used as features.

6.5.3 Statistical Features

Several other features related to statistical calculations on the $s_j(t)$. These features are explained below. In what follows, $|s|$ refers to the length of $s_j(t)$.

1. The mean of the EPC

$$\mu_j = \frac{1}{|s|} \sum_t s_j(t)$$

Note: All other properties are calculated from the mean-removed signal

2. The maximum cross-correlation of $s_j(t)$ with another EPC from the same tag, $s_k(t)$, where $\tau_j = \tau_k$

$$\max(\sum_t s_j^*(t) s_k(t+\tau))$$

3. The variance of $s_j(t)$

$$\frac{1}{|s|-1}\sum_t (s_j(t) - \mu_j)^2$$

4. The shannon entropy

$$\sum_t s_j^2(t)\ln(s_j^2(t))$$

5. The second central moment (similar to the variance, but with a divisor of $|s|$ instead of $|s|-1$)

$$\frac{1}{|s|}\sum_t (s_j(t) - \mu_j)^2$$

6. The skewness

$$\frac{1}{\sigma_j^3|s|}\sum_t (s_j(t) - \mu_j)^3$$

7. The kurtosis

$$\kappa_j = \frac{1}{\sigma_j^4|s|}\sum_t (s_j(t) - \mu_j)^4$$

6.5.4 A Comment on Feature Selection

Some feature selection has already been performed on the extracted features. The DWFP process in particular uses a distance metric to select and rank features from different fingerprint properties to use in a classifier designed to identify a particular tag, and the number of DWFP features chosen in this way will be shown to affect the performance of the classifier. However, no other feature selection has been applied on the data set. For the classification explained below, the features used include a fixed number of DWFP features, as well as the statistical and WPD features. The reason for this is twofold: one, formal feature selection is time intensive, and because the classification is one-versus-one, would have to be performed separately for each classifier tag; and two, a future goal of the project is to adapt genetic algorithm techniques to design feature selection algorithms for the SEI project. So the feature selection involved at this stage of the project only involves the distance metric and number of features selected from the DWFP process.

6.6 Classifier maps

The actual classifiers tested include such well-known algorithms as quadratic discriminant classifier (QDC), linear discriminant classifier (LDC), k-nearest-neighbor (kNN) for several different choices of k, and linear support vector machines (SVM). The results of using these four types of classifiers on a simple example can be seen in Figure 6.7. In this example, features selected from EPCs from two tags, $AD01$ (the classifier tag) and $AD10$, are plotted against each other. Two features were selected to show in the plot, including the best DWFP feature (x_{DWFP}) and the kurtosis (κ), though often more features are required for best classification. The classifiers were applied to discriminate between the two tags, and the discriminant function is plotted as a black line.

For these classifier functions, we used MATLAB package routines as well as LDC and QDC from MATLAB with PRTools. In Figure 6.7a-b, we can see that the linear discriminant functions from PRTools does not have as much flexibility as the quadratic discriminant in prohibiting overlap between the classes. The quadratic discriminant is able to curl along the boundary between the classes. In Figure 6.7c, the results of

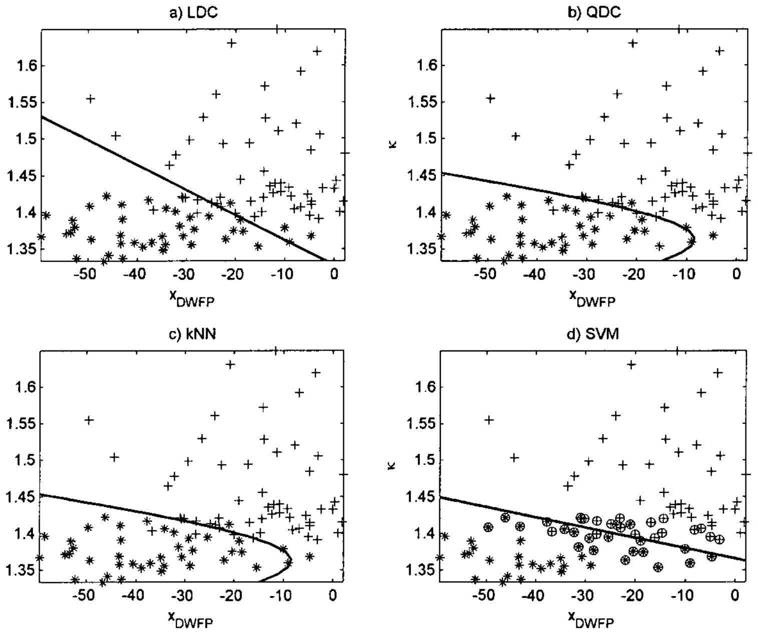


Figure 6.7: A comparison between the different types of classifiers used to classify the RFID tags are shown here, including a) LDC, b) QDC, c) kNN, and d) SVM classifiers. The example is composed of two features, the best DWFP feature (x_{DWFP}), and the kurtosis (κ). The features selected were optimized to identify $AD01$, and the two different markers represent the extracted features from EPCs extracted from $AD01$ (+) and $AD10$ (*). The decision boundary is also indicated in a black line. In the case of the SVM classifier (d), the circles represent the support vectors used in defining the maximal margin.

kNN classification on the example problem is plotted. The kNN function used here was from PRTools, which selects the optimal k-value from a leave-one-out error on the data, so the k-value used here was found to be 14. Note the similarity between the kNN and the QDC functions. Figure 6.7d shows the results of classifying the sample problem with support vector machines. The support vectors, which are the vectors used to make the decision about the maximal margin hyperplane, are shown as circles.

6.7 Classification Design and Algorithm

The procedure in Section 6.5 described the extraction of features $x_{j,k}$ for the EPCs $s_j(t)$ from the tags τ_j from one manufacturer. The features have been chosen specifically to detect one particular tag as an individual. As described in Section 6.2, the next step is to apply a mapping to calculate the posterior probabilities $p(\omega_i|\mathbf{x})$ in order to assign a predicted class to all the tags. The classifiers used have been described in Section 6.6.

We next present the method used to classify the EPCs from the individual tags of both types. In this algorithm, there are two main considerations. The problem of separating the $N \times M$ features $x_{j,k}$ into training and testing subset(s) is described in Section 6.7.1. In addition, the proportion of τ_j labeled $\omega = -1$ versus $\omega = 1$ must be controlled, since there may be as many as 65 times as many $\omega = -1$ labels as $\omega = 1$ labels in features $x_{j,k}$ designed to classify tag τ_L. Class imbalance is a well-known problem that may hinder the performance of some classifiers [19], and so undersampling is also considered in the RFID classifier design.

6.7.1 Training and Testing Data Sets

There are a variety of ways to select $j \in \{1, \ldots, N\}$ so that $X = x_{j,k}$ is split into training (X_R)and testing (X_T) sets. They are summarized in [20]. Of these, two will be used:

1. Hold-out: Split X into X_R and X_T once, usually $|X_R| = \frac{1}{2}|X|$. Train the classifier using X_R and test it using $X_T = X - X_R$. The results are pessimistically biased.

2. Bootstrap: Randomly select subset R from X and let $X_T = X - X_R$ as before, but repeat the random selections with replacement and average the resulting error.

6.7.2 Undersampling

In the features $x_{j,k}$ extracted to identify tag τ_L, the actual classes will be $\omega_i = 1$ whenever $\tau_i = \tau_L$ and $\omega_j = -1$ when $\tau_j \neq \tau_L$. Because there are as many as 24 tags that are not the same as the classifier tag τ_L, there is a class imbalance so that $|\omega_j = -1| \gg |\omega_j = 1|$. The level of imbalance affects the results depending on the complexity of the system the features were drawn from, but even a small imbalance can have a large effect on the results [19] [21]. Several solutions have been proposed: undersampling, in which $\omega_j = -1$ is sampled to a smaller size relative to $\omega_j = 1$; oversampling, in which more samples for τ_L are generated to make τ_L a closer size to $|\omega_j = -1|$; and cost-modifying, in which the cost of misclassification is adjusted so that the class imbalance is less likely to affect the results. We chose the undersampling method because cost-modification has failed to help with other class imbalance classification examples in the past, and the data set is small enough that we cannot be confident that oversampling would accurately replicate the underlying distribution of τ_L.

These aspects of classifier design are incorporated in the RFID classification algorithm and are represented by two variables, η and ρ, defined by Eqns 6.10 and 6.11:

$$\eta = \frac{|(\tau_j = \tau_S)\&(j \in T)|}{|\tau_j = \tau_L|} \tag{6.10}$$

$$\rho = \frac{|\omega_j = -1|}{|\omega_j = 1|}, j \in R \tag{6.11}$$

Here, L and S are subsets of $\{1, \ldots, N\}$ that indicate indices of $x_{j,k}$ corresponding with the classifier tag ($\tau_j = \tau_L$) and the·testing tag ($\tau_j = \tau_S$). Similarly, R and T represents subsets of $\{1, \ldots, N\}$ corresponding to the training set (R) and testing set (T). Therefore, η represents the fraction of EPCs from the testing tag that were withheld for T, with the rest inserted into R, so that $0 < \eta \leq 1$; and ρ represents the fraction of negative versus positive EPCs in R, so that $0 < \rho < (|\omega_j = -1|/|\omega_j = 1|)$ and $\rho \in \mathcal{Z}^+$. Effectively, η is the variable that controls how much data from the testing tag is kept aside for testing, while ρ controls the amount of undersampling.

The actual algorithm for the holdout method is presented in Program 4. The algorithm for the bagging method is similar except for each classifier tag τ_i and testing tag τ_j, the selection of training and testing sets (R, T) repeats until every feature $\mathbf{x}_k$ has been chosen for the training set at least once. In addition, the evaluation methods differ, which will be addressed in the next section.

To summarize the classifier design, there are several different variables to configure for the RFID classification. Each combination of these variables leads to a different classifier and different result.

1. Sampling method: holdout or bootstrap.

2. EPC compression method: amplitude, phase, or instantaneous frequency (Eqn 6.3), or feature vectors combining both amplitude and phase or amplitude and instantaneous frequency

3. Classifier map: QDC and LDC from MATLAB or PRTools, kNN using $k = 1, 2, 3$, and SVM.

4. Number of DWFP features: as described in Section 6.5.1, the features are selected by maximizing the distance measure, but the number of features selected is left free. We let the number of features be any one of $\{1, 5, 10, 15, 20, 50, 75, 100\}$.

5. Different values of η: we chose $\eta = 0.1 : 0.1 : 1$, where $\eta = 1$ represents all EPCs from the testing tag being withheld from the training set, the most strenuous possible test of the selected features.

6. Different values of ρ: we chose $\rho = 1 : 1 : 20$, and we also tested an option where all the EPCs not already reserved for the testing set were used for training (ρ =all).

The goal will be to find the best classifier with these different configurations.

