INTERPRETABLE CLASSIFICATION MODEL FOR AUTOMOTIVE MATERIAL FATIGUE

By
Kee Khoon LEE

A thesis submitted for the degree of
Doctor of Philosophy

Department of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ
United Kingdom.

June 2002

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING

ELECTRONICS AND COMPUTER SCIENCE DEPARTMENT

Doctor of Philosophy

Interpretable Classification Model For Automotive Material Fatigue

by Kee Khoon LEE

This thesis describes the problem of classifying and predicting fatigue crack initiation sites in automotive material through microstructure quantification and develops machine learning methods to address this task. The work is novel in that it develops machine learning techniques for: 1. handling of imbalanced classification data which recovers an underlying structure, 2. the development of a new understanding of the relationship between the inputs and crack initiation site predictions, hence improving interpretability of the model.

A typical learning machine requires modification to its cost function in terms of misclassification cost and sampling bias in order to deal with imbalanced data. The way the classification rate is obtained may be altered to the geometric mean (Gmean) where it is found to be less sensitive to the skewness in the distribution of the classification rate. These modifications, are then applied to Support Vector Machines (SVM) and various extension techniques. Results on two data sets obtained from camshaft and plain journal bearing linings show that a good Gmean value of 0.70 is achieved. The classification model structure was then decomposed to provide an interpretable model. While SUpport vector Parsimonious ANalysis Of VAriance (SUPANOVA) uses this technique for regression, it has now been extended to classification with imbalanced data to provide a parsimonious (interpretable) model. The original classification model structure of the camshaft and plain journal bearing lining consists of the sum of 512 and 2048 sub-components respectively. With our SUPANOVA, the sum of the sub-components was reduced significantly to 6 for both applications and yet retains a good predictive performance. Initial analysis of this data from the metallurgist user community has focused on univariate components by considering variations in the arithmetic means in each of the individual inputs. Here, the results extend to higher order terms which have been

compared with their understanding of the physical system. To enhance visualisation the results from the SUPANOVA parsimonious models, data on simulated particle distributions were generated. The simulated data set was generated systematically and assessed with parsimonious models. With this knowledge obtained from the modelling, the key microstructural features that optimise these automotive materials' fatigue performance have been identified.

Contents

List of Figures

List of Tables

Acknowledgements

This thesis could not have been made possible without the help and support given by many people. I would like to thank my supervisor, Prof. Chris Harris, for his support and encouragement. Also, I would like to thank Dr. Steve Gunn for his technical support and advice and Dr. Philippa Reed for her support in both coordinating the supply of data and sharing her knowledge of fatigue cracks linking it to our model. I would also like to thank Julian, Mark and Nihong for teaching me and providing me with the data. I am also grateful to Tony and Jason who have provided me with the opportunity to discuss things with them. Financial support from EPSRC Grant No. GR/M13879 and Federal Mogul Technology is gratefully acknowledged.

As for my family, I would like to thank my siblings for taking care of my parents while I am away for this period of time. To my parents, thank you for giving me the opportunity to pursue my dream. Also, to my in-laws for their encouragement and sacrifice in allowing their daughter to join me in the UK. The biggest support I have is from my wife, who made her way from Singapore to join me. Thanks also to my friends Jas and Nadim for their companionship when I felt down.

I would like to dedicate this thesis to my family, who are counting down the days to when I will be back home, and especially to my wife, Ann.

Nomenclature

ADI	Austempered Ductile Iron
AF	Area Fraction
Al-Si-Sn	Aluminum Silicon Tin
AM SVM	Adaptive Margin SVM
AMean	Arithmetic Mean
ANN	Artificial Neural Network
BPDN	Basis Pursuit De-Noising
CART	Classification And Regression Tree
COD	Curse Of Dimensionality
CS SVM	Control Sensitivity SVM
ERM	Empirical Risk Minimisation
FBT	Finite Body Tessellation
FLD	Fisher Linear Discriminant
FN	False Negative
FP	False Positive
GACV	Generalised Approximation Cross-Validation
GCKL	Generalised Comparative Kullback-Lieller distance
GP	Gaussian Processes
GMean	Geometric Mean
KKT	Karush Kuhn Tucker
LOO	Leave One Out
LVQ	Learning Vector Quantisation
MAP	Maximum A Posterior Estimate
MC	Misclassification Cost
MOF	Method Of Frames
MP	Most Probable Hyperparameter
NSS SVM	Non-Standard Situation SVM
QP	Quadratic Programming
RA	Retained Austenite
RBF	Radial Basis Function
RN	Regulariation Network
ROC	Receiver Operating Characteristic
SD	Standard Deviation
SLT	Statistical Learning Theory
SRM	Structural Risk Minimisation
SUPANOVA	SUpport vector Parsimonious ANalysis Of VAriance
SVs	Support Vectors
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

te	subscript "te" denotes testing set
tr	subscript "tr" denotes training set
VF	Volume Fraction
VC	Vapnik-Chervonenkis Dimension
$\mathbf{x}$	Input Vector
$\mathcal{X}$	Input Space
$\mathbf{y}$	Target Vector
$\mathcal{Y}$	Target Space
$\mathbf{w}$	Weight Vector
b	Bias offset
N	Number of Input Dimension
ℓ	Number of Training data
k	Number of Classes
T	Transpose Operator
$\Re$	Real number
$J(\mathbf{w})$	Fisher Criterion
P_k	the k^{th} class mean on the projected data
S_k^2	the k^{th} class within class covariance matrix
$\mathbf{m}_k$	the k^{th} class mean vector
S_w	total within class covariance matrix
$\mathbf{p}(.)$	Probability Density Function
$P(.)$	Posterior Probability
$\mathbf{f}(\mathbf{x}, \alpha)$	a class function approximation with parameter α
$R_{emp}(\alpha)$	Expected empirical risk function associated with α
$\mathcal{L}(y, \mathbf{f}(\mathbf{x}, \alpha))$	Loss Function associated with a class function
$1 - n$	the confidence level on the expected risk function $R(\alpha)$ to lie with a given bound
S_n	Nested Structure for SRM as n gets large the complexity increases
R	Radius
h	VC Dimension
$\mathcal{H}$	Hilbert Space
Λ	Abstract of element
$\mathcal{F}$	Feature space induced by a kernel
$K(.,.)$	Kernel Matrix
A	The bounded weight coefficient
σ	width of the RBF
$\phi(\mathbf{x})$	Basis Function
Φ	Basis Function Matrix
$Q(\alpha)$	Regularisation Function
λ	Regularisation Parameter
Λ	Set of parameters
τ	Margin separating the class
$\mathbf{w}_{opt}$	Optimal weight
α	Lagrange Multiplier
ξ	Slack Variables
C	Capacity Control
$G(.)$	Convex function (i.e G(0)=0 and always a postive value)
$\mathcal{L}_\epsilon$	ϵ-insensitive loss function

ϵ	a prescribed parameter that represents the allowance for the errors
$\mathcal{L}_q$	Quadratic Loss Function
$\| \mathbf{x} \|_p$	p-Norm (i.e. $=(\sum_{i=1}^{N} \|x_i\|^p)^{\frac{1}{p}}$)
$(.)_+$	A positive argument (i.e. equal $max(a, 0)$)
$\bar{\alpha}$	Dual Problem Solution
$\ell(\mathbf{w}, b, \alpha)$	Lagrange Function
$\phi(\mathbf{w})$	Cost function
$E(.)$	Expected risk of error
π	The prior probability of the target
L	The imbalance modification factor
J	Number of basis function
$\mathbf{a}$	Weighted coefficients for sparse representations
m	Number of additive kernels
$\rightarrow$	Vector
$\mathbf{I}$	Identity Matrix
$\mathbf{1}$	Vector of ones
$\mathbf{0}$	Vector of zeros
diag	Diagonal Matrix
COV_{dmean}	Clustering measurement for particles (i.e. $\frac{SD\ d_{Mean}}{Mean\ d_{Mean}}$)

Abbreviations for FBT Measurements

O.A	Object Area
O.A_r	Object Aspect Ratio
O.Ang	Object Angle
C.A	Cell Area
C.A_r	Cell Aspect Ratio
C.Ang	Cell Angle
L.A.F	Local Area Fraction
N.N.N	Number of Near Neighbours
d_{Min}	Nearest Neighbour Distance
d_{Mean}	Mean Near Neighbour Distance
N.N.Ang	Nearest Neighbour Angle

Abbreviations for Simulated Data sets

This consists of four letters :

First letter denotes the material used : A-ADI and B-Al-Si-Sn

Second letter denotes the Object Distributions : R-random and C-clustered

Third letter denotes the Object Area : C-constant and V-Varying

Final letter(s) denotes the Shape and Angle of the objects : C-Circular shape with no object angle and E-θ-Ellipse shape with an object angle θ to its loading axis

ARCC	ADI ; Random Object Distribution ; Constant Object Area ; Circular shape
ACCC	ADI ; Clustered Object Distribution ; Constant Object Area ; Circular shape

ARVC	ADI ; Random Object Distribution ; Varying Object Area ; Circular shape
ACVC	ADI ; Clustered Object Distribution ; Varying Object Area ; Circular shape
ARCE-90^o	ADI ; Random Object Distribution ; Constant Object Area ; Ellipse shape with object angle of 90^o along loading axis
ARCE-0^o	ADI ; Random Object Distribution ; Constant Object Area ; Ellipse shape with object angle of 0^o along loading axis
ACCE-90^o	ADI ; Clustered Object Distribution ; Constant Object Area ; Ellipse shape with object angle of 90^o along loading axis
ACCE-0^o	ADI ; Clustered Object Distribution ; Constant Object Area ; Ellipse shape with object angle of 0^o along loading axis
ARVE-θ	ADI ; Random Object Distribution ; Varying Object Area ; Ellipse shape with object angle of θ along loading axis
ACVE-θ	ADI ; Clustered Object Distribution ; Varying Object Area ; Ellipse shape with object angle of θ along loading axis

Note: A similar set of simulated data were generated for the case of Al-Si-Sn except the first letter is changed to B

Chapter 1

Introduction

1.1 Imbalanced Data and Model Interpretability for Classification

In real world applications data sets with limited samples are available. The problem of having small samples of data is further complicated by imbalanced data. For example, in a fault diagnostic and conditional monitoring problem,is it possible to obtain equal amounts of data for the positive and negative cases? Positive examples are usually difficult (and practically undesirable) to obtain, time consuming and costly. Can a statistical learning algorithm give a good prediction based on such imbalanced data? Would you be convinced by a derived model if it provides a good prediction and yet no understanding of the combination of input features which lead to this prediction? Generating a parsimonious model and yet retaining a good predictive performance is a desirable solution to the above problem. This thesis investigates the above two issues namely: imbalanced data and model interpretability for a classification system. The Support Vector Machine (SVM) approach and its associated extension techniques have been considered for two data sets on fatigue crack initiation features obtained from automotive material examples.

The accuracy of the probabilistic density estimation task depends on the input dimensions. As the input dimensions increase, the amount of data required must grow exponentially in order to provide consistent model estimation. The goal of most classical classification techniques is based on having a good density estimation of its data (Vapnik 1995). In real world problems, the amount of data is always limited. Hence, good probabilistic density estimation is difficult to achieve. As

such, the learning algorithm is required to handle the problem of small samples of data. Statistical Learning Theory (SLT) effectively describes statistical estimation with small samples (Cherkassky & Mulier 1998). The key ingredient of SLT is the use of the Structural Risk Minimisation (SRM) principle, which defines the tradeoff between complexity of approximation function and quality of the training data fitted. The generalisation of the model developed by SLT is achieved by the ability to control the set of approximation functions. A learning machine known as a Support Vector Machine (SVM) was developed based on this concept and is described further in chapter 3.

In classical SVM, as in most learning algorithms, its goal is to achieve a greater accuracy, assuming the misclassification costs of individual class are the same and there is no sampling bias. Imbalanced data is the problem when one class is heavily represented whilst the other is under represented. As such, the training distribution for each class may be pre-specified instead of being randomly selected, resulting in a sampling bias (i.e. violation of equal probability of selection principle of the populations). Another issue which is strongly related to the imbalanced data is the misclassification cost. Imposing a misclassification for each class reflects the importance of each class. Furthermore, using the Arithmetic Mean (AMean) for measuring the performance criteria for the imbalanced data is biased towards the majority class. As such, a more appropriate criterion is the Geometric Mean (GMean) which is less affected by extreme values (i.e skewness distribution). For an imbalanced data set, the assumption used in the classical SVM requires an appropriate modification. This modification is done via the SVM parameters (i.e. the capacity control) in order to obtain a good prediction. The above issue of imbalanced data have been investigated using several SVM (and extension techniques) are described in more detail in chapter 4.