6.8 Classifier Evaluation

The results of the classification can be written as a confusion matrix, $\mathcal{L}(i, j)$ where τ_i is the classifier tag and τ_j is the testing tag. The value of the confusion matrix differs depending on the sampling algorithm. For the holdout method, the value of the confusion matrix is the proportion of $y = 1$ labels for the EPCs of the testing tag in the testing set, or

$$\mathcal{L}(i, j) = \frac{|(\mathbf{x}_j \in T)\&(y_j = 1)|}{|\mathbf{x}_j \in T|} \tag{6.12}$$

Program 4 The holdout algorithm used to identify classifier and testing tags and to split the data into training and testing sets

```
input $\eta$, $\rho$, $\mathcal{C}$
FOR (tagtype=[AD, DN]){
    FOREACH($\tau_i$){
       Label $\omega_k = 1$ if $\tau_k = \tau_i$ and $\omega_k = -1$ if $\tau_k \neq \tau_i$
       Extract EPCs $s$ from all tags of the same type
       Preprocess EPCs $s$ so that $s \rightarrow \hat{s} \in \{\alpha, \theta, f_i\}$
       Select features with optimal interclass separation
       FOREACH($\tau_j$){
          FIND $N_i$=number of EPCs from tag $\tau_i$
          FIND $N_j$=number of EPCs from tag $\tau_j$
          IF ($\tau_i = \tau_j$){
             $\hat{\eta} = 0.5$
          ELSE
             $\hat{\eta} = \eta$
          }
          Reserve $\hat{\eta}N_j$-many $\mathbf{x}_j$ for testing set $T$
          Add $(1-\hat{\eta})N_j$-many $\mathbf{x}_j$ to the training set $R$
          Add $\mathbf{x}_i$ to $R$
          $\hat{\rho} = |\omega_k = -1|/|\omega_k = 1|$, $\mathbf{x}_k \in R$
          WHILE ($\hat{\rho} < \rho$){
             add $\mathbf{x}_k$ from tag $\tau_k \neq \tau_{i,j}$ to $R$
          }
          TRAIN classifier $\mathcal{C}$ on $R$
          TEST classifier $\mathcal{C}$ on $T$, yielding labels $y = \pm 1$
          SAVE results in confusion matrix, $\mathcal{L}(i,j) = |y = 1|/|y|$
       }
    }
}
```

In other words, the confusion matrix represents the number of EPCs from the testing tag that get classified as the classifier tag, and $0 \leq \mathcal{L} \leq 1$. An example of a confusion matrix resulting from holdout classification is in Figure 6.8. Here the value of $\mathcal{L}$ has been matched to a greyscale color intensity, so 0 → black and 1 → white. In reading the chart, we can see that $\mathcal{L}$ approaches 1 whenever $\tau_L = \tau_S$, and the lowest values of $\mathcal{L}$ occur for $\tau_L \neq \tau_S$. Hence the testing tags are more likely to be identified as the classifier tag whenever the testing tag is the same as the classifier tag. But there are some outliers, where as many as 66.7% of the EPCs from the tag AD15 were identified as the same as AD21.

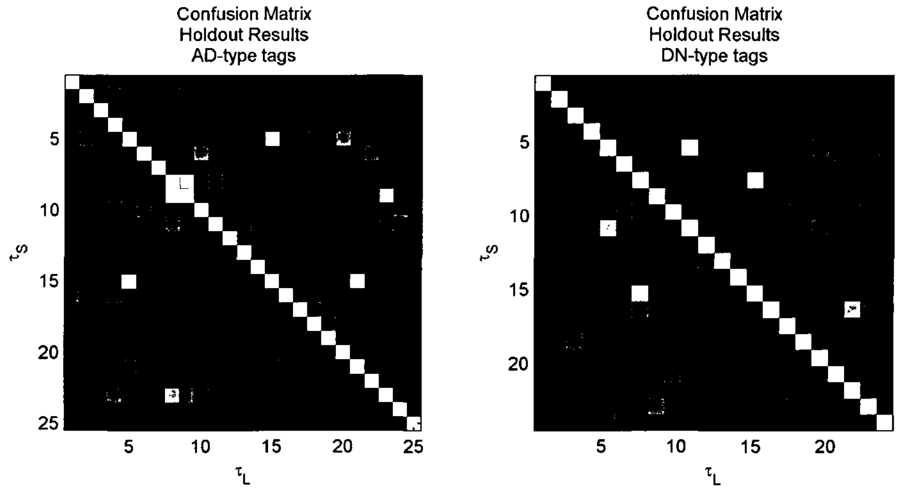


Figure 6.8: An example of a confusion matrix is shown as a grayscale image for both the AD and the DN-type tags. The color intensity relates to the percent of EPCs from the testing tag τ_S that were identified as coming from the classifier tag, τ_L.

Meanwhile, for the bootstrap method, the results of each random drawing of (R,T) from the data set is evaluated by finding the percent of correct labels, and $\mathcal{L}(i,j)$ represents the average percent correct.

6.8.1 Sensitivity and Specificity from Confusion Matrix

The confusion matrix described by Eqn 6.12 and Figure 6.8 give the percent of EPCs from the testing tag that were classified as originating from the classifier tag. In order to evaluate the performance of the classifier, we will apply a threshold h so that the false positive (f_+), false negative (f_-), true positive (t_+), and true negative (t_-) rates are given by Eqn 6.13

$$\begin{aligned} f_+ &= |\mathcal{L}(i,j) > h|, i \neq j \\ t_+ &= |\mathcal{L}(i,j) > h|, i = j \\ f_- &= |\mathcal{L}(i,j) \leq h|, i = j \\ t_- &= |\mathcal{L}(i,j) \leq h|, i \neq j \end{aligned} \tag{6.13}$$

We can then calculate the sensitivity (χ) and specificity (ψ) as in Eqn 6.14

$$\begin{aligned} \chi &= \tfrac{t_+}{t_+ + f_-} \\ \psi &= \tfrac{t_-}{t_- + f_+} \end{aligned} \tag{6.14}$$

A high rate of sensitivity means that all testing tags were correctly identified as not the same individual as the classifier tags whenever the classifier tag and the testing tag were not the same. Meanwhile, a high rate of specificity means that the testing tag was correctly identified as the classifier tag whenever the testing tag and the classifier tag were the same. Therefore, sensitivity and specificity are useful measures of successful pattern classification. However, sensitivity and specificity are actually functions of the threshold, $\chi(h)$ and $\psi(h)$. In order to summarize the predictive value of the RFID classifier as the threshold for the confusion matrix changes, we will need to use the receiver operating characteristic.

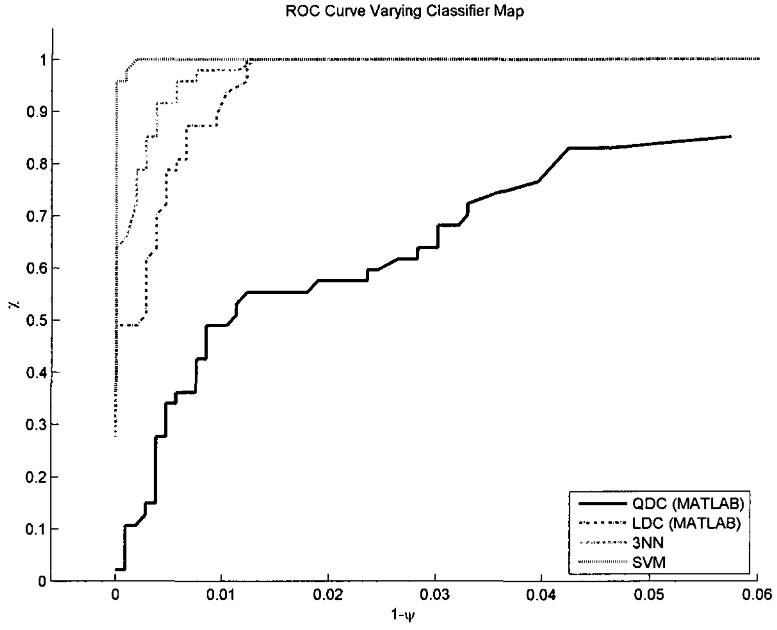


Figure 6.9: ROC curves for different classifiers are plotted here where all other variables but the classifier map were held constant and are the same as in Fig 6.8.

6.8.2 Receiver Operating Characteristic (ROC)

The receiver operating characteristic, otherwise known as ROC curve, is a useful summary of the sensitivity and specificity of a binary classifier as the threshold for discrimination changes [22]. The ROC curve is a plot of χ versus $1 - \psi$. In the case where $\chi = 1$ and $\psi = 1$, the ROC curve consists of a single point at $(0, 1)$ on the graph. A random guess would provide a point on the diagonal line of no-discrimination on the ROC curve that follows the equation $y = x$. A binary classifier that performs better than chance will have a curve in the upper left hand corner of the graph above the line of no-discrimination. Different classifiers can be compared in this way.

Figure 6.9 shows ROC curves from different RFID classifier configurations where all other variables are held constant except for the classifier map. Those other variables include features combined from EPC compression of both α and f_i, with 15

DWFP features selected from each. This classifier had design variables of $\eta = 0.5$ and $\rho = 10$, and the ROC curve was calculated by varying the threshold on $\mathcal{L}$ from $0.01 < h < 1$ as h varied in step sizes of 0.01. By looking at these curves, we can rank the classifier results so that SVM performed best with these variables, since the ROC curve of the SVM classifier lies closest to the upper left hand corner of the plot. The QDC classifier performed worst, since its ROC curve remained closest to the line of no-discrimination. However, there are many variables involved in the configuration of each classifier, and it would be difficult and time-consuming to choose the best by plotting all the ROC curves. A quantitative measure would be more useful.

One of the most common comprehensive measurement of the ROC curve is the measurement of the area under the ROC (AUC). Trapezoidal integration is usually sufficient to calculate AUC, but other methods are available [23]. Classifiers with an ROC curve leading to a larger AUC usually perform better than those with a lower AUC, but it is important to keep in mind that a classifier with a lower AUC may perform better in certain regions of the space than a classifier with a higher AUC.

Table 6.1: The AUC and normalized |AUC| are calculated for the classifiers shown in Fig 6.9. Note that SVM is the best classifier by the |AUC| measurement, which agrees with the ROC curves.

Classifier	AUC	|AUC|
QDC	0.036	0.628
LDC	0.107	0.973
3NN	0.089	0.986
SVM	0.084	0.999

As an example, consider the ROC curves shown in Fig 6.9. Table 6.1 shows the AUC measurement for each classifier as well as the AUC normalized by the range of $1 - \psi$ values, $|\text{AUC}| = \text{AUC}/(\max(1 - \psi) - \min(1 - \psi)$. As Fig 6.9 shows, SVM was the best-performing classifier of the four shown, yet LDC has a higher AUC value. It is the normalized AUC values, |AUC|, that more accurately rank the classifier performance so that QDC performs worst while SVM performs best. The disagreement between classifier ranking using AUC versus $|\text{AUC}|$ is most likely due to the fact that LDC does not have low enough ψ values for the values of the ordinate axis, $1-\psi$, to reach the origin. Therefore normalizing AUC allows for the ability to compare poor-performing classifiers like LDC whose range of $1-\psi$ is not as large with better-performing classifiers like SVM.

The measurement of AUC leads to a good general measure of predictiveness for binary classifiers but cannot be used to strictly rank classifier superiority. We will use AUC to narrow the results of the available classifiers to choose a few of the best ones. But the summary statistics f_+, f_-, t_+, t_- are still useful measures of the classifier performance for a given threshold h. Therefore, another metric we will use is $\min(f_+ + f_-)$ over the decision threshold, and we will present the minimum f_+ and f_- statistics as a percentage of the total number of combinations of the binary classifier for both tag types, which is 1109. The percentage of false positives ($f_+[\%]$) and false negatives ($f_-[\%]$) at that decision threshold are divided by the possible number of false positives and false negatives (1062 and 47) respectively. The decision threshold at which the minimum rate occurred will also be given. The evaluation procedure will conclude from these statistics the best decision threshold h for a given classifier configuration.

There were also different data sets used in the classification, including tags sampled at different sampling frequencies (f_1, f_2). Therefore, results will be shown for

1. Holdout classification results for tags sampled at f_1.
2. Bootstrap classification results for tags sampled at f_2.
3. Holdout classification results for tags sampled at f_2.
4. Holdout classification results for tags sampled at f_2 using features selected from

tags sampled at f_1.

5. Holdout classification results for tags sampled at f_2 removing the condition that $\tau_i = \tau_j$.

Lastly, the classifier results selected for presentation also have different restrictions placed on the classifier variables, which will be explained in the results for the tags sampled at f_1.

6.9 Results: Holdout Method

Since there are a lot of variables in configuring a single classifier, the dimensionality of the results space is large. Because the application is for security, we will attempt to find the fewest number of false positives that also have the best accuracy. But there are 3188 classifiers that meet the minimum number of false positives, which is zero. Therefore, we will present results narrowed by a few additional criteria. These results will be presented as a percentage of the total number of binary combination of classifier and testing tags.

First, let us consider the classifiers with the highest $\lvert\text{AUC}\rvert$. The results in Table 6.2 show the configuration of the classifier that yielded the highest $\lvert\text{AUC}\rvert$ in a holdout test. The table also shows that there were no errors made, since $\min(f_+ + f_-) = 0$.