Model interpretability is an important issue in classification if one would like to know about the input/output relationship in the model. Much of the work done on Artificial Neural Networks (ANN) is considered as a "black box" classification as it is difficult to explain simply or qualitatively the trends that the output has

determined. SUpport vector Parsimonious ANalysis Of VAriance (SUPANOVA) uses the idea of an ANOVA kernel to enforce a sparse representation of the model structure. The flexibility of the model lies in the use of the spline kernels and the sparseness relies on the norm which is used to enforce the penalty. The nature of the formulation of the ANOVA framework also favors small order terms being selected as all the univariate terms are required to pass through its origin. As such all the higher order terms are constrained to be zeros along these axes. The model structure is thus decomposed into a sum of smaller order terms that can provide easy visualisation and interpretability of the model. The original work of SUPANOVA was applied to regression tasks. In this work it has been extended to a classification problem with imbalanced data. This approach has been applied to two data sets obtained from automotive materials, namely, the camshaft and the plain journal bearing lining. The theory and algorithm developed for SUPANOVA for classification with imbalanced data is described in chapter 5. The results and discussion based on analysis of the two materials data sets are then described in chapter 6.

To further visualise the SPANOVA parsimonious model obtained, a particle simulation was used. The simulated particle distributions provided a systematic way to vary the parameters (e.g. the object area, the object shape, the object distribution and object angle) selected by the SUPANOVA model. By attempting to vary these inputs separately, a parametric assessment of the model predictions can also be achieved. These components selected as contributing to crack initiation have been compared with the understanding of the professional metallurgist and this further understanding can be used to optimise automotive materials performance. The detailed procedure, justification of each simulated data set and the discussion of the results obtained from this simulated data are described in chapter 7.

Our approach can also be applied to other real world problems such as in many fault diagnostic and condition monitoring problems where the data sets are usually imbalanced and a simple parsimonious model with easy interpretability is a desired outcome. As such, the techniques described here are broadly applicable.

1.2 Thesis Overview

This thesis explores the following: imbalanced data and model interpretability, applied to the real world application problem of fatigue crack initiation in automotive components. The outline of this thesis is as follows :

Chapter 2 - Fatigue and Microstructural Quantification Techniques This chapter describes the practical rationale for the work in this thesis. The cause and catastrophic effects of fatigue are first briefly described. Fatigue crack initiation can be captured by a microstructure quantification technique. The Finite Body Tessellation (FBT), an example of a microstructure quantification technique, produces a set of features that describe the prior domain knowledge of the microstructural distribution (e.g. morphology of secondary particles and the particles spatial distribution). This set of features is described in more detail in section 2.2. Two typical components in which fatigue crack initiation is an important issue are addressed here by looking at the fatigue of materials used in the camshaft (ADI) and plain journal bearing lining (Al-Si-Sn). Preliminary results use simple visualisation (e.g. comparing means and standard deviations) and physical understanding. However, this may not help to explain the dependency observed between large numbers of potentially independent variables (i.e features). Therefore a review of adaptive numerical modelling (especially Artificial Neural Network (ANN) approaches - which provide flexible data based models) commonly used in the material science field is then provided. A classical approach for classification, the Fisher Linear Discriminant (FLD), is also outlined. These approaches however may not be appropriate for small sample data sets as it usually requires large amount of data to be available. This is considered in more detail in the next chapter.

Chapter 3 - Learning from Data One of the goals for the ANN approach is to estimate the probabilistic density of the data. This requires an exponential increase of the number of training data as the input dimension increases to provide consistent results. This chapter provides the basis of constructing

a learning machine for small sample data sets. This is developed on the probabilistic dependency between the (input,output) from a class of functions restricted by the number of data pairs and is known as Statistical Learning Theory (SLT). To construct a learning machine with SLT, four important components, (namely, the learning task, induction principle, decision function and the algorithm to implement the aforementioned ingredients) are required and are described in section 3.4 to 3.7. The induction principle in SLT relies upon the Structural Risk Minimisation (SRM) principle. An understanding of SRM will provide an explanation for the misunderstanding between the conceptual and technical implementation for a pattern recognition problem. A learning machine built from SLT and the kernel methods is the Support Vector Machine (SVM). This is described in section 3.7 including a review of the use of SRM in SVM and Regularisation Network (RN) for which both use kernel methods. The kernel provides a mapping from input space to a high dimensional feature using its dot product. The work on SVM originated from a classification case and it can be extended to multi-class and regression estimation. The generalisation issue of SVM is related to tuning the parameter which was mainly based on minimising the bound of the expected risk and is described in the last section of this chapter. A common problem with classification problems is that of imbalanced data. Can the standard learning machine built be used for imbalanced data? This is further developed in chapter 4.

Chapter 4 - SVM for Imbalanced Data The conventional learning machine is built on the assumption that the misclassification costs of each individual class are the same and there is no sampling bias between training and testing. This chapter starts with the description of problems associated with imbalanced data and how to deal with it. Furthermore, the Arithmetic Mean (Amean) commonly used to measure the performance criteria may not be appropriate for assessing the performance with imbalanced data. Therefore, other performance measurement criteria such as the Geometeric Mean (Gmean) which

are less sensitive to large changes between their classification rate performance are also discussed in section 4.2. When classifying an imbalanced data set, the assumption used in the conventional learning machine requires an appropriate modification for the above two assumptions. For the case of using the SVM, this modification can be implemented via the SVM parameters (i.e. the capacity control) in order to obtain a good prediction. The following sections of this chapter describe several SVM extension techniques such as Control Sensitivity (CS) SVM, Non-Standard Situations (NSS) SVM and Adaptive Margin (AM) SVM which offer different ways to deal with imbalance of data in a SVM framework. The classification model structure produced here is difficult to interpret. The issue of model interpretation is then developed further in chapter 5.

Chapter 5 - Model Interpretation for classification Non-parametric models such as classical artificial neural networks (ANN) can be considered as a "black box" model. This kind of model is difficult to interpret. This chapter is concerned with model interpretability for classification within the SVM framework. This is done by decomposing the model structure in feature space into a smaller subset of its input variables described in section 5.2. The interpretability comes from using the spline kernel with a norm for enforcing sparseness of the model structure in the feature space. Inherently, the ANOVA favours the selection of lower order terms and provides a parsimonious model that is easy to interpret. This is the motivation of the work based on SUPANOVA which was then developed for the decomposition of the model structure of the kernel in SVM for regression problems. Here, its use is extended to the classification of imbalanced data using the appropriate hinge loss function and the assumption for imbalanced data as described in the previous chapter (i.e. misclassification cost and sampling bias). This approach will provide enforced sparseness of the kernels in the feature space to provide model structure interpretability for the imbalanced data model. As such, a smaller sum of the sub-components can be obtained and these can therefore be identified as the

important features. The features obtained from FBT described in chapter 2 (i.e. for the two automotive materials) are then used to apply the techniques described in chapter 4 and 5 for classifying and predicting fatigue crack initiation. Chapter 6 describes the model specification and results obtained.

Chapter 6 - Data Analysis This chapter uses the Finite Body Tessellation (FBT) data obtained from the automotive material (Chapter 2) used for the Camshaft (ADI) and the Plain Journal Bearing Lining (Al-Si-Sn). This chapter begins with an outline of the model specification for the ADI. Prior to using the SVM framework described in Chapter 3.8, the Fisher Linear discriminant (FLD), a simple and classical approach for classification, was investigated. A comparison of results obtained between FLD and SVM extension techniques (described in chapter 4.3) then follows. The comparisons are then extend to the SVM extension techniques that incorporate the modifications for imbalanced data (i.e misclassification cost and sampling bias implemented through the capacity control of the SVM). The best result for dealing with imbalanced data is then extended to provide model interpretability using the SUPANOVA for classification of imbalanced data (described in chapter 5). This produces a parsimonious model which comprises of a sum of a smaller set of sub-components. The sub-component plots are then discussed in section 6.2.1 with attempts to link this trend behaviour with the metallurgists' understanding. This process is then repeated for the case of Al-Si-Sn.

Chapter 7 - Simulated Data Analysis The parsimonious models produced by the SUPANOVA are still quite complex. It is necessary to vary the input features systematically in order to enhance our understanding of our model. Assessing the effect of the parsimonious components selected by our SUPANOVA can be done via particle distribution simulations. There are four main variations possible in the input parameters (e.g. the object shape, object area, object distribution and object angle) that are related to those components selected. This chapter starts off with the justification of the simulated data to be generated corresponding to the SUPANOVA components selected. This

is followed by a detailed description of the procedure and specification of how to generate the simulated data set in section 7.2. This simulated data is then used as the test set in the SUPANOVA model. The results are then presented in section 7.3, where the enhanced visualisation of the model now offered for the two sets of automotive materials investigated are discussed. The final section in this chapter makes a comparison of the results obtained from inspection of the SUPANOVA components and the predictions of fatigue initiations in the simulated particle distributions.

Chapter 8 - The Conclusions and Future Work This chapter provides a summary of the work presented in this thesis. The approaches used are discussed. Future extensions to improve the current work from both the modelling and metallurgists' point of view are also described.

1.3 Research Contributions

In development of the standard learning machine the main aim is to achieve maximum accuracy, assuming no sampling bias between the training and testing data and that the misclassification costs are equal. The performance criteria for a classification problem therefore uses the Arithmetic Mean (Amean). This leads us to make a necessary modification to the above assumptions of the machine learning for imbalanced data. Furthermore, in a classification model interpretability is often neglected. It is important to understand and visualise what trend of the (input,output) will initiate a fatigue crack in automotive material. This allows a more micromechanistic understanding to be built up and hence fatigue resistance to be optimised. A set of simulated data that produced a particle by particle distribution was used to visualise and extrapolate the model produced, by varying individual features of the particle distribution systematically. With the knowledge obtained from the modelling, the key production and microstructural features that will optimise automotive materials' performance can be obtained. Although this work has concentrated on automotive material performance, there is a diversity of real world application problems which require similar approaches. As such, the techniques described in this thesis are broadly applicable.

The main contributions of this work are based on the extension of the SVM framework to provide a model interpretation in a classification scenario which has imbalanced data. The work of this thesis has contributed in part or full to the following publications :

- K K Lee, C J Harris, S R Gunn and P A S Reed (2001). Classification of Imbalanced Data With Transparent Kernel, INNS-IEEE International Joint conference on Neural Network (IJCNN), Washington DC U.S.A, July 2001, pg 2410-2415.
- K K Lee, C J Harris, S R Gunn and P A S Reed (2001). Regression models for classification to Enhance interpretability, Proceeding of the 3rd International Conference on Intelligent Processing and Manufacturing of Materials (IPMM), Vancouver Canada, July/Aug 2001.
- K K Lee, C J Harris, S R Gunn and P A S Reed (2001). Control Sensitivity SVM for Imbalanced Data : A Case Study, 5th International Conference on Artificial Neural Networks and Genetic Algorithm (ICANNGA), Prague Czech Republic, April 2001.
- K K Lee, C J Harris, S R Gunn and P A S Reed (2001). Approaches to Imbalanced Data for Classification : A Case Study, International ICSC Congress on Computational Intelligent : Methods and Applications (CIMA) in Advances in Intelligent Data Analysis (AIDA), University of Wales Bangor, June 2001.
- K K Lee, C J Harris, S R Gunn and P A S Reed (2001). A Case Study of SVM Extension Techniques on Classification of Imbalanced Data, Congress on Neural Networks and Applications, Fuzzy Sets and Fuzzy Systems and Evolutionary Computing, Tenerife Spain, Feb. 2001. Eds. : Nikosmastoraks in the world Scientific and Engineering Artificial Intelligent Series, Advances in Neural Networks and Applications, pg 309-314.
- P A S Reed, R C Thomson, J S James, D C Putman, K K Lee and S R Gunn (2001). Microstructural effects in the fatigue of austempered ductile iron. Submitted to Journal of Materials Science and Engineering (Nov 2001).

- P A S Reed, K K Lee, C J Harris and S R Gunn (2002). Interpretable models for classification of fatigue crack initiation sites. Work in progress, to be submitted to Journal of Materials Science and Engineering.

Chapter 2

Fatigue and Microstructural Quantification Techniques

Approximately 90% of metallic failures are caused by fatigue (Callister 1997). Fatigue can result in a catastrophic failure caused by the initiation and growth to final failure of cracks in material subjected to fluctuating stresses (e.g. bridges, aircraft, train tracks and many machine components). Therefore, this chapter begins with a description of the importance of understanding the fatigue process. In this work, fatigue initiation is analysed through microstructure quantification techniques using Finite Body Tessellation (FBT). FBT provides a set of features that describe the prior domain knowledge of the microstructural distributions (e.g. morphology of secondary particles and the particles' spatial distribution). This is described in section 2.2. The importance of this analysis in an automotive material is demonstrated by looking at two applications (i.e. a camshaft and plain journal bearing lining). The experimental testing conditions used together with the preliminary results are described in the subsequent section. The ability to simulate the microstructural distribution of a secondary phase (e.g. in terms of the particles' distribution, size, shape, orientation) may provide a quantitative assessment and visualisation of the importance of such features in initiating fatigue. The method used to produce simulated particle distributions is briefly outlined in section 2.4.

The development and processing of materials is a complex process. Simple visualisation of the links between processing, microstructure and the resultant mechanical properties cannot be made easily without considering the effects of many variables.

As such, non-linear techniques such as artificial neural networks (ANN) have been used fairly recently in materials science to try to predict a range of processing-property relationships. An overview of these applications is provided along with the description of the classical approach to classification (the Fisher Linear Discriminant (FLD)). However, these techniques usually require large amounts of data which may not be available.

2.1 Fatigue

The failure of engineering materials is usually an undesirable event since it might involve loss of human life, economic loss and complete stoppage of work in order to replace failed components. The usual cause of this failure is the improper selection of materials for the service conditions, which includes inappropriate processing, inadequate design of components and misuse. Hence, appropriate prevention is vital against such failure incidents. The focus of this work is upon fatigue crack initiation in automotive materials. The term "Fatigue" is derived from the fact that failure occurs after a lengthy period of repeated stress or strain cycling. The process of fatigue failure in metals is characterised by three typical stages: (1) crack initiation, where a microcrack develops on the metal's surface at a point of high stress concentration; (2) crack propagation, where the crack length increases with each stress cycle; (3) final static failure, which occurs very rapidly when the critical crack size is reached (Suresh 1998). The crack initiation is of key importance, since crack propagation will not occur prior to this. The crack initiation site may include surface scratches, keyways, dents and sharp fillets. In the absence of mechanical stress raisers, microscopic surface discontinuities are produced under cyclic loading resulting from dislocation slip steps which may act as sufficient stress raisers. Secondary phase particles within a metal may also act as initiation sites, when dislocations impinge on hard phases leading to microscopic stress concentrations. In order to investigate how fatigue cracks initiate, a series of cyclic stress tests can be applied to a material. A bend test has been used in this work, using three point loading techniques. A bar or flat strip specimen of rectangular cross section is cyclically loaded until crack initiation is observed (see Fig. 2.1). During the fatigue test,

the specimen microstructure can be examined periodically using acetate replicas to identify crack initiation sites. In materials containing secondary phases the initiation events maybe related to the particles and quantifying their distribution is therefore important.

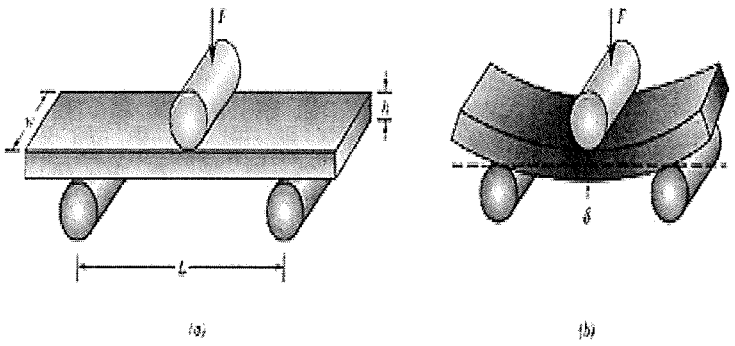


Figure 2.1: Three point flexural fatigue test geometry. At the point of loading, the top surface of the specimen is placed in a state of compression and the bottom surface is in tension, and it is this region of maximum tension that is observed closely as fatigue cracks will initiate here. Fatigue occurs when this specimen is cyclically stressed (i.e. with repeated bending), cracks initiate and propagate through the metal thickness to a point where the remaining sound structure fails by ordinary rupture (because the applied load can no longer be supported).

2.2 Microstructural Quantification Techniques

Various methods have been developed for characterising the microstructural distribution of discrete secondary phase bodies on two-dimensional sections. They include field methods (Vander Voort 1990), inter-particle spacing methods (Schwarz & Exner 1983) and tessellation methods (Mray *et al.* 1983). The field methods provide a broad evaluation of the microstructural distribution scales (using information about the particle density of varying test areas). The inter-particle spacing methods on the other hand describe more about the types of distribution and local clustering (using the measurement obtained from the nearest neighbour distances between particle centroids). Tessellation, a microstructure quantification technique, provides a particle-by-particle analysis of the distribution of secondary phase bodies (e.g. graphite nodules) rather than the overall distribution of such particles.

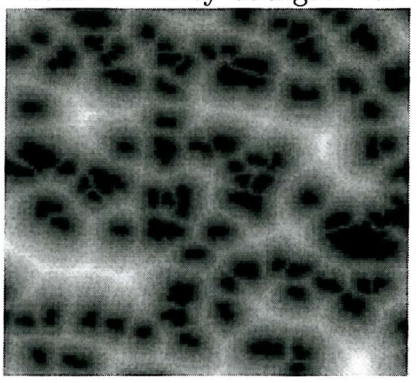
Finite body tessellation (FBT), an extension of Dirichlet tessellation, was introduced by (Boselli *et al.* 1999). The Dirichlet tessellation cells are constructed

based on the particles centroids. It is best applied in the representation of circular objects with a narrow size distribution. If the size of the object is large and its nearest neighbour close, then the side of its cell may intersect with the object (Spitzig *et al.* 1985). Furthermore, the restriction to effectively circular objects restricts its application. In the FBT, the tessellated cells are constructed from the actual interfaces of the object. It is subject to the constraint that every point within the cell is closer to the interface of its corresponding body than any other. Figure 2.2 shows three stages involved in the FBT procedure: binarisation of the image, a distance transformation and a watershed transformation. The image captured typically contains a noisy background. During this process, some of the edges of the secondary body may be discontinuous and must be corrected. A morphological filter is then used to fill holes within the bodies. This is a rather tedious stage as it involves particle by particle filling and also knowledge of the "correct" microstructure, i.e. expert filtering also occurs. The distance transformation which is the heart of the tessellation technique, converts a binary image consisting of feature and non-feature pixels into a greyscale image where every non-feature pixel is assigned a grey value that approximates the distance to the nearest feature pixel (Borgefors 1986). There are six common distance transformations namely; city block, chessboard, octagonal, chamfer 3-4, chamfer 5-7-11 and Euclidean. It has been shown that by using the chamfer 5-7-11, a reasonably high accuracy can be obtained (Boselli *et al.* 1999, Borgefors 1986). A watershed transformation is used to generate thin divisions between the objects (Vincent & Soille 1991).

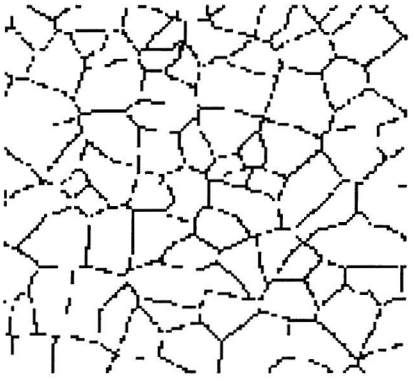
A set of measurements relating to the spatial distributions and morphology of the object can be obtained from tessellation. Figure 2.3 shows the definitions of these measurements from FBT. The measurements available are:

1. Object area, (O.A)
2. Object aspect ratio, ($O.A_r$)
   aspect ratio of the object (maximum chord length divided by maximum width perpendicular to the maximum chord length);
3. Object angle, (O.Ang)
   reference from the horizontal axis with the maximum chord length of the object (between 0 and $\frac{\pi}{2}$ radians);
4. Cell area surrounding the object, (C.A)

(a) A sample of the original microstructural image in binary format.
The grey images are thresholded in order to obtain a true representation of the binaries from the noisy background.

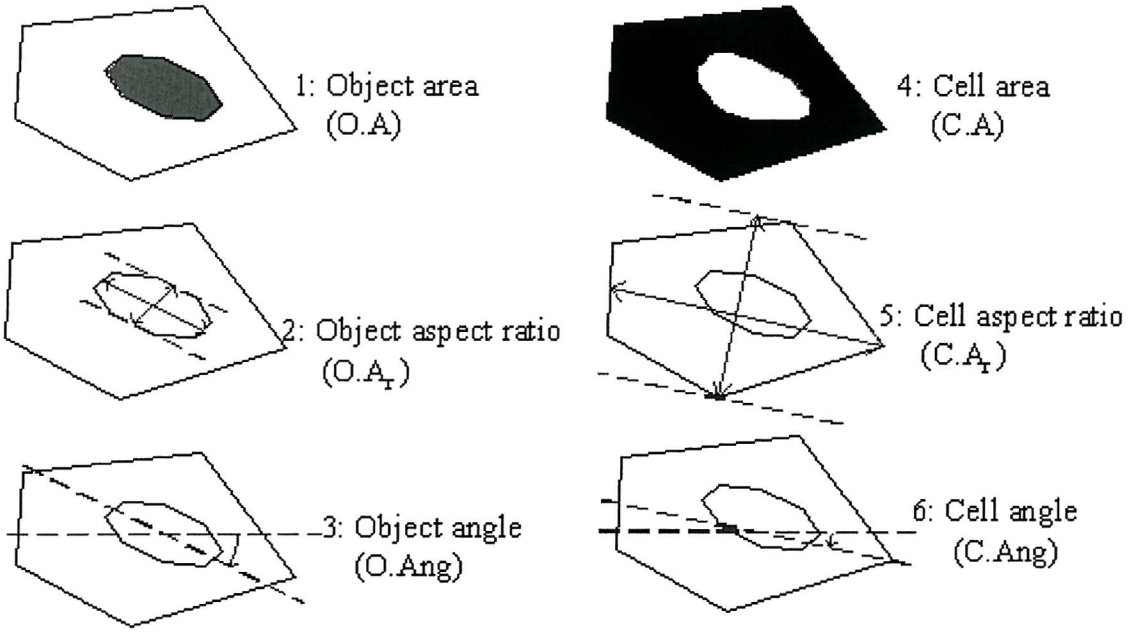


(b) The sample after the distance transformation.
The binary image consisting of feature and non-feature pixels is converted into a greyscale image where every non-feature pixel is assigned a grey value that approximates the distance to the nearest feature pixel.

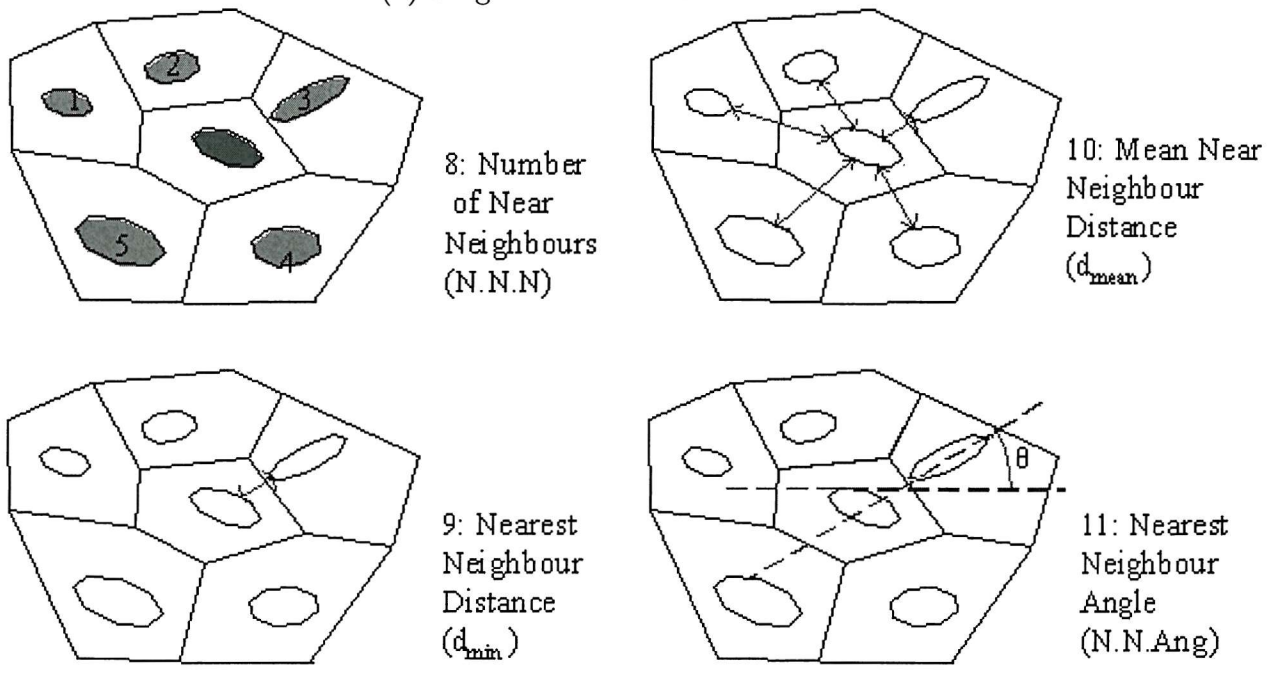


(c) The watershed images.
This draws a thin line dividing the objects, thus defining the C.A associated with each object.