Table 6.2: The classifier configurations with the highest $\lvert\text{AUC}\rvert$ are shown here. Also shown is that there were no classification errors made, or $\min(f_+ + f_-)$, at the decision threshold h corresponding to that minimum.

Classifier Configuration					Results [%]				$\lvert\text{AUC}\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	1	LDC (MATLAB)	0.1	1	0.00	0.00	0.00	50.1	0.9999
α	1	LDC (PRTools)	0.1	1	0.00	0.00	0.00	50.1	0.9999

These results show that for the configuration listed, no errors were detected upon a single split of the training and testing data. However, there were more than $3,000$ other configurations that also yielded no errors in a holdout test. The results in Table 6.2 were selected because their |AUC| measurement was higher, even if the difference in the |AUC| measurement was less than 0.02%. In fact, of all the 3188 holdout tests with $\min(f_+ + f_-) = 0$, the lowest value of |AUC| was less than 0.5% different than the value displayed in Table 6.2.

Since there are other classifiers with a similar |AUC| and no errors, we will also examine results drawn from other classifiers that may have more false positives. In order to choose among those to display, we will pick reasonable values of η and ρ. First, let us consider choosing a value of η, which is defined in Eqn 6.10 and represents the ratio of EPCs from the testing tag withheld for testing. If $\eta < 0.5$, as in the Table 6.2, then more EPCs from the testing tag were collected for the training set, and the training set was therefore more informed about the testing tag, leading to a more successful classification on the few EPCs left. However, if $\eta > 0.5$, then fewer EPCs are present in the training set than in the testing set, so classification is likely to be more unstable. Therefore, $\eta = 0.5$ balances the requirements of the training set and testing set best, and it is also the most common split value used in holdout tests. To pick a reasonable value of ρ, as defined in Eqn 6.11, we will first note that the holdout algorithm as described in Program 4 first allocates space in the training set to EPCs from the classifier tag and testing tag. It then calculates the proportion of class imbalance, $|\omega = -1|/|\omega = 1|$, and adds EPCs from other tags until ρ is achieved. However, when ρ is small, such as $\rho = 1$, it may be that the EPCs from the classifier tag and testing alone are enough to achieve or surpass the desired ρ value. That means the classifier is really only being trained on EPCs from the classifier tag and testing tag, which increases the likelihood of success, but does not perturb the training set enough to allow for a real world situation. Therefore, higher values of ρ, such as $\rho = 5, 10$ were examined.

Table 6.3: The best classifier results sorted in descending order by |AUC| are displayed here. The results include the minimum number wrong over all values of the decision threshold as a function of the total number of binary combinations ($\min(f_+ + f_-)[\%]$), as well as the values of f_+, f_-, and h at which the minimum occurs. These classifiers were selected with specific values of η and ρ.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	10	SVM	0.5	10	0.00	0.00	0.00	40.8	0.9990
α, f_i, θ	1	SVM	0.5	10	0.00	0.00	0.00	46.3	0.9990
α, f_i	1	SVM	0.5	10	0.09	0.09	0.00	48.3	0.9989
α, f_i, θ	5	SVM	0.5	10	0.00	0.00	0.00	42.6	0.9989
α, f_i, θ	20	SVM	0.5	10	0.09	0.09	0.00	39.7	0.9989
α, f_i	5	SVM	0.5	10	0.09	0.09	0.00	47.7	0.9989
α, f_i	10	SVM	0.5	10	0.09	0.09	0.00	57.2	0.9989
α, f_i, θ	15	SVM	0.5	10	0.09	0.09	0.00	34.0	0.9988
α, θ	15	SVM	0.5	10	0.09	0.09	0.00	48.2	0.9988
α, f_i	50	SVM	0.5	10	0.09	0.00	2.13	61.2	0.9987
α, f_i, θ	75	SVM	0.5	10	0.00	0.00	0.00	52.4	0.9987
α, f_i, θ	100	SVM	0.5	10	0.09	0.00	2.13	52.4	0.9986
α, f_i, θ	50	SVM	0.5	10	0.09	0.00	2.13	57.2	0.9986

The results in Table 6.3 and 6.4 show the classifier configurations that led to the highest |AUC|. Most of those have no misclassified tags, except a few which have 1 misclassified tag. Note that the best classifier used in all cases was SVM, and the EPC compression always combined α with either f_i or θ or both. The number of features chosen had less of an effect. The minimum threshold for h in most cases was around 50%, which makes sense in designing the classifier, since $h \simeq 50$ means that more than half of the EPCs from the testing tag have to be identified as the classifier tag in order to decide that the testing tag is the same as the classifier tag.

It is also helpful to examine the extreme values of the variables η and ρ. In the case of $\eta = 1$ (Table 6.5), all of the EPCs from the testing tag are withheld for testing, which is the most rigorous test of the classifier since the classifier is not trained on the tag being tested. As the results show, $< 1\%$ of misclassifications have occurred for

Table 6.4: Similar to Table 6.3, but these classifiers were selected with values of $\eta = 0.5$, $\rho = 5$.

Classifier Configuration					Results [%]				$\lvert$AUC$\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i	5	SVM	0.5	5	0.00	0.00	0.00	42.6	0.9991
α, f_i, θ	1	SVM	0.5	5	0.00	0.00	0.00	42.6	0.9991
α, f_i, θ	5	SVM	0.5	5	0.00	0.00	0.00	38.9	0.9991
α, f_i	1	SVM	0.5	5	0.00	0.00	0.00	50.1	0.9991
α, f_i	10	SVM	0.5	5	0.00	0.00	0.00	46.3	0.9990
α, θ	15	SVM	0.5	5	0.00	0.00	0.00	51.9	0.9990
α, f_i	15	SVM	0.5	5	0.00	0.00	0.00	42.6	0.9990
α, f_i	20	SVM	0.5	5	0.00	0.00	0.00	43.4	0.9990
α, θ	5	SVM	0.5	5	0.00	0.00	0.00	64.9	0.9990
α, θ	20	SVM	0.5	5	0.00	0.00	0.00	50.1	0.9990
α, f_i, θ	10	SVM	0.5	5	0.00	0.00	0.00	43.4	0.9990
α, f_i, θ	15	SVM	0.5	5	0.00	0.00	0.00	41.6	0.9990
α, θ	75	SVM	0.5	5	0.00	0.00	0.00	57.2	0.9989
α, θ	50	SVM	0.5	5	0.00	0.00	0.00	57.2	0.9989
α, f_i, θ	20	SVM	0.5	5	0.00	0.00	0.00	45.3	0.9989
α, f_i, θ	50	SVM	0.5	5	0.00	0.00	0.00	44.5	0.9988
α, f_i, θ	75	SVM	0.5	5	0.00	0.00	0.00	48.2	0.9988
α, f_i	100	SVM	0.5	5	0.00	0.00	0.00	44.5	0.9987
α, f_i	75	SVM	0.5	5	0.00	0.00	0.00	48.2	0.9987
α, f_i	50	SVM	0.5	5	0.00	0.00	0.00	44.5	0.9975

the classifiers displayed in the table. This means 9 or 10 tags have been misclassified. The other table of results shows the case where ρ is not used, so that all the EPCs from all the tags are included in the training set, except those withheld for testing. There are no errors in this example, but the |AUC| measurement is lower than the results in Table 6.2.

Table 6.5: The holdout classification results where all the EPCs from the testing tag were withheld from the training set($\eta = 1$) are displayed here.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	1	QDC (MATLAB)	1	15	0.90	0.75	4.26	90.1	0.9889
α	1	QDC-LDC	1	15	0.90	0.75	4.26	90.1	0.9889
α	1	QDC (MATLAB)	1	all	0.81	0.75	2.13	90.1	0.9887
α	1	QDC-LDC	1	all	0.81	0.75	2.13	90.1	0.9887
α	1	QDC (MATLAB)	1	18	0.90	0.75	4.26	90.1	0.9885
α	1	QDC-LDC	1	18	0.90	0.75	4.26	90.1	0.9885
α	1	QDC (MATLAB)	1	19	0.81	0.75	2.13	90.1	0.9875
α	1	QDC-LDC	1	19	0.81	0.75	2.13	90.1	0.9875
α	1	QDC (MATLAB)	1	10	0.90	0.75	4.26	89.9	0.9864
α	1	QDC-LDC	1	10	0.90	0.75	4.26	89.9	0.9864
α	1	QDC (MATLAB)	1	11	0.90	0.85	2.13	89.6	0.9861
α	1	QDC-LDC	1	11	0.90	0.85	2.13	89.6	0.9861

Table 6.6: The holdout classification results where all the EPCs are included in the training data set are displayed below.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	15	SVM	0.4	all	0.00	0.00	0.00	34.9	0.9990
α, f_i	5	SVM	0.2	all	0.00	0.00	0.00	23.9	0.9989
α, f_i, θ	20	SVM	0.2	all	0.00	0.00	0.00	37.6	0.9989
α, f_i, θ	20	SVM	0.5	all	0.00	0.00	0.00	38.1	0.9989
α, f_i, θ	10	SVM	0.3	all	0.00	0.00	0.00	37.6	0.9988
α, f_i, θ	50	SVM	0.4	all	0.00	0.00	0.00	43.8	0.9988
α, f_i, θ	75	SVM	0.4	all	0.00	0.00	0.00	38.7	0.9988
α, f_i, θ	50	SVM	0.2	all	0.00	0.00	0.00	37.6	0.9988

Since the results in Tables 6.3 and 6.4 were performed with the holdout tests, it is interesting to see how repeating those results changes the errors. Each of the classifier tests was re-tested using the holdout test and the errors were calculated using the original decision threshold h in the table. The classifiers with the least number of errors at that decision threshold over 10 repetitions are shown in Tables 6.7-6.10. Of all of these, the results shown in Table 6.9 are the highest, representing an average number of 12 or 13 misclassified tags, but note that the values of $\min(f_+ + f_-)$ and the f_+, f_-, and h measurements at which that minimum occur are displayed in terms of percentages over all 1109 binary combinations of classifier and testing tags, as before.

Table 6.7: The classifier from Table 6.3 with the least number of errors under repetition is presented here. The classifier encounters at most one misclassified tag.

Classifier Configuration					Results [%]				$\lvert$AUC$\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i	1	SVM	0.5	10	0.09	0.00	2.13	48.3	0.9989
					0.00	0.00	0.00	48.3	0.9991
					0.00	0.00	0.00	48.3	0.9990
					0.00	0.00	0.00	48.3	0.9989
					0.00	0.00	0.00	48.3	0.9992
					0.00	0.00	0.00	48.3	0.9990
					0.00	0.00	0.00	48.3	0.9991
					0.00	0.00	0.00	48.3	0.9990
					0.09	0.09	0.00	48.3	0.9989
					0.00	0.00	0.00	48.3	0.9989
				μ	0.02	0.01	0.21	-	0.9990
				σ	0.04	0.03	0.67	-	0.0001

6.10 Results: Bootstrap

The bootstrap method is similar to the holdout method, but the training and testing split is repeated, randomly and with repetition, and the average of the accuracy of each split is returned as the result. The confusion matrix appears similar to Figure 6.8, but in this case, color intensity relates to the average percent correct of each training/testing split. Instead of presenting results in terms of $\lvert$AUC$\rvert$ as before, we

Table 6.8: Similarly to Table 6.7, this table shows the classifier from Table 6.4 with the least number of errors under repetition. The classifier also encounters at most one misclassified tag.

Classifier Configuration					Results [%]				$\lvert$AUC$\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i	10	SVM	0.5	5	0.00	0.00	0.00	46.3	0.9990
					0.00	0.00	0.00	46.3	0.9990
					0.00	0.00	0.00	46.3	0.9989
					0.00	0.00	0.00	46.3	0.9992
					0.00	0.00	0.00	46.3	0.9989
					0.00	0.00	0.00	46.3	0.9990
					0.09	0.09	0.00	46.3	0.9990
					0.00	0.00	0.00	46.3	0.9990
					0.00	0.00	0.00	46.3	0.9991
					0.00	0.00	0.00	46.3	0.9990
				μ	0.01	0.01	0.00	-	0.9990
				σ	0.03	0.03	0.00	-	0.0001

calculate f_+ and f_- at a threshold of $h = 50\%$ to indicate the majority error. In addition, all of the results presented here are for the most rigorous case of $\eta = 1$.

Table 6.11 shows the classifiers with the fewest number of errors under the restrictions $h = 50\%$, $\eta = 1$. The results are presented as a percent of the 1109 binary combinations of classifier and testing tags, as in the previous section. Note that the error rate for the bootstrap tests is higher than the holdout tests, which is reasonable since the resampling method attempts more permutations of the training and testing sets. But still we are still observing more than 97% accuracy. Table 6.12 uses the maximum value of ρ, so that all the EPC collects not already in the testing set are inserted into the training set. Lastly, Table 6.13 shows the error under bootstrap resampling for the same classifiers presented in Table 6.5, which showed the best holdout results under the restriction $\eta = 1$. Again, the errors for the bootstrap resampling are higher than the holdout method, but the accuracy remains above 97%. We have not presented here our results for $\eta \neq 1$ for the bootstrap sampling method,

Table 6.9: The classifiers with the least number of errors under repetition from Table 6.5, in which all the EPCs from the testing tag were withheld from the training set, are displayed below.

Classifier Configuration					Results [%]				$\lvert$AUC$\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	1	QDC (MATLAB)	1	19	1.35	0.85	12.77	90.1	0.9849
					0.99	0.66	8.51	90.1	0.9858
					1.26	0.75	12.77	90.1	0.9867
					1.26	0.85	10.64	90.1	0.9839
					0.90	0.75	4.26	90.1	0.9901
					1.08	0.85	6.38	90.1	0.9872
					1.17	0.85	8.51	90.1	0.9859
					1.17	0.75	10.64	90.1	0.9869
					0.90	0.75	4.26	90.1	0.9872
					0.99	0.75	6.38	90.1	0.9851
				μ	1.11	0.78	8.51	-	0.9864
				σ	0.16	0.06	3.17	-	0.0017
α	1	QDC-LDC	1	19	1.35	0.85	12.77	90.1	0.9849
					0.99	0.66	8.51	90.1	0.9858
					1.26	0.75	12.77	90.1	0.9867
					1.26	0.85	10.64	90.1	0.9839
					0.90	0.75	4.26	90.1	0.9901
					1.08	0.85	6.38	90.1	0.9872
					1.17	0.85	8.51	90.1	0.9859
					1.17	0.75	10.64	90.1	0.9869
					0.90	0.75	4.26	90.1	0.9872
					0.99	0.75	6.38	90.1	0.9851
				μ	1.11	0.78	8.51	-	0.9864
				σ	0.16	0.06	3.17	-	0.0017

Table 6.10: The classifiers with the least number of errors under repetition from Table 6.6, in which all the EPCs not already included for testing were inserted into the training set, are displayed here.

Classifier Configuration					Results [%]				$\lvert$AUC$\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	20	SVM	0.2	all	0.27	0.19	2.13	37.6	0.9989
					0.09	0.00	2.13	37.6	0.9988
					0.09	0.09	0.00	37.6	0.9988
					0.09	0.00	2.13	37.6	0.9987
					0.00	0.00	0.00	37.6	0.9989
					0.09	0.09	0.00	37.6	0.9989
					0.09	0.09	0.00	37.6	0.9990
					0.09	0.00	2.13	37.6	0.9989
					0.18	0.19	0.00	37.6	0.9987
					0.09	0.00	2.13	37.6	0.9989
				μ	0.11	0.07	1.06	-	0.9988
				σ	0.07	0.08	1.12	-	0.0001
α, f_i, θ	10	SVM	0.3	all	0.18	0.09	2.13	37.6	0.9988
					0.27	0.19	2.13	37.6	0.9985
					0.09	0.00	2.13	37.6	0.9989
					0.09	0.00	2.13	37.6	0.9989
					0.00	0.00	0.00	37.6	0.9989
					0.18	0.09	2.13	37.6	0.9989
					0.00	0.00	0.00	37.6	0.9989
					0.09	0.09	0.00	37.6	0.9988
					0.09	0.00	2.13	37.6	0.9987
					0.09	0.00	2.13	37.6	0.9989
				μ	0.11	0.05	1.49	-	0.9988
				σ	0.08	0.07	1.03	-	0.0001

but the accuracy is even higher in that case, sometimes as high as 100%.

Table 6.11: The minimum number of errors using the bootstrap sampling method for $h = 50\%$ are presented here as a percent of the 1109 binary combinations of classifier and testing tags.

Classifier Configuration					Results [%]		
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-
α, θ	15	QDC (PRTools)	1	19	2.34	1.32	25.53
α, θ	15	QDC (PRTools)	1	21	2.34	1.32	25.53
α, f_i	10	QDC-LDC	1	17	2.43	2.07	10.64
α, f_i	10	QDC-LDC	1	19	2.43	1.98	12.77
α, f_i	10	QDC-LDC	1	21	2.43	1.98	12.77
α, θ	15	QDC (PRTools)	1	15	2.43	1.41	25.53
α, θ	15	QDC (PRTools)	1	16	2.43	1.32	27.66
α, θ	15	QDC (PRTools)	1	20	2.43	1.32	27.66

Table 6.12: The minimum number of errors using the bootstrap sampling method for $h = 50\%$ are limited by $\eta = 1$, ρ =all.

Classifier Configuration					Results [%]		
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-
α, θ	15	QDC (PRTools)	1	21	2.34	1.32	25.53
α, f_i	10	QDC-LDC	1	21	2.43	1.98	12.77
α	5	QDC (MATLAB)	1	21	2.61	2.73	0.00
α	5	QDC-LDC	1	21	2.61	2.73	0.00
α, f_i	5	QDC-LDC	1	21	2.61	2.07	14.89

6.11 Results: Lower Sampling Frequency

The above results were performed in a preliminary study from RFID tags of two different manufacture with about 25 individuals each. To test the classification method described in this chapter, additional data was collected, covering three different types of tags with about 50 individuals each. Some of the data included the same tags

Table 6.13: The minimum number of bootstrap errors from classifiers presented in Table 6.5 are presented here

Classifier Configuration					Results [%]		
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-
α	1	QDC (MATLAB)	1	15	3.07	3.20	0.00
α	1	QDC-LDC	1	15	3.07	3.20	0.00
α	1	QDC (MATLAB)	1	21	2.98	3.11	0.00
α	1	QDC-LDC	1	21	2.98	3.11	0.00
α	1	QDC (MATLAB)	1	18	2.98	3.11	0.00
α	1	QDC-LDC	1	18	2.98	3.11	0.00
α	1	QDC (MATLAB)	1	19	2.98	3.11	0.00
α	1	QDC-LDC	1	19	2.98	3.11	0.00
α	1	QDC (MATLAB)	1	10	2.98	3.11	0.00
α	1	QDC-LDC	1	10	2.98	3.11	0.00
α	1	QDC (MATLAB)	1	11	2.98	3.11	0.00
α	1	QDC-LDC	1	11	2.98	3.11	0.00

as those tested above, namely tags AD1-25 and DN1-25. A third tag type, Avery-Dennison Alien Omni-Squiggle Sticky Label (abbreviated as OM) was new to the study. The procedure and equipment remained the same, but the sampling frequency was lowered from $f_1 = 3.84$ MHz to $f_2 = 1.48$ MHz. The purpose of these results is twofold: first, to perform the classification routine again from start to finish with the new data to show that classifiers can be developed that accurately identify individual tags, and second, to extract the same features that were selected from the tags sampled at f_1 on the new tags that were sampled at f_2.

First, we examine classification results using the holdout sampling method that were obtained by selecting features on the new tags sampled at f_2. The accuracy results are presented as a percentage of the 7106 combinations, where the total number of false negatives possible was 6960 and the total number of false positives possible was 146. Table 6.14 shows the classifier configurations with the highest |AUC| from features selected from the tags that were sampled at f_2. Note that some had 100%

accuracy. Next, Tables 6.15 and 6.16 narrow down the classifier configurations that have very high accuracy by restricting the values of η and ρ to $\eta = 0.5$ and $\rho = 5, 10$ respectively. Table 6.17 shows the classifier configurations with the highest accuracy when the EPC collects from the testing tag were withheld from the training set ($\eta = 1$). Lastly, Table 6.18 shows the best classifier configurations under the constraint that all the EPC collects not already in the testing set are included for training.

Table 6.14: The classifier configurations with the highest |AUC| from tags sampled at f_2 are shown here.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	1	SVM	0.2	1	0.00	0.00	0.00	70.6	0.9999
α	1	SVM	0.3	1	0.00	0.00	0.00	70.9	0.9999
α	1	SVM	0.4	1	0.01	0.01	0.00	64.8	0.9999

Table 6.15: Classifier configurations from tags sampled at f_2 and that had the highest accuracy restrained by the conditions $\eta = 0.5$, $\rho = 10$ are shown in the table below.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	5	SVM	0.5	10	0.13	0.07	2.74	40.1	0.9991
α	1	1NN	0.5	10	0.20	0.06	6.85	56.1	0.9990
α	1	2NN	0.5	10	0.20	0.06	6.85	56.1	0.9990
α, f_i	5	SVM	0.5	10	0.18	0.06	6.16	46.4	0.9974
α, f_i	10	SVM	0.5	10	0.17	0.04	6.16	45.3	0.9919
α, θ	5	SVM	0.5	10	0.20	0.03	8.22	52.6	0.9916

The results in Tables 6.14-6.18 show the best classifier results under particular constraints of the classifier variables for RFID tags that were sampled at the lower sampling frequency f_2. The best results for those tables under repetition are shown in Tables 6.19-6.22. These results tend to show false positive rates of less than 1% but higher false negative rates as the restrictions on η, ρ become greater.

As previously mentioned, another test comparing tags sampled at f_1 and f_2 was performed. In this case, the features selected for classification were the same as those

Table 6.16: Classifier configurations from tags sampled at f_2 and that had the highest accuracy restrained by the conditions $\eta = 0.5$, $\rho = 5$ are shown in the table below.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	15	SVM	0.5	5	0.06	0.03	1.37	46.4	0.9997
α, f_i, θ	5	SVM	0.5	5	0.06	0.01	2.05	47.7	0.9997
α, f_i	15	SVM	0.5	5	0.07	0.01	2.74	52.4	0.9997
α, f_i	20	SVM	0.5	5	0.07	0.01	2.74	52.4	0.9997
α, θ	100	SVM	0.5	5	0.06	0.04	0.68	47.8	0.9997
α, f_i, θ	10	SVM	0.5	5	0.06	0.01	2.05	46.4	0.9996

Table 6.17: Classifier configurations from tags sampled at f_2 that had the highest accuracy restrained by the condition $\eta = 1$, so that all the EPC collects from the testing tag were withheld from the training set, are displayed below.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, θ	75	QDC (MATLAB)	1	2	0.65	0.24	19.86	51.9	0.9699
α, f_i, θ	50	QDC (MATLAB)	1	2	0.76	0.36	19.86	57.2	0.9443
α, f_i	75	QDC (MATLAB)	1	2	0.69	0.23	22.60	58.5	0.9391
α, θ	75	QDC (MATLAB)	1	3	0.83	0.30	26.03	16.1	0.9211
α, f_i, θ	50	QDC (MATLAB)	1	3	0.93	0.43	24.66	29.7	0.9167
α, f_i	75	QDC (MATLAB)	1	3	0.97	0.32	32.19	21.0	0.8803

Table 6.18: Classifier configurations from tags sampled at f_2 that had the highest accuracy restrained by the condition $\rho = all$, so that all EPC collects not from the testing set are included in the training set, are displayed below.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	1	1NN	0.3	all	0.76	0.26	24.66	30.8	0.9874
α	1	2NN	0.3	all	0.76	0.26	24.66	30.8	0.9874
α	1	1NN	0.2	all	0.75	0.27	23.29	31.3	0.9640
α	1	2NN	0.2	all	0.75	0.27	23.29	31.3	0.9640
α	1	1NN	0.1	all	0.72	0.09	30.82	37.6	0.9428
α	1	2NN	0.1	all	0.72	0.09	30.82	37.6	0.9428

Table 6.19: The classifier from Table 6.15 with the least number of errors under repetition using the holdout method on tags sampled at f_2 is shown below.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	5	SVM	0.5	10	0.14	0.07	3.42	40.1	0.9980
					0.15	0.10	2.74	40.1	0.9925
					0.14	0.07	3.42	40.1	0.9977
					0.10	0.04	2.74	40.1	0.9982
					0.14	0.07	3.42	40.1	0.9989
					0.14	0.07	3.42	40.1	0.9978
					0.15	0.10	2.74	40.1	0.9988
					0.17	0.09	4.11	40.1	0.9972
					0.13	0.06	3.42	40.1	0.9984
					0.08	0.06	1.37	40.1	0.9980
				μ	0.14	0.07	3.08	-	0.9975
				σ	0.03	0.02	0.74	-	0.0018

found best for the tags sampled at f_1 but they were extracted from the new tags that were sampled at f_2. We compare three different ways of selecting the features used for classification using variables that were determined to be the best classification configuration from the holdout results for tags sampled at f_1, shown in Tables 6.7-6.10. The three different feature selection schemes include:

1. Scheme 1: The results shown in Tables 6.7-6.10 in which the features were selected at f_1 and the classifier was tested at f_1.