Figure 2.2: Successive steps to obtain a Finite Body Tessellation (FBT).



(a) Single cell measurements



(b) Cell and near neighbour measurements

Figure 2.3: Definitions of the FBT measurements. These describe the spatial distributions and morphology of the objects.

5. Cell aspect ratio, ($C.A_r$)
   aspect ratio of the cell (maximum chord length divided by the maximum width perpendicular to the maximum chord length);
6. Cell angle, (C.Ang)
   angle of the cell's longest chord with respect to the horizontal (between 0 and $\frac{\pi}{2}$ radians);
7. Local area fraction, (L.A.F)
   area of object/area of associated cell;
8. number of near neighbours, (N.N.N)
   number of objects sharing a cell boundary with object of interest;
9. nearest neighbour distance, (d_{Min})
   the minimum edge to edge distance between the object of interest with any of its neighbours;
10. mean near neighbour distance, (d_{Mean})
    average of the minimum edge to edge distance between the object of interest with all its neighbours;
11. nearest neighbour angle, (N.N.Ang)
    the angle between the horizontal axis and the centroid of the object of interest with its nearest neighbour (between 0 and $\frac{\pi}{2}$ radians).

Identifying the importance of each measurement in fatigue initiation requires a interpretable model for classification, this will be described in chapter 5. With this knowledge, the key microstructural features that will optimise automotive materials fatigue performance can be obtained.

2.3 The Industrial Applications

Most machine components are subjected to fluctuating stresses leading to fatigue. In this work, we investigate the automotive materials used in camshafts and plain journal bearing linings (Hockley *et al.* 1999). The motivation for investigating these two components in automotive applications is as follows :

1. Camshaft

   The Camshaft controls the opening and closing of the poppet valves in a combustion engine. Modification of its motion from sliding to rolling contact will provide better power and fuel efficiency. However, the contacting surface of the roller camshaft requires resistance to rolling fatigue, high strength and ductility. Figure 2.4 shows the roller-follower design Camshaft. Mechanical property understanding of the heat-treatment effects and crack initiation within the selected material is required. For certain heat-treatments, service

conditions may lead to multiple fatigue crack initiation sites that greatly accelerate crack growth. Therefore understanding potential crack initiation sites through evaluation of fatigue tests for a selected material is very important.

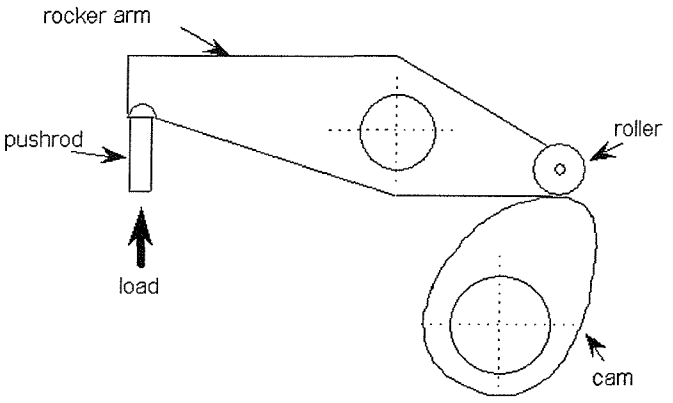


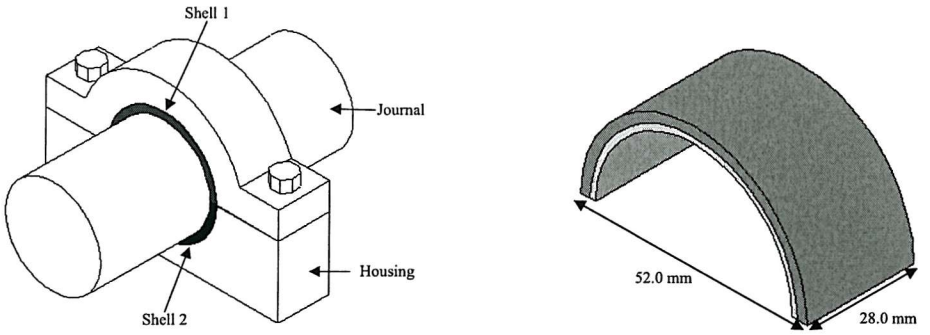
Figure 2.4: The roller-follower design camshaft.

2. Plain journal bearing lining

   A modern plain journal bearing lining consists of two half shells which are clamped together within the bearing housing to support the journal between them. Each shell comprises several layers of different materials (see Fig. 2.5). A bearing must have a long life span. There are several factors affecting the life span of a bearing: the loads on the bearing, the lubrication used (hydrodynamic pressure), fit of the bearings on the shaft and in the housing, friction coefficient of the materials and the material used, etc. The plain journal bearing lining we are investigating here requires the production of high output transmission and must be able to withstand the varying hydrodynamic pressure of the journal in the automotive engine. In order to do this, it is necessary to investigate both the loading conditions and the fatigue failure behaviour of the bearing's lining material.

2.3.1 Experimental and Testing conditions

Prior to discussing some of the results obtained by Hockley and Joyce *et al.* at Southampton, the two experimental testing conditions used are briefly described here, alongside a brief discussion of the current metallurgical understanding of the materials used in these automotive applications.



(a) Bearing mounting (b) Plain journal bearing lining

Figure 2.5: The plain journal bearing lining mounted on a housing to support the journal between them.

1. Camshaft

   Cast iron is a cheap metallurgical substance with good mechanical rigidity and strength under compression. The mechanical strength and toughness of cast iron can be improved by altering the graphite flakes to a spheroidal-graphite shape (produced by adding a small amount of either cerium or magnesium to molten iron just before casting). The metal matrix is a complex mixture of different microconstituents such as, ferrite, retained austenite (RA), carbides, cementite and bainite surrounding the graphite nodules that will affect mechanical properties. The relative proportions of these microconstituents can be altered by subsequent solid-state heat-treatment. Austempering (a low temperature heat treatment carried out after a high temperature austenitising step)is used to refine the microstructure and produce a more uniform and desirable size distribution of matrix phases. Furthermore, it relieves stress, whilst reducing brittleness and hardness of the matrix structure. Hence, Austempered Ductile cast Iron (ADI) can be used in the camshaft application with appropriate adjustment of the heating parameters and material composition.

   (Hockley *et al.* 1999) have shown that austenitising at 950^oC for 1 hour and austempering at 400^oC for 2 hour yields good fatigue resistance. This is due to the coarse bainitic lath structure and increased RA content. Reducing the austempering temperature to 250^oC increases the strength but decreases

the fatigue resistance. This leads to a trade off between strength and fatigue resistance. However, the wear requirements of the camshaft requires high strength/hardness so the austempering temperature of $250°C$ is likely to be used, but is known to produce multiple fatigue cracks.

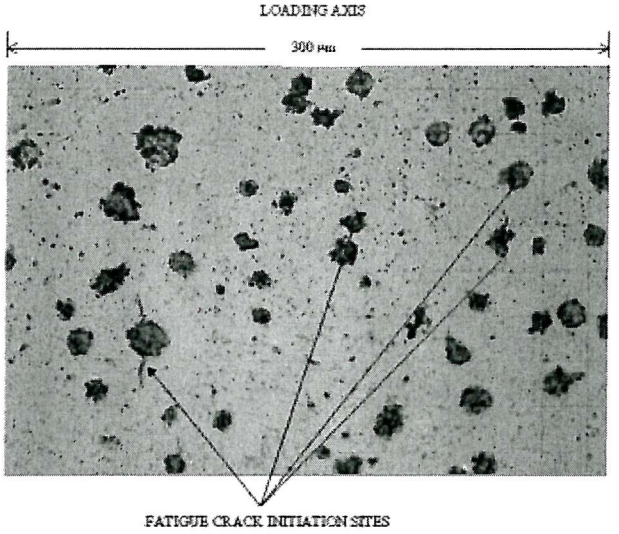
Microstructural quantification provides a means of understanding fatigue damage evolution through assessing the spatial distribution of those graphite nodules that initiate cracks. However, there will be other effects on the mechanical properties due to ADI microstructure, such as:

- the volume fraction of RA that is present in the matrix;
- the shape, size and distribution of the bainite phases; and
- the presence of carbides.

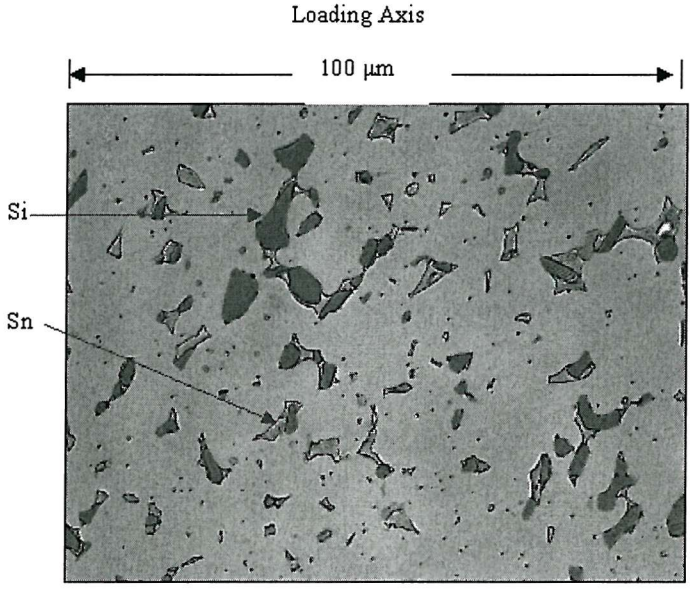
The effect of these factors on crack initiation has not been taken into account. Here, we assess only the graphite nodule morphology. This seems to be reasonable as the majority of fatigue cracks (95%) were associated with graphite nodules (See Fig. 2.6a) in the 850/250 condition.

2. Plain journal bearing lining

The shell of the plain journal bearing lining studied here consists of three material layers, namely, Aluminum-Silicon-Tin (Al-Si-Sn) alloying lining (0.244 mm), Al interlayers ($0.06 \pm 10\%$ mm) and Steel backing (1.505 mm). The fatigue behaviour of the plain bearings is dependent on many complex factors. Loading is via the oil layer separating the bearing surface from the journal. The behaviour of this hydrodynamic oil film causes discontinuous and rapidly changing stress fields to be set up across the bearing surface. The Al-Si-Sn lining comes into direct contract with the hydrodynamic pressure. Hence, the crack is likely to be initiated from this layer and it is important to investigate the dependencies of the material/component combination that initiates these cracks. It we seen that fatigue crack initiation in this material was exclusively associated with the debonding of the Si phase from the surrounding matrix (Joyce 2001). In this work, the distribution of the secondary phase of the Si is taken into account only (see Fig. 2.6b).



(a) Crack initiation at spheroidal graphite nodule in $850/250^{o}C$ condition (length of 300μm). Acetate replica of polished surface. The majority of crack initiations were observed to be from graphite nodules.



(b) Optical microscopy of the Al-Si-Sn alloy lining material showing spheroidised Si distribution with recticular Sn (length of 100μm). The Sn phase occasionally encapsulates the Si.

Figure 2.6: Optical microscopy of the automotive camshaft (ADI) and plain journal bearing lining (Al-Si-Sn). A crack initiation point could be veried by assessing the repica record.

Short crack tests were carried out in both materials under a three point bend plain rectangular bar configuration (described in section 2.1) as multiple cracks were expected. This reduced the area of likely crack initiation sites and thus area of crack monitoring. Interrupted cycling and acetate replication were used to monitor the microstructural features initiating the fatigue cracks. A crack initiation point could be veried by assessing the repica record. Polished plain bend bars with dimensions of 10mm x 10mm x 70mm were used in the short crack testing for the ADI camshaft material. For the case of the plain journal bearing lining a 80mm x 20mm test specimen was obtained from the flat strip material produced prior to bearing formation and the lining surface ground down to about 0.25mm. Interrupted cycling and acetate replication was also used to monitor the microstructure and the initiation of short fatigue cracks on the specimen surface. The graphite nodule and the Si distributions for those initiating cracks and those not initiating cracks are then assessed using the tessellation techniques described in section 2.2.

2.3.2 Preliminary Results

The experimental results are given below for each case.

1. Camshaft

   In the $850/250^oC$ condition, about 116 "crack" initiation sites and 2803 "no crack" sites at graphite nodules were found in one sample. The surrounding particles may affect the microconstituent formation (discussed previously) which may also affect the crack initiation. However, 95% of these initiation sites were due to the graphite nodules and the rest were within the surrounding matrix. Preliminary results comparing simple means of the features obtained from FBT, shows that initiation of cracks occurs for larger nodules of individual high O.A surrounded by a relatively low average O.A of smaller graphite nodules (see table 2.1) (Hockley *et al.* 1999).

2. Plain journal bearing lining

   A total of 10 regions were selected randomly for identifying areas of microstructure containing crack initiation sites. The total number of cells was 2938. Only the Si secondary phase distribution is considered here as fatigue

crack initiation in this material was exclusively associated with the debonding of the Si phase from the surrounding matrix (Joyce 2001). Also, the Si phase is occasionally encapsulated by the Sn completely making the Sn distribution difficult to assess. The cells produced by the FBT were then divided into three populations :

i) Initiating Cells (163) - those cells containing a Si particle at which a fatigue crack initiated.

ii) Bordering Cells (810) - those cells sharing a common boundary with an initiating cell (i.e. near neighbours of the initiation particles).

iii) Background Cells (1965) - those cells containing a particle that is not sharing any boundary and showing no sign of fatigue crack initiation.

Results comparing the mean and standard deviation of the FBT features show that there is not much significant difference between the bordering and background cells (see table 2.2). As such, the bordering and the background cells were assumed to belong to the same class. The results further show that, as the O.A, C.A , d_{Mean} and L.A.F increase crack initiation is more likely to occur (Joyce 2001).

2.4 Particle Distribution Simulation

The ability to extract detailed geometrical information on a particle-by-particle basis, and then examine such measurement information across the whole microstructure, makes tessellation a uniquely powerful approach to assess both localised (crack initiation) and global (secondary processing effects on particle distribution) process behaviour. Previous work at Southampton (Yang *et al.* 2000, Yang *et al.* 2001) has been carried out to enhance understanding of the effect of random and clustered particle distributions. A variety of two-dimensional finite-size particle distributions have been simulated to achieve this. The approach taken can be generalised to other systems containing low aspect ratio finite bodies of low to moderate area fraction. The motivation of this work was that ductility and fracture toughness are seen to decrease with increasing inhomogeneity of the reinforcement distribution (i.e increased clustering) with many researchers indicating that particle clusters

may provide easy crack initiation and/or propagation sites. In order to compare the real and simulated particle distributions, it is essential to ensure that the area fraction of the particles are the same. To further address this issue, the simulation packages for particle distributions needs to be able to investigate the influence of particle morphology, random and clustered (i.e microstructural characteristics).

Several microstructure characteristics were defined for microstructure simulations in the previous work by (Boselli *et al.* 1999). The main parameters were the definition of shape, size and spatial distribution of the objects and object orientation which can be simulated using an in-house (Yang *et al.* 2000) Fortran program. Figure 2.7 shows an example of circular particles with a constant size that are :(a) randomly distributed and (b) clustered. The location of the object is generated using a two dimensional rectangular area with a specified nominal width and height. The percentage area fraction and the size of the object has to be determined prior to generating the distributions. This information enables specification of the number of objects to be simulated in the area under consideration. There are two stages involved in order to generate each microstructure characteristic: (1) generating the O.A and (2) generating the location of the centroids and ensuring that no overlapping occurs between the objects. The locations of the object centroids were generated with a normal distribution in their x and y coordinates. For the clustered distribution, a set of "parent" locations were generated based on a specified distance being maintained between them and then a normal distribution for a specified number of "children" particles were placed locally around each parent. This simulated particle distribution program may then be used to systematically vary the object shape, size, orientation and spatial distribution to visualise the model produced from our learning system. The full procedure and related results will be described in more detail in chapter 7.

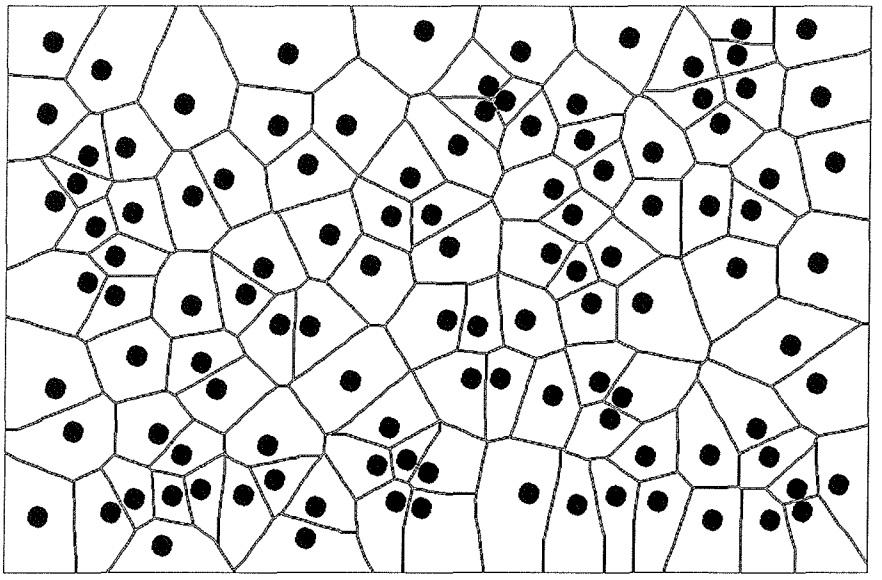
This simulated particle distribution program may then be used to systematically vary the object shape, size, orientation and spatial distribution to visualise the model produced from our learning system. The full procedure and related results will be described in more detail in chapter 7.

Description	Initiating		Background		Overall	
	Mean	SD	Mean	SD	Mean	SD
Object Area, (O.A) x_1 $(\mu m)^2$	2326.88	2549.83	476.45	890.87	549.88	1071.70
Object Aspect Ratio, ($O.A_r$) x_2	1.30	0.28	1.40	0.38	1.40	0.37
Object Angle, (O.Ang) x_3 (rad)	0.69	0.45	0.79	0.41	0.78	0.41
Cell Area, (C.A) x_4 $(\mu m)^2$	12340.84	7628.74	5761.52	4653.25	6022.60	4973.30
Local Area Fraction, (L.A.F) x_5	15.87	10.13	6.34	7.03	6.71	7.42
Number of Near Neighbours, (N.N.N) x_6	7.60	2.21	5.68	1.82	5.76	1.87
Nearest Neighbour Distance, (d_{min}) x_7 (μm)	16.23	16.36	17.40	17.04	17.36	17.02
Mean Nearest Neighbour Distance, (d_{mean}) x_8 (μm)	64.81	21.08	56.71	24.16	57.03	24.09
Nearest Neighbour Angle, (N.N.Ang) x_9 (rad)	0.77	0.47	0.75	0.46	0.75	0.40

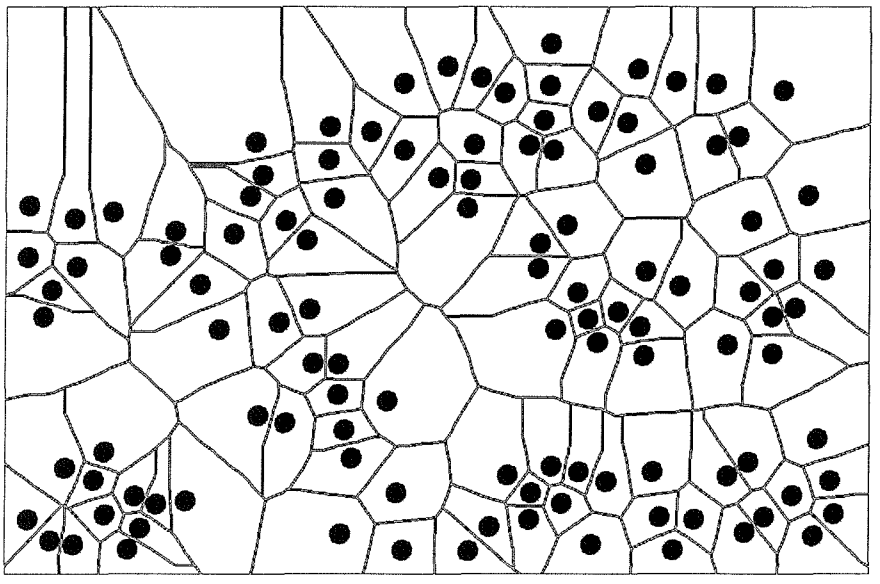
Table 2.1: Results obtained from the automotive camshaft ADI material. The mean and standard deviations (SD) of the FBT features between the "crack" (initiating), "no crack" (background) and their overall distribution are shown here. Units are in micrometers and radians where applicable.

Description	Initiating		Bordering		Background		Overall	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Object Area, (O.A) x_1 $(\mu m)^2$	12.17	11.54	6.22	6.73	4.24	5.19	5.23	6.44
Object Aspect Ratio, ($O.A_r$) x_2	1.49	0.36	1.50	0.41	1.48	0.50	1.49	0.47
Object Angle, (O.Ang) x_3 (rad)	0.90	0.41	0.88	0.42	0.87	0.4	0.88	0.41
Cell Area, (C.A) x_4 $(\mu m)^2$	113.65	58.59	87.59	53.45	62.46	43.21	72.23	49.49
Cell Aspect Ratio, ($C.A_r$) x_5	1.49	0.42	1.61	1.22	1.70	1.51	1.67	1.40
Cell Angle, (C.Ang) x_6 (rad)	0.88	0.42	0.79	0.44	0.77	0.44	0.78	0.45
Local Area Fraction, (L.A.F) x_7	10.48	6.41	6.98	5.34	6.93	5.73	7.15	5.72
Number of Near Neighbours, (N.N.N) x_8	6.39	1.38	6.00	1.45	5.52	1.42	5.70	1.45
Nearest Neighbour Distance, (d_{min}) x_9 (μm)	2.61	1.87	2.53	1.92	1.94	1.7	2.14	1.88
Mean Nearest Neighbour Distance, (d_{mean}) x_{10} (μm)	7.59	2.5	7.30	2.64	6.03	2.56	6.47	2.66
Nearest Neighbour Angle, (N.N.Ang) x_{11} (rad)	0.73	0.44	0.79	0.46	0.83	0.45	0.82	0.46

Table 2.2: Results obtained from the automotive plain journal bearing lining material. The mean and standard deviations (SD) of the FBT features between the "crack" (initiating), bordering, the "no crack" (background) and their overall distribution are shown here. Units are in micrometers and radians where applicable.



(a) Random distribution of the objects



(b) Clustered distribution of the objects

Figure 2.7: A sample of the simulated particles corresponding to a circular shape with (a) randomly and (b) clustered distributions. The particles are in black and the background is white.