2. Scheme 2: The same classifier configuration using features extracted at f_2 and tested at f_2.

3. Scheme 3: The same classifier configuration using features extracted at f_1 and tested at f_2.

Because the selection of training (R) and testing (S) sets is random in the holdout sampling method, the classification was repeated 10 times to assure the stability of the results. All 10 repetitions of the holdout method using the same settings are

Table 6.20: Similarly to Table 6.19, this table shows the classifier from Table 6.16 with the least number of errors under repetition using the holdout method on tags sampled at f_2.

Classifier Configuration					Results [%]				$\lvert\text{AUC}\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	15	SVM	0.5	5	0.08	0.03	2.74	46.4	0.9997
					0.06	0.01	2.05	46.4	0.9997
					0.08	0.06	1.37	46.4	0.9996
					0.07	0.04	1.37	46.4	0.9994
					0.11	0.07	2.05	46.4	0.9996
					0.07	0.03	2.05	46.4	0.9997
					0.06	0.03	1.37	46.4	0.9997
					0.07	0.06	0.68	46.4	0.9997
					0.06	0.03	1.37	46.4	0.9996
					0.07	0.03	2.05	46.4	0.9997
				μ	0.07	0.04	1.71	-	0.9996
				σ	0.02	0.02	0.58	-	0.0001
α, f_i, θ	5	SVM	0.5	5	0.10	0.07	1.37	47.7	0.9997
					0.06	0.01	2.05	47.7	0.9995
					0.06	0.03	1.37	47.7	0.9993
					0.08	0.04	2.05	47.7	0.9996
					0.10	0.06	2.05	47.7	0.9996
					0.04	0.01	1.37	47.7	0.9997
					0.07	0.03	2.05	47.7	0.9997
					0.06	0.03	1.37	47.7	0.9997
					0.11	0.07	2.05	47.7	0.9996
					0.06	0.01	2.05	47.7	0.9997
				μ	0.07	0.04	1.78	-	0.9996
				σ	0.02	0.02	0.35	-	0.0001

Table 6.21: The classifiers with the least number of errors under repetition from Table 6.17, in which all the EPCs from the testing tag were withheld from the training set, are displayed below. These tags were sampled at frequency f_2.

Classifier Configuration					Results [%]				$\lvert$AUC$\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, θ	75	QDC (MATLAB)	1	2	0.70	0.29	20.55	51.9	0.9520
					0.76	0.23	26.03	51.9	0.9636
					0.75	0.32	21.23	51.9	0.9350
					0.72	0.34	18.49	51.9	0.9782
					0.80	0.30	24.66	51.9	0.9572
					0.72	0.26	22.60	51.9	0.9428
					0.68	0.20	23.29	51.9	0.9674
					0.53	0.22	15.75	51.9	0.9440
					0.82	0.39	21.23	51.9	0.9532
					0.66	0.24	20.55	51.9	0.9509
				μ	0.71	0.28	21.44	-	0.9544
				σ	0.08	0.06	2.96	-	0.0128
α, f_i	75	QDC (MATLAB)	1	2	0.72	0.20	25.34	58.5	0.9365
					0.75	0.22	26.03	58.5	0.9272
					0.65	0.13	25.34	58.5	0.9448
					0.72	0.24	23.29	58.5	0.9241
					0.63	0.19	21.92	58.5	0.9509
					0.72	0.14	28.08	58.5	0.9521
					0.68	0.16	25.34	58.5	0.9559
					0.66	0.20	22.60	58.5	0.9490
					0.93	0.24	33.56	58.5	0.9273
					0.69	0.23	22.60	58.5	0.9375
				μ	0.71	0.20	25.41	-	0.9405
				σ	0.08	0.04	3.44	-	0.0116

Table 6.22: The classifiers with the least number of errors under repetition from Table 6.18, in which all the EPCs not already included for testing were inserted into the training set, are displayed here. These results were generated for tags sampled at frequency f_2.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	1	1NN	0.2	all	0.76	0.27	23.97	31.3	0.9699
					0.72	0.26	22.60	31.3	0.9770
					0.75	0.29	22.60	31.3	0.9760
					0.75	0.22	26.03	31.3	0.9644
					0.77	0.29	23.97	31.3	0.9703
					0.72	0.24	23.29	31.3	0.9835
					0.75	0.29	22.60	31.3	0.9764
					0.79	0.26	26.03	31.3	0.9763
					0.77	0.24	26.03	31.3	0.9838
					0.75	0.20	26.71	31.3	0.9764
				μ	0.75	0.26	24.38	-	0.9754
				σ	0.02	0.03	1.65	-	0.0060
α	1	2NN	0.2	all	0.76	0.27	23.97	31.3	0.9699
					0.72	0.26	22.60	31.3	0.9770
					0.75	0.29	22.60	31.3	0.9760
					0.75	0.22	26.03	31.3	0.9644
					0.77	0.29	23.97	31.3	0.9703
					0.72	0.24	23.29	31.3	0.9835
					0.75	0.29	22.60	31.3	0.9764
					0.79	0.26	26.03	31.3	0.9763
					0.77	0.24	26.03	31.3	0.9838
					0.75	0.20	26.71	31.3	0.9764
				μ	0.75	0.26	24.38	-	0.9754
				σ	0.02	0.03	1.65	-	0.0060

displayed in the tables. Because of the long computation time required to compute all permutations of the classifier variables, a full accounting of the best classifiers configured according to features selected by Schemes 2 and 3 has not been fully completed. Instead, we will compare classifiers that were shown to perform well using Scheme 1 in the previous report. Table 6.23 compares the results of the three schemes according to the classifier with the highest |AUC| from Table 6.2. Table 6.24 and 6.25 show results for the best-performing classifiers under repetition for particular values of η and ρ from Tables 6.7 and 6.8. Finally, Table 6.26 shows the best classifier under repetition from the previous report when all the EPC collects from the testing tag were withheld for testing ($\eta = 1$) from Table 6.9.

The comparison of results shown in Tables 6.23-6.26 show that Scheme 1 performs better than Scheme 2, so that the features selected at the higher-sampled tags work better than features selected at lower-sampled tags. However, Scheme 3 also performs better than Scheme 2 and comparably to Scheme 1, so that the features selected from the higher-sampled tags work well even when they were actually extracted from lower-sampled tags. Therefore, the features should always be selected from tags sampled at the highest possible sampling frequency, but implementing the classification on a large scale can be done by extracted features from tags sampled at a lower sampling frequency.

6.12 Results: Holdout Tests On All Tag Types

The last test performed was to use all the data sampled at the lower sampling frequency (f_2) for all three tag types AD, DN, and OM. The classifier routine shown in Program 4 remained the same except the condition on selecting EPCs from the same tag type for sampling was removed. In this case, a single individual tag's EPCs are

Table 6.23: The classifier with the highest |AUC| (Table 6.9) is compared here using the three feature selection schemes. The classifier variables included EPC compression method α, 1 DWFP feature, LDC (MATLAB) classifier, and $\eta = 0.1$, $\rho = 1$.

Feature Selection Scheme														
	1				2					3				
$\min(f_+ + f_-)$	f_+	f_-	h		$(f_+ + f_-)$	f_+	f_-	h		$(f_+ + f_-)$	f_+	f_-	h	
[%]	[%]	[%]	[%]	|AUC|	[%]	[%]	[%]	[%]	|AUC|	[%]	[%]	[%]	[%]	|AUC|
0.00	0.00	0.00	50.1	0.9997	0.23	0.23	0.00	50.1	0.9998	0.05	0.05	0.00	50.1	0.9997
0.09	0.09	0.00	50.1	0.9998	0.18	0.19	0.00	50.1	0.9999	0.00	0.00	0.00	50.1	0.9989
0.09	0.09	0.00	50.1	0.9997	0.20	0.20	0.00	50.1	0.9999	0.09	0.10	0.00	50.1	0.9998
0.00	0.00	0.00	50.1	0.9998	0.15	0.16	0.00	50.1	0.9997	0.14	0.14	0.00	50.1	0.9998
0.09	0.09	0.00	50.1	0.9998	0.14	0.14	0.00	50.1	0.9998	0.00	0.00	0.00	50.1	0.9998
0.09	0.09	0.00	50.1	0.9999	0.15	0.14	0.68	50.1	0.9998	0.00	0.00	0.00	50.1	0.9998
0.09	0.09	0.00	50.1	0.9999	0.17	0.17	0.00	50.1	0.9997	0.05	0.05	0.00	50.1	0.9997
0.00	0.00	0.00	50.1	0.9999	0.24	0.24	0.00	50.1	0.9999	0.05	0.05	0.00	50.1	0.9998
0.18	0.19	0.00	50.1	0.9999	0.23	0.23	0.00	50.1	0.9994	0.05	0.05	0.00	50.1	0.9998
0.00	0.00	0.00	50.1	0.9999	0.11	0.10	0.68	50.1	0.9998	0.00	0.00	0.00	50.1	0.9999

Table 6.24: The classifier that performed best under repetition using $\eta = 0.5$, $\rho = 10$ (Table 6.7) is compared here using the three feature selection schemes. The classifier variables included EPC compression method α and f_i, 1 DWFP feature, and SVM classifier

Feature Selection Scheme														
1					2					3				
$\min(f_+ + f_-)$	f_+	f_-	h		$(f_+ + f_-)$	f_+	f_-	h		$(f_+ + f_-)$	f_+	f_-	h	
[%]	[%]	[%]	[%]	|AUC|	[%]	[%]	[%]	[%]	|AUC|	[%]	[%]	[%]	[%]	|AUC|
0.09	0.00	2.13	48.3	0.9989	0.18	0.04	6.85	48.3	0.9849	0.05	0.00	2.22	48.3	0.9772
0.00	0.00	0.00	48.3	0.9991	0.21	0.04	8.22	48.3	0.9981	0.05	0.00	2.22	48.3	0.9766
0.00	0.00	0.00	48.3	0.9990	0.24	0.06	8.90	48.3	0.9923	0.05	0.00	2.22	48.3	0.9772
0.00	0.00	0.00	48.3	0.9989	0.18	0.04	6.85	48.3	0.9845	0.05	0.00	2.22	48.3	0.9772
0.00	0.00	0.00	48.3	0.9992	0.18	0.06	6.16	48.3	0.9852	0.05	0.00	2.22	48.3	0.9773
0.00	0.00	0.00	48.3	0.9990	0.14	0.04	4.79	48.3	0.9919	0.05	0.00	2.22	48.3	0.9773
0.00	0.00	0.00	48.3	0.9991	0.15	0.03	6.16	48.3	0.9920	0.05	0.00	2.22	48.3	0.9769
0.00	0.00	0.00	48.3	0.9990	0.18	0.04	6.85	48.3	0.9980	0.05	0.00	2.22	48.3	0.9772
0.09	0.09	0.00	48.3	0.9989	0.17	0.03	6.85	48.3	0.9919	0.05	0.00	2.22	48.3	0.9772
0.00	0.00	0.00	48.3	0.9989	0.23	0.07	7.53	48.3	0.9857	0.05	0.00	2.22	48.3	0.9773

Table 6.25: The classifier that performed best under repetition using $\eta = 0.5$, $\rho = 5$ (Table 6.8) is compared here using the three feature selection schemes. The classifier variables are otherwise the same as Table 6.24 above.