2.5 Modelling in the Material Science Community

Artificial Neural Networks (ANN) have been notably successful in the field of material science in tackling the problems of regression and classification. Their applications, focus mainly on metal property/process predictions (i.e. regression problems), (Sumpter & Noid 1996, Bhadeshia 1999, Linkens & Yang 2001) have provided a good review of the wide application of ANN to the field of materials science. Early work on ANNs in material science has used Multi-Layered Perceptrons (MLP). MLPs are non-linear data driven models that have the advantage of finding the interrelationships between variables without having to specify their prior relationship, but they require a large set of training data to make the derived model robust. These requirements come from the goals set, which are usually based on probability density estimation. This requirement extends to many classical approaches for classification, such as nearest neighbourhood, linear discriminant and parametric models which were built based on the assumption that the data set is large. Using the probabilistic density estimation of the data will lead to the problem of the "Curse Of Dimensionality" (COD) (Bishop 1995). As the input dimension increases, the number of data required increases exponentially in order to provide the same consistent result. A Bayesian framework uses the prior to overcome the problem of COD. This is done by imposing some prior knowledge about the parameters of the model. Mackay and Bhadeshia have done substantial work on applying Bayesian frameworks in the field of materials science problems (Bhadeshia *et al.* 1995, Gavard *et al.* 1996, Fuji *et al.* 1996). The Bayesian framework in ANN is based on its ability to infer the model complexity from data. Furthermore, it incorporates error bars, which represent the uncertainty involved in the model prediction. Alternatively, the *ad hoc* decision of selecting the parameters in an MLP are made explicit by Gaussian Processes (GP), implementing a Gaussian prior over the function space, which the learning machine computes. (Bailer-Jones *et al.* 1997, Bailer-Jones *et al.* 1999) compare the use of Bayesian NN with a GP model in the prediction of deformed and annealed microstructures and also uses these approaches to model austenite formation in steel. They concluded that in their GP model, its hyperparameters

are more interpretable than the weights obtained in the Bayesian ANN (i.e the weights are explicitly parameterised). Furthermore, the prediction results gained from GP are superior to those of Bayesian ANN. A good review of Bayesian ANN and GP can be found in (MacKay 1991, Neal 1996).

ANN outperform classical linear pattern recognition techniques as the neural network is a non-linear model (Bhadeshia 1999). This non-linearity has its advantages and disadvantages. Bhadeshia claims that the parameters in ANN, such as the derived function and the associated coefficients (i.e. weights), can be revealed as relationships and interaction of the model features. (Plate 1999, Schooling *et al.* 1999) point out that the ANN can be considered as a "black box" as it is difficult to understand/visualise both the functions computed and the structure that is computed. Another difficulty involved in using ANN is the problem of overfitting (leading to poor generalization). Although this can be reduced using a cross-validation technique, it is both computationally expensive and requires a large set of data to be available. (Linkens & Yang 2001) highlighted that a more robust model can be produced with a "Grey box" modelling approach. Grey Box modelling is where some physical properties of the model can be incorporated into he model. A well known example of Grey box modelling is the neuro-fuzzy model. This combines fuzzy rules (physical understanding of the system) with an ANN (intelligent model). The interpretability of the model can be obtained through assessing linguistic fuzzy rules (Schooling *et al.* 1999). The advantages of Grey box modelling, as mentioned by (Linkens & Yang 2001) are its robustness, improvement of generalisation ability and reduction in dependence on the process data (i.e. a transparent model). There are many successful applications of both ANN and its extension, neuro-fuzzy networks, notably in the field of materials at both the material departments of Southampton University and Sheffield University, Institute for Microstructural and Mechanical Process Engineering (IMMPETUS).

Part of the theme of this thesis is to provide an interpretable model (i.e. Grey box modelling). Our focus here is on using the SUpport Parsimonious ANalysis Of VAriance (SUPANOVA). This approach obtains its parsimonious model (hence

interpretability) through decomposing its model structure (i.e. the kernel function is decomposed in the Support Vector Machine (SVM)). The SVM was developed based on Statistical Learning Theory (SLT) which was thought to be the best approach for modelling with small sample data set in accordance to (Cherkassky & Mulier 1998). The development of SVM and SUPANOVA will be described in more detail in chapter 3 and 5 respectively

2.6 Classical Classification Approach - Fisher Linear Discriminant

Prior to using complex models (e.g. SVM) for a classification problem, simple classical approaches should be attempted. From SLT, these simple approaches may indeed outperform the more complex approaches. (This is explained in more detail in chapter 3.1 and 3.5.2.) Therefore the next section describes a simple classification approach - The Fisher Linear Discriminant (FLD). Discrimininant functions are used to distinguish the differences between two or more groups of data or features in classification problems (Bishop 1995). A typical choice of a discrimination function is one which is linear in the input vector $\mathbf{x}$ and can be written in the form of:

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \tag{2.1}$$

where $\mathbf{w}$ is the weight vector b is the bias and T is the transpose. $\mathbf{w}$ can be optimally selected so as to accommodate class overlapping by maximising the class separation in which the decision boundary is given by $y(\mathbf{x}) = 0$. In a two class problem, an input vector $\mathbf{x}$ is assigned to class $+1$ if $y(\mathbf{x}) > 0$ and to class -1 if $y(\mathbf{x}) < 0$. Linear discriminant analysis provides the basis of generalisation to non-linear discrimination functions and other non-linear methods. Fisher linear discrimination (FLD) is a classical method for classification (Fukunaga 1990). It is used to reduce the increasing dimensionality of the input feature space by maximising the separation between class means while minimising the class variance direction. Fisher discrimination is "not strictly" a discriminant, but it can easily be used to construct a discriminant (Cover 1965) using the idea of the least squares approach.

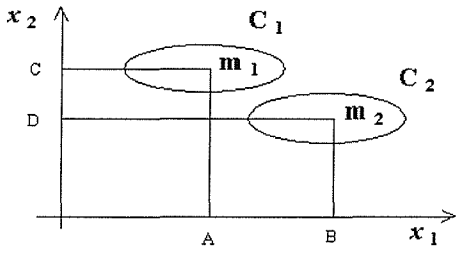


Figure 2.8: The necessity of considering within class covariance for Fisher Linear Discriminant. Projecting the mean of both classes along $\mathbf{x}_1$ will result in large separations and overlaps (A and B), compared to projecting the mean along $\mathbf{x}_2$ which will result in small separations and no overlaps (C and D).

For a two-class problem, the Fisher criterion is given by :

$$J(\mathbf{w}) = \frac{(p_1 - p_2)^2}{\sum_{k=1}^{2} \mathbf{S}_k^2} \tag{2.2}$$

where $p_k = \mathbf{w}^T \mathbf{m}_k$ is the class mean of the projected data for class k, $\mathbf{S}_k^2 = \sum_{\mathbf{x}^n \in C_k} (y(\mathbf{x}^n) - p_k)^2$ is the within class covariance for class k and $\mathbf{m}$ is the mean vector class for k class. It can be seen that the necessity to measure the within class covariance here as the larger separation between the means implies good separation as there is a tradeoff between larger separation and overlapping between classes (see Fig. 2.8). Maximisation of the Fisher criterion results in maximisation of the separation of the projected class mean and the minimisation of total within class variance. Therefore, the Fisher criterion maximises a function representing the projection of the class means and hence class separation. There exists a closed form solution to the weight maximisation as :

$$\mathbf{w} \propto \mathbf{S}_w^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{2.3}$$

where $\mathbf{S}_w$ is the total within class covariance matrix given by :

$$\mathbf{S}_w = \sum_{\mathbf{x}^n \in C_1} (\mathbf{x}^n - \mathbf{m}_1)(\mathbf{x}^n - \mathbf{m}_1)^T + \sum_{\mathbf{x}^n \in C_2} (\mathbf{x}^n - \mathbf{m}_2)(\mathbf{x}^n - \mathbf{m}_2)^T \tag{2.4}$$

The weight vector $\mathbf{w}$ is a specific choice of direction for the projection of data onto a one-dimensional space (i.e. finding the direction of the weighting vector $\mathbf{w}$)

(Rencher 1998). The weight direction is the main concern here and hence it is a descriptive model of the training data.

In order to construct a discriminant function as in Eq. (2.1), a least squares approach can be used to obtain the bias b. The basic idea of least squares is to minimise the Fisher criterion J of Eq. (2.2) with respect to $\mathbf{w}$. The bias b can be expressed as :

$$b = -\mathbf{w}^T\mathbf{m} \tag{2.5}$$

where $\mathbf{m}$ is the mean of all the patterns k. Now the discriminant function is similar to the one in a linear discriminant. In comparing the least squares and Fisher approach, least-squares makes the output space as close as possible to the target output while Fisher maximises the class separation in the output space. Although the approaches in the two methods are different, the resulting weights for the FLD coincides with the least-squares approach.

The solution to the FLD is equivalent to that of the Bayes linear classifier (i.e. $P(y|\mathbf{x}) = p(\mathbf{x}|y)P(y)$) when the class conditional density (i.e. $p(\mathbf{x}|y)$) is assumed to be a multi-dimensional Gaussian distribution with equal covariance matrix and prior of its target (i.e. $P(y)$). At first sight, the original FLD seems not to suffer the "curse of dimensionality" that was due to the class conditional density estimation. It becomes clear when we consider that the mean vector $\mathbf{m}$ and the class covariance matrix S_w from the FLD are taken to be the maximum likelihood estimation corresponding to the mean and class covariance of the Bayes approach. Therefore, the classification approach to solve classification problems still requires the density estimation of its data. Furthermore, the FLD is a *descriptive* rather than a *predictive* model. The main difference between these two models is that the descriptive model provides a description of the data while the predictive model (e.g. Support Vector Machines (SVM), provides predictions about the data. The FLD uses empirical data to describe the models and also focuses on getting the right values for weights only, rather than the loss incurred from the decision function. The use of Empirical Risk Minimisation (ERM) alone is only justifiable when we

have a large set of data or the underlying data distribution is known. However, in practice, the data is limited and using ERM will not in general reflect the true distribution. This briefly describes why the classical approach is not appropriate for classification.

Another approach to deal with large input dimensionality is to investigate the effect of limited data. Statistical Learning Theory (SLT) was developed based on small sample data sets in which it takes into account the capacity of the class function which is better known as the Vapnik-Chervonenkis (VC) dimension (Vapnik 1995). From the view point of the VC dimension, if the mapping from the input space to the feature space allows small training error and low capacity, then good generalisation is guaranteed. Model interpretability was developed based on the SLT framework based on model structure decomposition. Work done by several authors at Southampton (Kandola *et al.* 1999, Christensen *et al.* 2001) uses this approach for regression tasks for application to processing-property relationships in aluminum alloys.

2.7 Summary

The importance of fatigue crack initiation has been highlighted. The FBT provides a set of features that captures the distribution of the secondary phase and the morphology of the particle which may cause fatigue crack initiation. Further visualisation of the model may be achieved by using simulated particle distributions. This work focuses on the automotive industry looking into two components and material combinations, namely, the camshaft with ADI and plain journal bearing lining with Al-Si-Sn. The advantages and disadvantages of ANN modelling approaches in materials science have been described, bringing out the issue of data with limited samples and model interpretability. The theoretical basis of classical approaches to classification have also been outlined using the FLD.

Chapter 3

Learning From Data

Real world data sets have a restricted amount of data. Statistical learning theory (SLT) is perhaps the best currently available theory for finite sample statistical estimation and predictive learning (Cherkassky & Mulier 1998). This chapter provides a basic understanding of SLT and its conversion of a learning problem with a limited number of data samples into a function approximation estimator known as machine learning. The main requirements for setting up machine learning are then described in section 3.3 to 3.7. Understanding of Structural Risk Miminisation (SRM) for implementation in pattern recognition requires understanding of both the underlying conceptual and technical implications. These two requirements are not consistent but have led to the assumption that an accurate probability density estimation of the data provides a good classification model. This is assumed in the classical approach of solving this problem, indicating that the classical approach is inappropriate for small sample data set learning. As such, the Support Vector Machine (SVM) was then constructed from the framework of the SLT. The key success of SVM lies in using the SRM and kernel methods. The relationship between the SVM and the Regularisation Network (RN) are then highlighted in section 3.7.3, both use the kernel method to handle the problem of dimensionality transformation but with different ways to determine their associated parameters. The application of the SVM to tasks such as multi-class classification and regression then follows. The need to provide good generalisation is then discussed in section 3.9, describing how the parameters in SVM can be tuned.

3.1 Classical Statistical Classification Approach

Pattern recognition problems can be defined as follows: given an input point $\mathbf{x} \in \Re^N$, a class decision is made by determining which region the point lies in and providing an index for the region as the decision output, y. Classical formulation of classification problems is based on statistical decision theory. The simplest example is the construction of the optimal decision rule using prior probability of the target assuming that this is known (i.e. simply assign the output label to the class with the largest probability)

$$\begin{aligned} y(\mathbf{x}) &= 0 \text{ if } P(y=0) > P(y=1) \\ &= 1 \text{ otherwise} \end{aligned} \tag{3.1}$$

Upon observing the input, $\mathbf{x}$ and making the decision thereafter, provides more information about the decision region. As such, the decision can be made based on the posterior probability of $\mathbf{x}$ (i.e. replace $P(y=0) > P(y=1)$by$P(y=0|\mathbf{x}) > P(y=1|\mathbf{x})$). Expressing the posterior probabilities via Bayes theorem :

$$\begin{aligned} P(y=0|\mathbf{x}) &= \frac{p(\mathbf{x}|y=0)P(y=0)}{p(\mathbf{x})} \\ P(y=1|\mathbf{x}) &= \frac{p(\mathbf{x}|y=1)P(y=1)}{p(\mathbf{x})} \end{aligned} \tag{3.2}$$

where $P(\mathbf{x}|y)$ is the probability density/likelihood estimation of the data $\mathbf{x}$ for a given class y. The accuracy of this function was then used to minimise the misclassification error in the classical approach. We will see in the next chapter that for imbalanced data a misclassification cost has to be incorporated into each class and the drift of the target needs to be incorporated as well in order to make a better prediction.

The above sets up the basis of the classical pattern recognition problem associated with Statistical Learning Theory. Clearly, in order to solve the posterior probability, one would be required to solve the probability density of estimation of $\mathbf{x}$ (i.e $p(\mathbf{x}|y)$). Much of the classical approach to pattern recognition such as Fisher Linear Discriminant and Artificial Neural Networks all require a good estimation

on the probabilistic density or underlying distribution on its data. An accurate estimation of the probabilistic density requires a large set of data with respect to its input dimension. Given that we typically have insufficient data in real world data sets to generate an adequate density function, this approach to solving pattern recognition problems is limited. In fact, according to (Friedman 1997), the more commonly used loss functions for classical pattern recognition such as squared error and entropy use the concept of density estimation. In such instances, the goal for classification is (incorrectly) interpreted as posterior probability estimation (i.e the $P(y|\mathbf{x}) = p(\mathbf{x}|y)P(y)$). Friedman observed that accurate estimation of the posterior probability is not necessary for accurate classification. The explanation of his observation can be derived from the statistical learning point of view (see section 3.5.2). As such, the estimation of the density function should be made redundant and one should solve the classification directly, this leads to the motivation of a learning machine the Support Vector Machine (SVM) which was developed based on Statistical Learning Theory (SLT).

3.2 Statistical Learning Theory

(Vapnik 1995) suggests that when solving problems with limited data, *"Do not attempt to solve a specified problem by indirectly solving the harder general problem as an intermediate step"*.

The intermediate step referred to is the probability density estimation step described above, the complexity of which is higher than the desired classification problem. The classical approach uses empirical risk minimisation (ERM) indirectly to estimate the densities, which are then used to formulate the decision rule. Under SLT, the goal is to find a decision boundary minimising the expected risk. This is based on the concept of Structural Risk Minimisation (SRM) that will be described in more detail in the next section.

The classical approach to learning requires either the underlying distribution of the data to be known or that the data set is large in order to obtain good probabilistic density estimation to its input. This however is not the case for most practical applications. Learning from another point of view can be established from the structure

of the data given. Such learning from data requires us to build a model from an insufficient set of information (usually small sample data sets) in order to attain some underlying structure of the (unknown) process and using this to achieve good prediction performance. Given that the data set is small and its representation of this underlying structure and also the model usually are in numerical form, from a statistical perspective this can be cast as the problem of *Function Approximation.* This setting is equivalent to using learning for multivariate function approximation from limited data, which is an *ill-posed* problem. A problem is well posed when a solution exists, is unique and depends continuously with the data set. It is ill-posed when it fails to satisfy at least one of these criteria.

Statistical learning theory effectively describes statistical estimation for small data samples (Cherkassky & Mulier 1998). The proponents of SLT are set using the probabilistic dependency between the (input,output) to form a class of function restricted by the amount of data given to handle the ill-posed problem. Then, in a statistical learning framework, learning is an estimation of a class function :

$$y = f(\mathbf{x}, \alpha) \tag{3.3}$$

The class of function is determined by its parameter α. α is used to describe how the output y is obtained and needs to be determined. The risk of obtaining this parameter is associated with a loss function and the joint probability density function. This is the general setting of SLT and it provides a wide range of possibilities in learning which will be elaborated later in this chapter. A robust model for constructing SLT is the powerful learning machine which is a learning algorithm that can provide accurate function approximation with good generalisation by bounding the risk associated with the parameter.

3.3 Learning Machines

The properties that we would like the learning machine to have are : 1.) a good estimator of the unknown function (i.e. estimating an unknown dependency from known observation), 2.) it must be computationally efficient - to solve the problem with a reasonable computation time and 3.) to guarantee good generalisation ability

- to deal with problems of predictive learning (using the estimated dependency to predict new unseen data). There are four important components that comprise this machine : 1) the definition of the learning task (learning associated with a loss function) , 2) an induction principle , 3) a set of decision functions and 4) an algorithm to implement the previous 3 components. The following section describes each component in more detail.

3.4 Loss Function And Risk Minimisation

The capacity of a set of functions, to which the solution belongs, lies in hypothesis space and is given as $f(\mathbf{x}, \alpha)$, where $\alpha \in \Lambda$ and Λ is any abstract set of parameters. Given the hypothesis space, the best estimation of the function $f(\mathbf{x}, \alpha)$ for which the risk function associated with $\mathbf{x}$ is minimised is given as :

$$R(\alpha) = \int \mathcal{L}(y, f(\mathbf{x}, \alpha))p(\mathbf{x}, y)d\mathbf{x}dy \tag{3.4}$$

Where $\mathcal{L}(y, f(\mathbf{x}, \alpha))$ is the loss function, that measures the difference between the actual value, y and its estimates from the learning, $f(\mathbf{x}, \alpha)$ given by the unknown structure of a point $\mathbf{x}$ associated with its parameter α. The $p(\mathbf{x}, y)$ defines the joint probability density function (PDF) since we typically do not know about the PDF, it is possible to find an approximation according to minimising the expected average loss. This will be described in the next section. The common three learning problems are classification, regression and density estimation, each requiring an appropriate loss function in order to minimise its respective expected loss function. The definitions of each problem are briefly described as follows :

Classification For a two class pattern recognition problem, each vector x is labeled by an output $y \in 0$ or 1 in Eq. 3.3. The corresponding loss function is an indicator function that measures the classification errors, given as :

$$\begin{aligned} \mathcal{L}(y, f(\mathbf{x}, \alpha)) &= 0 \quad \text{if } y = f(\mathbf{x}, \alpha) \\ &= 1 \quad \text{otherwise} \end{aligned} \tag{3.5}$$

If the estimation of the unknown function ($f(\mathbf{x}, \alpha)$) is the same as the labeled, y, there is no penalty imposed. Otherwise, a misclassification cost is involved.

Regression Estimation The label of the output in this task is a real number (i.e. $y \in \Re$) and usually, it is assumed that it consists of a sum of deterministic function ($g(\mathbf{x}) = \int yp(y|\mathbf{x})dy$) and a random error (noise) with zero mean (i.e. $y = g(\mathbf{x}) + \epsilon$) A metric representation is usually used as a tool to get the estimate closer to the unknown estimated functions. A common loss function for regression is the squared error :

$$\mathcal{L}(y, f(\mathbf{x}, \alpha)) = (y - f(\mathbf{x}, \alpha))^2 \tag{3.6}$$

Density Estimation The density estimation of a input vector $\mathbf{x}$, has no output y involved. If the unknown input vector belongs to the set of joint probabilistic densities $P(\mathbf{x}, \alpha), \alpha \in \Lambda$, the loss function of the density estimation can be written as :

$$\mathcal{L}(\mathbf{x}, P(\mathbf{x}, \alpha)) = -\log P(\mathbf{x}, \alpha) \tag{3.7}$$

3.5 Induction Principle

Given a limited data set, estimating the optimal function of $f(\mathbf{x}, \alpha)$ exactly is not possible. The approximated optimal function is found by using the induction principle. The induction principle allows $f(\mathbf{x}, \alpha)$ of the "true dependency" to be found in the class of estimation function with limited data. The simplest induction principle is the so-called Empirical Risk Minimisation (ERM) principle and in statistical learning is known as the Structural Risk Minimisation (SRM). The SRM sets the basic framework for the learning machine known as a Support Vector Machine. The learning machine then uses a constructive implementation of the induction principles (Vapnik 1995).

3.5.1 Empirical Risk Minimisation (ERM)

Minimising the risk function in Eq. 3.4 requires the joint PDF of finite data to be known, but usually this is unknown. One can approximate it using the empirical risk function :

$$R_{emp}(\alpha) = 1/\ell \sum_{i=1}^{\ell} \mathcal{L}(y_i, f(\mathbf{x}, \alpha)) \tag{3.8}$$

where $\alpha \in \Lambda$ and Λ is any abstract set of parameters and ℓ is the number of data. The selection of the decision rule is based on its empirical performance on the finite number of training samples. This induction principle is known as the Empirical Risk Minimisation (ERM). Using the ERM as an approximation to minimising the risk function especially when ℓ is small is inappropriate. (Vapnik 1995) showed that for the case of pattern recognition, there exists a bound on the expected risk that holds with probability $1 - \eta$:

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(\log(\frac{2\ell}{h}) + 1) - \frac{\eta}{4}}{\ell}} \tag{3.9}$$

Where h is the Vapnik-Chervonenkis (VC) dimension of the set of decision functions parameterised by α, and (typically $\eta = \min(\frac{4}{\sqrt{\ell}}, 1)$. The VC dimension of a set of decision functions is the maximum number of points that can be separated in all possible ways by that set of decision functions. For a known value of h, our goal is to make the bound as small as possible so that the best choice can be calculated. From this existing bound, the use of ERM is justified only if a large data sample is provided. That is, if the ratio of ℓ/h is large, then the confidence interval (second term in 3.9) approaches 0, then the ERM is close to the expected risk. However, if ℓ/h is small, then both the terms need to be minimised. To minimise both terms, however, requires the VC dimension of a set of decision functions to be a control variable and at the same time should generate a simple model rather than a complex model (relating to Occam's Razor Principle [1].)

[1] William of Occam (1285-1349) :"Causes Should Not be multiplied beyond necessity"

There are two ways to solve this minimisation of the bounded problem (Cherkassky & Mulier 1998),

1) Keep the confidence interval fixed and minimise the empirical risk. The model structure is controlled by the number of basis functions and for a given number of basis functions the empirical risk is minimised using numerical optimisation. For a given number of data samples there is an optimal structure providing the smallest estimate of the expected risk. An example of this principle is used in the Radial Basis Function (RBF) network commonly used in artificial neural networks (ANN) and the regularisation network (RN).

2) Keep the empirical risk fixed and minimise the confidence interval. A special structure (i.e. structural risk minimisation (SRM)) is required to ensure that the empirical risk is small for all approximation functions. Under this, the best value from the structure is that which minimises the value of the confidence interval. An example is the Support Vector Machine (SVM). In accordance with (Evgeniou *et al.* 1999), the RN and the SVM are very similar in their properties except the way in which their bounds are minimised. This will be described in more detail in section 3.7.3.

As mentioned in the induction principle section, the optimal decision function that is selected might not reflect the true unknown function. Therefore, generalisation is used to control the set of functions $f(\mathbf{x}, \alpha)$. The capacity of this set of functions (also known as hypothesis space) controls the empirical risk achieved. A large hypothesis space will produce low empirical risk but poor generalisation. On the other hand, a small hypothesis space will produce good generalisation but will not be able to describe the data variable dependency in the data. The characteristic of the derived model generalisation ability typically has a bowl shape with respect to the capacity of the set of functions. As such, it can be controlled by either choosing the appropriate VC-dimension or some other embodiment of capacity in the set of functions.

3.5.2 Structural Risk Minimisation (SRM)

The SRM induction is an induction principle based on statistical learning. It provides a formal mechanism for choosing an optimal model complexity for limited data. Implementation of the SRM principle depends on two concepts : the set of approximation functions has to be a nested structure ordered according to complexity (VC dimension) and the expected risk, where the sum of the empirical risk and the confidence interval is minimised. Rigorous estimation of the prediction risk is difficult since it is difficult to estimate the VC-dimension for non-linear functions. This requires separation between complexity and dimensionality using a decision function of a linear form and kernel methods (see section 3.7.2).

The confidence bound in Eq.3.9 justified the use of SRM principle. It attempts to control both empirical risk on the training data and the capacity of the set of decision functions to obtain the expected risk. The structure is defined as :

$$S_1 \subset S_2 \subset .... \subset S_n \tag{3.10}$$

Where the set of decision functions $S = f(\mathbf{x}, \alpha), \alpha \in \Lambda$ and it ranks according to their complexity as the subscribed n increases. Hence satisfying a VC dimension that are:

$$h_1 \leq h_2 \leq ..... \leq h_n \tag{3.11}$$

From here, the appropriate structure is selected that minimises the bound in Eq. 3.9. As such, the SRM principle defines a tradeoff between the accuracy or fitting and complexity of the approximation based on a set of data given.