Feature Selection Scheme														
1					2					3				
$\min(f_+ + f_-)$ [%]	f_+ [%]	f_- [%]	h [%]	|AUC|	$(f_+ + f_-)$ [%]	f_+ [%]	f_- [%]	h [%]	|AUC|	$(f_+ + f_-)$ [%]	f_+ [%]	f_- [%]	h [%]	|AUC|
0.00	0.00	0.00	46.3	0.9990	0.14	0.09	2.74	46.3	0.9995	0.09	0.05	2.22	46.3	0.9993
0.00	0.00	0.00	46.3	0.9990	0.08	0.04	2.05	46.3	0.9975	0.05	0.00	2.22	46.3	0.9993
0.00	0.00	0.00	46.3	0.9989	0.11	0.06	2.74	46.3	0.9995	0.05	0.00	2.22	46.3	0.9992
0.00	0.00	0.00	46.3	0.9992	0.13	0.10	1.37	46.3	0.9994	0.05	0.00	2.22	46.3	0.9995
0.00	0.00	0.00	46.3	0.9989	0.08	0.04	2.05	46.3	0.9995	0.05	0.00	2.22	46.3	0.9987
0.00	0.00	0.00	46.3	0.9990	0.07	0.03	2.05	46.3	0.9996	0.05	0.00	2.22	46.3	0.9992
0.09	0.09	0.00	46.3	0.9990	0.11	0.07	2.05	46.3	0.9996	0.05	0.00	2.22	46.3	0.9990
0.00	0.00	0.00	46.3	0.9990	0.10	0.07	1.37	46.3	0.9996	0.05	0.00	2.22	46.3	0.9994
0.00	0.00	0.00	46.3	0.9991	0.08	0.03	2.74	46.3	0.9995	0.05	0.00	2.22	46.3	0.9993
0.00	0.00	0.00	46.3	0.9990	0.10	0.06	2.05	46.3	0.9995	0.05	0.00	2.22	46.3	0.9993

Table 6.26: The classifier that performed best under repetition using $\eta = 1$ (Table 6.9) is compared here using the three feature selection schemes. The classifier variables included EPC compression method α, 1 DWFP feature, QDC (MATLAB) classifier, and $\eta = 1$, $\rho = 19$.

Feature Selection Scheme														
1					2					3				
$\min(f_+ + f_-)$ [%]	f_+ [%]	f_- [%]	h [%]	|AUC|	$(f_+ + f_-)$ [%]	f_+ [%]	f_- [%]	h [%]	|AUC|	$(f_+ + f_-)$ [%]	f_+ [%]	f_- [%]	h [%]	|AUC|
1.35	0.85	12.77	90.1	0.9849	2.31	2.36	8.22	90.1	0.9665	1.35	1.38	15.56	90.1	0.9752
0.99	0.66	8.51	90.1	0.9858	2.38	2.43	4.79	90.1	0.9544	1.26	1.29	6.67	90.1	0.9742
1.26	0.75	12.77	90.1	0.9867	2.20	2.24	3.42	90.1	0.9603	1.35	1.38	8.89	90.1	0.9742
1.26	0.85	10.64	90.1	0.9839	2.28	2.33	5.48	90.1	0.9538	1.17	1.19	2.22	90.1	0.9789
0.90	0.75	4.26	90.1	0.9901	2.36	2.41	6.16	90.1	0.9631	1.17	1.19	6.67	90.1	0.9806
1.08	0.85	6.38	90.1	0.9872	2.36	2.41	6.85	90.1	0.9412	1.31	1.34	8.89	90.1	0.9741
1.17	0.85	8.51	90.1	0.9859	2.35	2.40	6.16	90.1	0.9646	1.07	1.10	6.67	90.1	0.9833
1.17	0.75	10.64	90.1	0.9869	2.28	2.33	6.16	90.1	0.9493	1.17	1.19	4.44	90.1	0.9750
0.90	0.75	4.26	90.1	0.9872	2.32	2.37	4.11	90.1	0.9649	1.35	1.38	4.44	90.1	0.9776
0.99	0.75	6.38	90.1	0.9851	2.17	2.21	2.05	90.1	0.9696	1.26	1.29	13.33	90.1	0.9775

compared against all other EPCs from all other individual tags of the three manufacture types. This last and most robust test is meant to test whether EPCs from tags of different types can be directly compared. These tests were performed at the lower sampling frequency (f_2) because that was the data set available for all of the tags, whereas only some of the AD and DN tags were sampled at the higher sampling frequency (f_1). In this scenario, the total number of binary combinations of individual tags is $21,316$. As before, the number of false positives and false negatives will be presented as a percentage of the possible number of false positives (21170) and false negatives (146).

Table 6.27 shows the classifier with the highest |AUC| for tags sampled at f_2 and tested in a holdout method regardless of the tag manufacture type. This is the highest |AUC| measured so far. But again, the η, ρ values are low, which suggests looking for classifier combinations at more reasonable values of η, ρ. Tables 6.28 and 6.29 show the best classifiers that were configured with $\eta = 0.5$ and $\rho = 5, 10$ respectively. The accuracy is still remarkably high, with no more than 6 total misclassified tags. More extreme restrictions on the classifier variables are shown in Table 6.30 with $\eta = 1$ and Table 6.31 with $\rho =$all. Though many more tags have been misclassified in these cases, the accuracy remains better than 99%.

As before, the holdout sampling was repeated for those classifiers already shown in Tables 6.28-6.31 in order to find a highly-accurate yet stable classifier for each restriction on the classifier variables. These results are shown in Tables 6.32-6.35. Again, the accuracy remains high in all cases. These results show that the restriction on tag type shown in Program 4 may be unnecessary, since comparing unlike tags has comparable accuracy to previous tests. It should be noted, however, that the features selected for classification here were selected from tags sampled at f_2 because that was the largest dataset available. As discussed in Section 6.11, features selected at higher sampling frequencies perform better than those selected at lower sampling frequencies. We anticipate even the excellent results shown in Tables 6.32-6.35 may be improved by increasing the sampling frequency before selecting features for classification.

Lastly, Table 6.36 summarizes the classifier results under repetition for three of the tests performed: holdout tests for tags sampled at f_1 and f_2, as well as the current tests on tags sampled at f_2 but where all types of tags were included in the holdout sampling method. The results show an overall high accuracy and also show that removing the condition previously used, in which only tags of the same type can be compared, actually results in an accuracy increase. This is likely due to the fact that tags of different manufacture are dissimilar enough from each other that the classifier has little difficulty distinguishing them.

Table 6.27: The classifier configurations with the highest |AUC| from tags regardless of manufacture sampled at f_2 are shown here.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	1	LDC (MATLAB)	0.2	1	0.00	0.00	0.00	86.7	1.0000

Table 6.28: The classifier configurations with $\eta = 0.5$, $\rho = 10$ from tags regardless of manufacture sampled at f_2 are shown here.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	5	SVM	0.5	10	0.02	0.02	0.68	50.1	0.9999
α, f_i, θ	15	SVM	0.5	10	0.02	0.01	1.37	55.1	0.9999
α, f_i, θ	10	SVM	0.5	10	0.02	0.01	1.37	52.3	0.9999
α, f_i, θ	20	SVM	0.5	10	0.02	0.02	0.68	52.3	0.9999
α, f_i	20	SVM	0.5	10	0.03	0.02	1.37	54.6	0.9998
α, θ	15	SVM	0.5	10	0.03	0.01	2.74	54.6	0.9998

Table 6.29: The classifier configurations with $\eta = 0.5$, $\rho = 5$ from tags regardless of manufacture sampled at f_2 are shown here.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	10	SVM	0.5	5	0.01	0.01	0.00	56.1	0.9999
α, f_i, θ	15	SVM	0.5	5	0.01	0.00	0.68	62.0	0.9999
α	10	1NN	0.5	5	0.01	0.01	0.68	65.1	0.9999
α, f_i, θ	20	SVM	0.5	5	0.01	0.00	1.37	59.6	0.9999
α, f_i, θ	1	SVM	0.5	5	0.01	0.01	0.00	66.7	0.9999
α, θ	5	SVM	0.5	5	0.01	0.01	0.00	62.0	0.9999
α, f_i, θ	5	SVM	0.5	5	0.01	0.01	0.00	61.0	0.9999

Table 6.30: The classifier configurations with $\eta = 1$ from tags regardless of manufacture sampled at f_2 are shown here.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	1	SVM	1	16	0.36	0.22	20.55	84.1	0.9935
α, f_i, θ	1	SVM	1	10	0.38	0.26	17.81	87.1	0.9932
α, f_i, θ	1	SVM	1	11	0.38	0.24	20.55	87.0	0.9931
α, f_i, θ	5	SVM	1	19	0.39	0.27	17.81	78.5	0.9924
α, θ	10	SVM	1	20	0.39	0.23	23.29	79.8	0.9917
α, f_i, θ	1	SVM	1	20	0.37	0.25	17.81	80.8	0.9868

Table 6.31: The classifier configurations with ρ =all from tags regardless of manufacture sampled at f_2 are shown here.

Classifier Configuration					Results [%]				|AUC|
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	10	1NN	0.3	all	0.28	0.09	26.71	30.8	0.9948
α	10	1NN	0.5	all	0.32	0.04	40.41	40.1	0.9943
α	10	1NN	0.4	all	0.30	0.08	32.19	34.4	0.9941
α	10	1NN	0.6	all	0.33	0.11	32.88	36.2	0.9931
α	10	1NN	0.2	all	0.29	0.06	34.25	35.3	0.9882
α	10	1NN	0.1	all	0.29	0.04	36.99	37.6	0.9410

Table 6.32: The classifier from Table 6.28 with the least number of errors under repetition using the holdout method on tags sampled at f_2 for all tag types is shown below.

Classifier Configuration					Results [%]				$\lvert\text{AUC}\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	15	SVM	0.5	10	0.02	0.01	1.37	55.1	0.9999
					0.03	0.01	3.42	55.1	0.9999
					0.05	0.03	2.74	55.1	0.9998
					0.03	0.02	1.37	55.1	0.9998
					0.03	0.01	2.74	55.1	0.9998
					0.02	0.01	1.37	55.1	0.9999
					0.04	0.03	2.05	55.1	0.9998
					0.03	0.02	2.05	55.1	0.9998
					0.04	0.02	3.42	55.1	0.9999
					0.03	0.02	2.05	55.1	0.9999
				μ	0.03	0.02	2.26	-	0.9998
				σ	0.01	0.01	0.79	-	0.0000

Table 6.33: The classifier from Table 6.29 with the least number of errors under repetition using the holdout method on tags sampled at f_2 for all tag types is shown below.

Classifier Configuration					Results [%]				$\lvert\text{AUC}\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	1	SVM	0.5	5	0.01	0.01	0.00	66.7	0.9999
					0.02	0.01	1.37	66.7	0.9999
					0.01	0.01	0.68	66.7	0.9999
					0.00	0.00	0.00	66.7	0.9999
					0.02	0.01	0.68	66.7	0.9999
					0.01	0.01	0.00	66.7	0.9999
					0.02	0.02	0.00	66.7	0.9998
					0.01	0.01	0.68	66.7	0.9999
					0.02	0.02	0.00	66.7	0.9999
					0.01	0.01	0.68	66.7	0.9999
				μ	0.01	0.01	0.41	-	0.9999
				σ	0.00	0.00	0.48	-	0.0000

Table 6.34: The classifier from Table 6.30 with the least number of errors under repetition using the holdout method on tags sampled at f_2 for all tag types is shown below.

Classifier Configuration					Results [%]				$\lvert$AUC$\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α, f_i, θ	1	SVM	1	16	0.41	0.25	23.29	84.1	0.9929
					0.43	0.26	23.97	84.1	0.9930
					0.40	0.24	23.97	84.1	0.9925
					0.41	0.24	24.66	84.1	0.9923
					0.43	0.26	25.34	84.1	0.9867
					0.42	0.25	26.03	84.1	0.9863
					0.44	0.27	24.66	84.1	0.9864
					0.40	0.26	21.23	84.1	0.9929
					0.44	0.26	25.34	84.1	0.9866
					0.42	0.24	26.03	84.1	0.9865
				μ	0.42	0.25	24.45	-	0.9896
				σ	0.01	0.01	1.45	-	0.0033

Table 6.35: The classifier from Table 6.31 with the least number of errors under repetition using the holdout method on tags sampled at f_2 for all tag types is shown below.

Classifier Configuration					Results [%]				$\lvert$AUC$\rvert$
EPC Compression	#DWFP Features	Classifier	η	ρ	$\min(f_+ + f_-)$	f_+	f_-	h	
α	10	1NN	0.2	all	0.32	0.07	36.99	35.3	0.9813
					0.30	0.07	34.93	35.3	0.9882
					0.31	0.05	37.67	35.3	0.9815
					0.30	0.07	32.88	35.3	0.9886
					0.31	0.07	36.30	35.3	0.9882
					0.31	0.07	36.30	35.3	0.9882
					0.30	0.07	33.56	35.3	0.9819
					0.36	0.09	39.73	35.3	0.9876
					0.33	0.09	35.62	35.3	0.9748
					0.33	0.08	36.99	35.3	0.9884
				μ	0.32	0.07	36.10	-	0.9849
				σ	0.02	0.01	1.99	-	0.0047

Table 6.36: The best classifiers under repetition for the RFID tags sampled at f_1 and f_2 are compared against the present results for the tags sampled at f_2 for all tag types. The arithmetic mean and standard deviation of the holdout sampling results for 10 repetitions are shown as a percentage of the total number of binary combinations, total number of possible false positives, and total number of possible false negatives for each method under study.

Classifier Selection	Reference Table	Sampling Method	Sampling Frequency	Results $(\mu \pm \sigma)[\%]$			
				$\min(f_+ + f_-)$	f_+	f_-	|AUC|
|AUC|	6.2	Holdout	f_1	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.9999 ± 0.0000
	6.14	Holdout	f_2	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.00	0.9999 ± 0.0000
	6.27	All tag types	f_2	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.0000 ± 0.0000
$\eta = 0.5$, $\rho = 10$	6.7	Holdout	f_1	0.02 ± 0.04	0.01 ± 0.03	0.21 ± 0.67	0.9990 ± 0.0001
	6.19	Holdout	f_2	0.14 ± 0.03	0.07 ± 0.02	3.08 ± 0.74	0.9975 ± 0.0018
	6.32	All tag types	f_2	0.03 ± 0.01	0.02 ± 0.01	2.26 ± 0.79	0.9998 ± 0.0000
$\eta = 0.5$, $\rho = 5$	6.8	Holdout	f_1	0.01 ± 0.03	0.01 ± 0.03	0.00 ± 0.00	0.9990 ± 0.0001
	6.20	Holdout	f_2	0.07 ± 0.02	0.04 ± 0.02	1.71 ± 0.58	0.9996 ± 0.0001
	6.33	All tag types	f_2	0.01 ± 0.00	0.01 ± 0.00	0.41 ± 0.48	0.9999 ± 0.0000
$\eta = 1$	6.9	Holdout	f_1	1.11 ± 0.16	0.78 ± 0.06	8.51 ± 3.17	0.9864 ± 0.0017
	6.21	Holdout	f_2	0.71 ± 0.08	0.20 ± 0.04	25.41 ± 3.44	0.9405 ± 0.0116
	6.34	All tag types	f_2	0.42 ± 0.01	0.25 ± 0.01	24.45 ± 1.45	0.9896 ± 0.0033
$\rho =$all	6.10	Holdout	f_1	0.11 ± 0.07	0.07 ± 0.08	1.06 ± 1.12	0.9988 ± 0.0001
	6.22	Holdout	f_2	0.75 ± 0.02	0.26 ± 0.03	24.38 ± 1.65	0.9754 ± 0.0060
	6.35	All tag types	f_2	0.32 ± 0.02	0.07 ± 0.01	36.10 ± 1.99	0.9849 ± 0.0047

6.13 Discussion

We next discuss the results presented above in light of the variables involved in the classifier design. Both sampling methods, including the holdout and the bootstrap methods, present highly accurate results for the tags sampled at f_1, which gave accuracies above 99% and 97%, respectively. The holdout method may be more feasible to implement in practice because of its smaller computational time. In addition, the repeated holdout tests shown in Tables 6.7, 6.8, 6.9, 6.10 demonstrate that the extracted features are stable, since the results vary little under repetition. However, the bootstrap results were presented for completeness, as they are a more thorough test of the training/testing split.

The EPC compression method seems to depend on the limits placed on ρ and η. Lower values of η seem to require more EPC compression methods included in the features, while higher values of ρ or η tend to correlate with only one or two values of the EPC compression method. It is also interesting to note that when more than one EPC compression method is used in the feature vector, it is usually a combination of α and θ or α and f_i, less often a combination of all three at once. This is logically consistent with the fact that f_i and θ are both measures of the phase of the complex $s(t)$ and may be redundant.

Lastly, the differing values of η and ρ used in the classifier configuration do not seem to have a large difference on the classifier performance. The most rigorous values of these variables, $\eta = 1$ and $\rho =$all, still lead to a highly accurate classifier. The accuracy of the RFID classification is improved when $\eta \neq 1$, since then some EPC collects from the testing tag are present in the training data set, but accuracy values greater than 97% are still feasible without having EPC collects from the testing tag in the data set. Also, the fact that high accuracy is achievable with $\rho =$all implies that the class imbalance is not always diverse enough to require careful proportion of classes in the training data set.

The results shown comparing the three different ways of extracting features from the two data sets (one sampled at f_1, one sampled at f_2, and one extracted from tags sampled at f_2 using features selected at f_1) show that selecting features at the higher sampling frequency (f_1) work better on tags sampled at a lower frequency (f_2) than selecting features at that lower sampling frequency. The previous section, in which individual tags from any type of manufacture were compared against each other, show that the technique is robust between different types of RFID tags.

6.14 Conclusion

We have created a binary classifier that can identify whether or not RFID tag A is the same as RFID tag B with better than 98% accuracy, despite the fact that tag A and tag B present the same EPC information. The classifier has some limitations, namely, it cannot determine the identification of tag A, only whether or not it is the same as tag B, which is better in some applications (such as ID badges) than others. It is an improvement from current RFID security, which relies more on cryptography than pattern classification. It is also interesting to note that the classifier designed here relies only on unintended qualities of the EPCs themselves, not relying on demodulation of the RFID amplitude shift key modulation. Further research needs to be conducted with many more tags from different manufactures to assure that the method presented here is general. Lastly, since the actual RFID signal is being used for pattern matching, it is possible that a degradation of the RFID tag itself over time might decrease the performance of the pattern classifier since the unintentional modulations of the signal arising from the individual tag may change. The robustness of this technique will also require further testing.

6.15 Appendix: Notation

Table 6.37 contains a list of mathematical notation used in this chapter.

Table 6.37: A list of mathematical notation

N	=	number of measurements
M	=	number of features
i, j, k	=	indices
$\mathbf{x}$	=	feature vector
$\mathbf{x}_i$	=	feature vector of the ith measurement
$x_{i,j}$	=	the jth feature corresponding to the ith measurement
p	=	probability
ω	=	actual classes ($\omega \in \{1, -1\}$)
y	=	predicted classes
t	=	time
f	=	frequency
τ	=	individual tag number
$s_i(t)$	=	an RFID signal or extracted EPC from the RFID signal
$r(t)$	=	the real part of $s(t)$
$c(t)$	=	the imaginary part of $s(t)$
$\alpha(t)$	=	the amplitude or magnitude of $s(t)$
$\theta(t)$	=	the phase of $s(t)$
$f_i(t)$	=	the instantaneous frequency of $s(t)$
a, b	=	DWFP parameters ($a \propto f, b \propto t$)
$\psi_{a,b}$	=	mother wavelet
$C(a, b)$	=	wavelet coefficient
$I(a, b)$	=	binary DWFP fingerprint image
$D_{j,v}$	=	DWFP feature, with $\nu = 1 \ldots, 16$
μ	=	mean
σ	=	standard deviation
κ_j	=	kurtosis of $s_j(t)$
$g(\mathbf{x})$	=	discrimination function
$\hat{\mathbf{\Sigma}}_i$	=	sample covariance matrix
$\mathbf{S}_W$	=	common group covariance matrix
$\mathbf{m}_i$	=	mean of features
L	=	subset of classifier tags, $1 \leq L \leq N$
S	=	subset of testing tags
R	=	training set
T	=	testing set
η	=	proportion of S withheld for T
ρ	=	proportion of $\omega = -1$ to $\omega = 1$ in R
$\mathcal{L}$	=	confusion matrix
f_+	=	number of false positives
f_-	=	number of false negatives
t_+	=	number of true positives
t_-	=	number of true negatives
χ, ψ	=	sensitivity, specificity

Bibliography

[1] A. Juels, “Rfid security and privacy: A research survey,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–395, February 2006.

[2] S. Bono and M. Green, “Security analysis of a cryptographically-enabled rfid device,” in 14th *USENIX Security Symposium*, P. McDaniel, Ed., 2005, pp. 1–16.

[3] L. E. Langley, “Specific emitter identification (SEI) and classical paramter fusion technology,” in *Western Electronic Show and Convention (WESCON)*, El Segundo, California, 1993, pp. 377–381.

[4] C. Shieh and C. Lin, “A vector neural network for emitter identification,” *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 8, pp. 1120–1127, 2002.

[5] A. Kawalec and R. Owczarek, “Radar emitter identification using intrapulse data,” in 15th *International Conference on Microwaves, Radar and Wireless Communications*, vol. 2, 2004, pp. 435–438.

[6] J. Dudczyk, M. Wnuk, and J. Matuszewski, “Applying the radiated emission to the specific emitter identification,” *Journal of Telecommunications and Information Technology*, vol. 2, pp. 57–60, 2005.

[7] J. Matuszewski, “Specific emitter identification,” in *2008 International Radar Symposium*, 2008, pp. 1–4.

[8] K. I. Talbor, P. R. Duley, and M. H. Hyatt, "Specific emitter identification and verification," *Technology Review*, pp. 113–133, 2003.

[9] K. A. Remley, C. A. Grosvenor, R. T. Johnk, D. R. Novotny, P. D. Hale, and M. D. McKinley, "Electromagnetic signatures of WLAN cards and network security," in *2005 IEEE International Symposium on Signal Processing and Information Technology*, 2005, pp. 484–488.

[10] K. Kim, C. Spooner, I. Akbar, and J. Reed, "Specific Emitter Identification for cognitive radio with application to ieee 802.11," in *IEEE Globecom 2008*, 2008, pp. 1–5.

[11] A. R. Webb, *Statistical Pattern Recognition*, 2^{nd} ed. Hoboken, NJ: Wiley, 2002.

[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines.* New York: Cambridge University Press, 2000.

[13] EPCglobal Inc., *EPC Radio-Frequency Identity Protocols: Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz Version 1.2.0*, October 2008. [Online]. Available: http://www.epcglobalinc.org/standards/uhfc1g2

[14] K. J. Ellis and N. Serinken, "Characteristics of radio transmitter fingerprints," *Radio Science*, vol. 36, no. 4, pp. 585–597, 2001.

[15] J. Hou and M. K. Hinders, "Dynamic Wavelet Fingerprint identification of ultrasound signals," *Materials Evaluation*, vol. 60, no. 9, pp. 1089–1093, 2002.

[16] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision.* Boston, MA: Addison-Wesley, 1992, vol. I.

[17] B. K. P. Horn, *Robot Vision.* Cambridge, MA: MIT Press, 1986.

[18] R. E. Learned and A. S. Wilsky, "A wavelet packet approach to transient signal classification," *Applied and Computational Harmonic Analysis*, vol. 2, pp. 265–278, 1995.

[19] N. Japkowicz and S. Stephen, "The class imbalance problem: a systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[20] L. I. Kuncheva, *Combining pattern classifiers methods.* New York: Wiley, 2004.

[21] G. M. Weiss and F. Provost, "Effect of class distribution on classifier learning: An empirical study," Rutgers University Department of Computer Science, Tech. Rep. ML-TR-44, August 2001.

[22] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006.

[23] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145 – 1159, 1997.

Chapter 7

Conclusions & Future Work

7.1 Conclusions

This dissertation has described pattern classification techniques for signals with the goal of enhancing detection techniques using artificial intelligence. The motivation for performing this work was to solve detection problems using different kinds of fields in which the physics becomes very complicated because the acoustic or electromagnetic fields propagate throughout real world structures and materials. All of the applications shown here required pattern classification to perform detection; existing detection methods were insufficient.

The technique described in this work includes a novel feature extraction mechanism using DWFP images. These images were generated by performing a stationary wavelet transform and projecting the 3D coefficients onto the time-scale plane, and the feature extraction method involved measuring properties of the resulting 2D binary images using image recognition techniques. In many cases, features were selected from the fingerprint properties to use for classification by applying a distance metric, since the features most likely to distinguish between the assigned categories were the ones with the largest distance in the feature space. The DWFP has been used to extract features for all of the applications, even though the feature selection method differed.

In addition, the pattern classification methods developed for each application followed two main rules of the theory. As the Ugly Duckling theorem states, there was no *a priori* best feature representation for the applications, only the best based on the assumptions made about the distance metric. And as the No Free Lunch Theorem states, there was no particular classifier (like kNN) or learning theory (like k-means clustering) that worked best for all the applications as a whole. Instead, each application had a uniquely-selected feature selection scheme drawn from the experimental population as well as a best classifier configuration.

We have described four applications of pattern classification for signals: the development of an ultrasonographic periodontal probe, the identification of flaw type in Lamb wave tomographic scans of an aluminum pipe, prediction of roof falls in a limestone mine, and SEI for RFID tags. The accuracy was generally high for all of the applications, and the classification accuracy was best in those applications with a large amplitude displacement in the signal, such as the RFID classification, as opposed to those with relatively low energies, such as the periodontal probe. While the classification design differed somewhat for each application based on its function, the feature extraction method with DWFP and image recognition is general enough to be applied to any signal. As is often the case in pattern classification, sometimes the best features are those drawn from physical interpretation of the domain, such as the corner frequency and low-frequency spectral level derived from geophysics, in addition to the automatically-generated wavelet features.

7.2 Future Work

In general, the artificial intelligence algorithms developed in this dissertation could be improved in several ways. First, the DWFP process is time-consuming, and fingerprinting an entire signal for feature selection, as has been done here, requires significant computer time. In this work, we have merely used the fingerprint created for feature selection in order to extract the features later for classification. The speed of the process could be improved for real-time applications by only fingerprinting windowed portions of the signal where the features have been selected for extraction. This would require some careful windowing since the feature selection in the periodontal application showed in Fig. 3.13 that the fingerprint features distort at the edges of the windows.

Another improvement would be to add more formal feature selection processes to selection DWFP features to use for pattern classification. Feature selection schemes generally require testing all possible combinations of the feature space with a classifier to find the feature vector that yields the lowest error. Some, such as principal component analysis, require projecting the feature space onto a lower-dimensional subspace to explain the variance in the data. These formal feature selection schemes are computationally time-consuming and take hours to days, depending on the size of the classification matrix, which requires several orders of magnitude more computational time than the distance metric scheme used here. However, feature selection can strongly improve the classification results. One less time-consuming method is genetic algorithms, which use evolutionary methods to produce the optimal feature vector [1]. The features are treated like chromosomes and features are selected based on their fitness. The transformation from the feature domain to chromosomes as well as the heuristic used for chromosome selection can vary in a multitude of ways, but the algorithm is fairly standard. Genetic algorithms are therefore used for feature selection and are computationally less expensive because of the fitness selection process.

In what follows, we examine the further work that could be done on each individual application.

7.2.1 An Ultrasonographic Periodontal Probe

This chapter was able to associate manual with ultrasonographic pocket depth measurements with between 70-87% accuracy using pattern classification. However, the research performed on the ultrasonographic periodontal probe used clinical data on only 12 patients, and part of the problem of statistically associating manual with ultrasonographic pocket depths was that the clinical population was mostly healthy. Clearly, in order to claim that the probe has widespread utility, there must be a much higher clinical population measured using both probes, at least on the order of 500 patients. With more data, there would likely be different features selected, but the process of DWFP feature extraction and selection should be general enough to use the same methods. Additionally, the next clinical trials should contain more patients with periodontal disease, that is, with pocket depths ≥ 4 mm. This would reduce the need for the binary classification algorithm as long as the distribution of manually-measured pocket depths is less skewed.

Obtaining a higher quantity and wider variety of data is the most necessary next step, but there could also be technical improvements to the probe itself. For example, the transducer used in the current periodontal probe is 10 MHz, which was the highest frequency transducer that could perform reliably that could be manufactured at the time. It may be possible to obtain even higher frequency transducers now, at either 15 MHz or 20 MHz. The simulations and experiments have not demonstrated a lot of attenuation at the current 10 Mhz frequency, so penetrating depth should not be a problem at higher transducer frequencies. The goal of increasing the transducer frequency is to reduce the wavelength of the ultrasonic waves, which increases detection capability of the device. Another improvement could be to further optimize the tip using the 3D elastodynamic simulations described in the chapter. The artifacts from the tip are a main reason why windowing the signal for DWFP transformation is required and they can also overlap some of the lower pocket depth reflections. The simulations could also be used to test the effect of different anatomies and different tooth types. The classification results may be improved if they differentiated by tooth site, which could not be performed in this study because there would only be 12 waveforms for each tooth site. Further, the noise in the signal could be significantly reduced by improving the water control, which is the largest factor in the signal noise, or removing the water altogether by introducing a solid tapered delay line with ultrasonic gel couplant to the gum interface. We have already investigated solid tapered delay lines as shown in Fig. 2.16, and a similar device could be used for soft tissue as long as the material properties of the delay line more closely matched soft tissue material properties as opposed to the acrylic used in the hard tissue example.

7.2.2 Classification of Pipe Ray Paths for Tomography

The flaw type identification of Lamb wave tomographic ray paths in an aluminum pipe was successful in identifying individual ray paths with up to 80% accuracy. Even more useful was the qualitative flaw description, which was often accurate when the same flaw size was used for training and testing. The application could be expanded to different types of flaws as well as different geometries, since the pipe geometry is analogous to plates, and the method does not rely upon the pipe geometry at all beyond categorizing the actual class of each ray path. In addition, the same method could be applied to indicate flaw severity as well as flaw type. All that is required is to change the nature of the category labels and the classification process remains similar. An indication of flaw severity could assist repair decisions for structural health monitoring.

Another way to improve the classification method described in this application is to change the method of assigning known category labels, in this case, the flaw type. Since the flaws in the pipe were induced in the laboratory and therefore the location and type was known exactly, we were able to anticipate with a high degree of accuracy which ray paths would intersect these flaws in order to generate known category labels. Further work in this area could involve generating a training data set with a large number of flaw types and severities, and then real world applications would generate testing data sets that could be used to describe flaws. However, when describing flaw severity instead of identifying flaw type, another method of assigning category labels is to use information from the tomographic reconstructions themselves. The tomographic reconstruction images, such as Fig. 4.1, calculate the anticipated mode arrival roughly by thresholding the ray path. These empirically-determined times shift when a flaw is present, and they shift more when more severe flaws occur. Therefore, the empirical mode arrival times can be used to indicate flaw severity. Therefore, flaw characterization using the empirical mode arrival times is a natural extension of the work present in this dissertation.

7.2.3 Roof Fall Predictors

The prediction of roof falls using fuzzy classification of microseismic data anticipated the event by between a few hours to a few days over the current method of visual identification. While an obvious improvement over previous methods, there are some limitations. For example, the event cannot be exactly localized in the mine, only a general location indicated by the known locations of the geophones with high RFI values. There are two possible was of overcoming this. One would involve placing more geophones in a mine in order to provide a finer grid on which the roof fall prediction could take place. Another option is to use triaxial geophones because such instruments are capable of capturing three components of seismic energy at once and can therefore be useful in calculating hypocentral distance, which is the geophysical measurement of seismic source location. Lastly, the research described in this chapter involved detecting roof falls in limestone mines, but another application may be in coal mines, since coal provides a useful source of domestic energy and roof falls in coal mines therefore have great interest to the mining community.

The research in Chapter 5 could also be improved by testing the fuzzy classification technique on more data. While there were 13,228 events in this study, only 2 roof falls occurred. NIOSH kindly provided the 6-month period of microseismic data, but there is more data available in which more geophones were added to the same mine. Another possible improvement is to take more data in a new mine with a technician on site to label every recorded event with an actual category label so that supervised pattern classification could also be tested. However, this last suggestion is likely to be above the budget of most mining safety operations.

7.2.4 SEI for RFID

The results in Chapter 6 demonstrate that the DWFP and statistical features extracted from the RFID tags' EPC codes were successful in identifying unique individual RFID tags with up to 98% accuracy. The technique should always be tested on more tags of different types before being implemented, but the results in this dissertation are a start. Other possible improvements include testing different classifiers. Formal feature selection could be added, though the distance metric was used here to reduce the long computational time otherwise required from exhaustive searches for optimal features. However, the genetic algorithm may yield improvements over the distance metric for feature selection and is often less computationally expensive. One particular avenue of investigation to distinguish individual RFID tags would be to examine the difference between the expected and actual EPC codes, which may yield additional features to use for classification.

One major problem with the RFID classification is the computational time required to identify the tag. In most applications, such as identification card verification, computational times of under a few seconds would be optimal, yet the procedure here takes minutes. To speed the procedure, the signal could be windowed around the feature used for extraction instead of fingerprinting the entire waveform. Improved feature selection using the genetic algorithm could also speed up the classification algorithm. Faster classifiers, such as the discriminant classifiers, should be the focus of further research instead of the more computationally intensive SVMs. Lastly, the results of classification using features extracted with the DWFP could be compared to standard features extracted through commercial packages like ICEPak ®(TISEC Inc, Montreal, QC, Canada) to test their effectiveness against standard techniques.

In sum, this dissertation has demonstrated the successful implementation of artificial intelligence techniques to aid the detection capability of four different applications. The method is an improvement over existing signal analysis techniques and may be a general tool for time-domain signals.

Bibliography

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

Vita

Crystal Bertoncini was born in Camp Hill, Pennsylvania on December 6, 1982. She attended Cumberland Valley High School where she graduated in June, 2001 with a 4.0 GPA. She also received an "Excellence in Mathematics" award. Crystal received her Bachelor of Arts degrees in Physics and Mathematics from Vassar College located in Poughkeepsie, New York in May of 2005. At this time, she was also inducted into the Scientific Research Society Sigma Xi. In the fall of 2005, Crystal started graduate studies in the Physics Department at the College of William and Mary, where she worked as a Teaching Assistant. Crystal began work as a Research Assistant in the Nondestructive Evaluation laboratory in the summer of 2006. She completed and successfully defended her dissertation on January 14, 2010, where she then moved to Washington, DC.