Prior to implementing SRM, it is important to notice two important issues regarding its conceptual and technical implication. This important issue is extended to learning associated with the classification task :

- The misclassification error empirical risk is binary. Hence, conceptually minimising it will lead to a combinatorial optimisation problem. For technical implementation, a continuous optimisation is only used to approximate the misclassification error.

- The estimation of the prediction performance is known as model selection. The conceptual and technical implementation uses the direct misclassification error.

The requirement for the distinction between the conceptual and technical implementation leads to two different costs involved in practical implementation of SRM. Firstly the empirical risk uses a continuous loss function (while conceptually it uses the misclassification error - via continuous nonlinear optimisation). Secondly, the estimation of the expected risk uses the misclassification error (model selection) (Cherkassky & Mulier 1998). This unfortunately was not obvious in the classical approach to classification problems, leading to the interpretation of the goal of classification problems as the probabilistic density estimation described in section 3.1. Understanding the different basis between classical (focused on the probabilistic density estimation of the data) and statistical learning (SRM) focused on approaches to pattern recognition, explains why some simple methods such as nearest neighbour distance, or linear discriminant sometimes outperform sophisticated non-linear methods such as ANN (Cherkassky & Mulier 1998). For example,

- Simple classification methods such as nearest neighbour may not require a non-linear optimisation solution (the empirical risk is minimised directly)
- If the simple methods provide the same level of empirical misclassification in the minimisation stage as the more complex model, then the use of the complex model may not provide a better performance as the empirical risk estimation is an approximation of the misclassification error.
- The sensitivity of classification is less than regression, as slight changes (provided by the non-linear model based on the practical implementation) may not have much impact on the misclassification error.

Therefore, SLT provides a good justification for the success of some classical approaches as compared to non-linear modelling. The problem of classification is simpler than regression and hence should be solved prior to developing regression approaches.

3.6 The Decision Functions

The output from the learning machine is a set of linear functions defined as :

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{3.12}$$

where $\mathbf{x}_i$, $i = 1, ..., \ell$ are input vectors in $\Re^N$, b is a scalar and α is a parameter. The kernel Matrix $K(\mathbf{x}, \mathbf{x}_i)$ is symmetric (the kernel function will be described in more detail in section 3.7.2). A linear function can be more accurately estimated than non-linear functions. However, sometimes a linear function is not flexible enough to represent the function, hence a non-linear function is required. In ANN non-linear functions are used directly over the input space and then the classification is done in the input space. This does not provide a unique solution to the problem because the complexity constraint is not defined clearly (i.e. how many basis functions are good enough for the estimate). We will see next that the SVM uses these linear functions with constraint on complexity to form the set of decision functions as above using the SRM principle.

3.7 Support Vector Machine (SVM)

Many practical applications have a set of data that are insufficient for drawing accurate inferences. Limited data may lead to selecting a model that is too simple (as a consequence of insufficient data). This implies that the data set is too small to identify any complex model with certainty. Note: this is different from Occam's Principle for selecting a simpler model or imposing a simpler prior model. Rather, the prediction performance of this simpler model has to be questioned. We accept that our model might be partially wrong. A question is now raised how much we can reliably infer from the data if given a statistical model? Support Vector Machine (SVM) is a learning machine that was developed based on the SLT (Cortes & Vapnik 1995). SVM comprises of two important features, namely, the use of SRM and kernel methods. The next section will describe in more detail these two features. This is then followed by a brief review of the regularisation network (RN) (i.e. it

can be considered as a kernel learning machine too) and how these are related to SVM.

3.7.1 Construction of SVM

The SVM was developed by (Cortes & Vapnik 1995, Boser *et al.* 1992) who implemented the SRM principle into a learning machine. In SLT, the classification problem is conceptually inherently less complex than the regression estimation problem. Therefore, a classification problem should be used directly instead of estimating the probabilistic density function of the data. However, the technical implementation is not that straightforward as described in section 3.5.2. This leads to a different goal for the classical approach to a classification problem and the SLT approach. The goal set for the classical approach requires estimation of the probability density of the data, which may not imply a good classification rate to be obtained. On the other hand, the goal for a SLT is to find the decision boundary minimising the expected risk (Eq. 3.9). SVMs were therefore developed based on their conceptual simplicity (i.e. developed for a classification problem and then extended to regression problems).

In classification, a hyperplane (usually a linear function) capable of separating the training data without error is used. Given the training data consisting of l samples $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_\ell, y_\ell)$, $\mathbf{x} \in \Re^N, y \in \pm 1$ can be separated by the hyperplane decision function :

$$f(\mathbf{x}) = (\mathbf{w}^T \mathbf{x}) + b \tag{3.13}$$

where $\mathbf{w}$ are the weight vector coefficients and b the bias. This defines a general hyperplane and there exists many possible solutions. In order to fix the misclassification error from the empirical risk to be as small as possible (this is the second idea of how to minimise the bound described in section 3.5.1), it is important that all the possible hyperplanes can be represented in the form of Eq.3.13. In order to implement the SRM principle with the hyperplane, the VC dimension (which measures the capacity of a set of functions) must be bounded. In accordance with (Vapnik 1995), the bounded VC dimension, h of the set of canonical hyperplanes

in N dimensional space is :

$$h \leq min[R^2A^2, N] + 1 \tag{3.14}$$

where R is the radius of a hypersphere enclosing all the data points and A is the bound of weighted coefficients (i.e. $\| \mathbf{w} \| \leq A$). This effectively controls the capacity of the function by reducing the number of possible hyperplanes. Notice the notion of complexity and dimensionality are separated. The complexity is controlled by the bound and it is independent of the dimensionality. In the classical approach such as a ANN, the complexity is controlled by the number of basis functions and it is dependent on the input dimensionality. The dimensionality is handled by the kernels method described next. The output of the decision functions given in section 3.6 is of linear form and in SVM, it is subjected to the constraint of the canonical hyperplane. In some practical applications, linear decision boundaries might be inappropriate, and non-linear decisions are applied. The non-linear decision in SVM uses the elegance of the kernel methods (denoted by $K(\mathbf{x}, \mathbf{x}')$ to transform the input vector, $\mathbf{x}$ into a high dimensional feature $\mathcal{F}$ (via prior non-linear mapping) and then to construct the optimal hyperplane.

3.7.2 Feature Space and Kernel Functions

Linear models provide little flexibility to our model as they only use the linear dependencies between the data. An example of this is the Fisher linear discriminant (FLD) described in chapter 2.6, which assumes the distribution of the data to be Gaussian, given the covariance between both classes is the same. This model can be useful if prior knowledge about the problem is good and the estimate of the parameters can be accurately obtained. On the other hand, non-linear models such as ANN have been successfully used (as reported in the material science literature (see chapter 2.5)). Their success is due to the flexibility of their structure in being able to adapt to a wide range of functions, hence allowing a non-linear model. (Bishop 1995) view ANN as a framework for transforming a non-linear functional input to a set of output variables. The input vector $\mathbf{x}$ in the space $\Re^N$ is mapped non-linearly by a function into the feature space $\mathcal{F}$. The learning then proceeds

to use this $\mathcal{F}$ space rather than the original input space. It is believed that by transforming the data to a high dimensional space (i.e. $\mathcal{F}$ space), the data can be separated more easily (Cover 1965). However, working in $\mathcal{F}$ space requires the ability to control its complexity (in a basis function model this means controlling the number of basis functions) leading to the same problem of dimensionality. Kernel methods are different from the ANN approach as there is no restriction placed on the number of basis functions used to construct the high dimensional mapping of the input variables. Work by (Smola 1998) shows that the kernels correspond to regularisation operators which can be used to provide a smooth mapping, hence providing a good generalisation. This will be described in more detail in the next section.

The solution to the above problem of complexity versus dimensionality is separated in SVM. The complexity is bounded by the SRM principle and the dimensionality is managed via the kernel methods which map the input vector to the $\mathcal{F}$ space using only its dot products $\mathbf{x}^T\mathbf{x}'$. This eliminates the need to calculate the mapping into a $\mathcal{F}$ space directly which will then run into the dimensionality problems.

Candidates for the kernel functions have to satisfy Mercer's theorem (Mercer 1909). Mercer's theorem provides the condition for a valid Kernel to be used (i.e. it must be positive definite).

This allows for the mapping of the dot product of the input vector, $\mathbf{x}$ to the $\mathcal{F}$ vector (i.e. $\mathbf{x}^T\mathbf{x}' \rightarrow \phi(\mathbf{x})^T\phi(\mathbf{x}')$) which can be done implicitly through the selection of the kernel function as :

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}'). \tag{3.15}$$

where $\phi(\mathbf{x})$ is the basis function. This transforms the inner product of the input vector to a high dimensional $\mathcal{F}$ space known as a Hilbert space. A Hilbert space $\mathcal{H}$, is defined as a complete inner product space where the completeness is due to the metric defined by the inner product and it can be thought of as an extension of $\Re^N$ with a linear transformation to an infinite dimensional space. An example of $\mathcal{H}$ space is the well known Euclidean space. These kernels are then readily substituted

into the SVM for both classification and regression problems.

Here are some of the commonly used kernel functions :

Polynomial Kernel - with p degree of freedom

$$K(\mathbf{x}, \mathbf{x}^{'}) = (\mathbf{x}^T\mathbf{x}^{'} + 1)^p \tag{3.16}$$

Radial Basis Function (RBF) Kernel - with σ width

$$K(\mathbf{x}, \mathbf{x}^{'}) = \exp(-\frac{|\mathbf{x} - \mathbf{x}^{'}|^2}{\sigma^2}) \tag{3.17}$$

Spline Kernel - given an order of m and b nodes in a 1 dimension input, its inner product kernel is :

$$K(x, x^{'}) = \sum_{j=0}^{m}(xx^{'})^j + \sum_{k=1}^{b}(x - t_k)_+^m(x^{'} - t_k)_+^m \tag{3.18}$$

where $(x - t)_+ = max((x - t), 0)$ and $t_1, ..., t_b \in [0, 1]$. For a linear spline, (Smola 1998) show that with the order of $m = 1$ and an infinite number of nodes, the kernel is :

$$K(x, x^{'}) = 1 + xx^{'} + xx^{'}\min(x, x^{'} - \frac{(x, x^{'})}{2}(\min(x, x^{'}))^2 + \frac{(\min(x, x^{'}))^3}{3} \tag{3.19}$$

With a N dimensional splines, the solution for a linear spline is the product of the N one dimensional splines.

Mercer's theorem only provides information on which kernels can be used but it does not provide us with information as to which kernel is best. (Vapnik 1995) views the choice of kernel as equivalent to choosing features, $\phi(\mathbf{x}_i)$, related to the original inputs. He observed that the upper bound on the VC dimension is a potential avenue to provide a means for comparing the kernels. This approach is widely used

for feature selection (Chapelle & Vapnik 1999, Weston *et al.* 2000). It is important to realise that even though a strong theoretical method might exist for selecting the best kernel, it still requires an independent test to be used for kernel selection. Finally, a good reference to kernel methods can be found in (Vapnik 1995).

3.7.3 SRM in Regularisation Networks and SVMs

Work by (Evgeniou *et al.* 1999), provides a unified view of kernel learning approaches with Regularisation Networks (RN) based on SLT, showing that the bound used for the SVM is equivalent to that of RN. There are several notable learning algorithms such as Radial Basis Function (RBF) networks, Gaussian Processes (GP) and SVM that use kernel methods. Their main differences are how they attempt to minimise the bound as described in section 3.5.1. Also, their approach to optimising the associated parameters is different. For example using: the least square estimation (RBF network), the duality representation (SVM), and the Most Probable (MP) for GP. These correspond to parameterisation of the basis function and the kernel representation. This section provides a brief description of how the regularisation network and the SVM are related.

The RN attempts to penalise a model's parameters and structure by avoiding overfitting of data and restoring the well-posed condition for learning. Regularisation uses prior knowledge about the desired function to make the problem a well-posed one. The commonly used form of the priori is the "smoothness" of the function parameters (e.g. in Eq. 3.3 the α). The smoothness is defined as lack of oscillation behaviour of the function (e.g. two similar inputs will correspond to two similar outputs if the function is smooth) compared with the possible function behaviour in local neighbours of input space. The accuracy of the function estimation depends on having enough samples within the neighbourhood to specify the smoothness constraint. This is then inherent to the problem of dimensionality because as the dimensionality increases the number of samples must increase exponentially to give consistent results. This could be offset by increasing the number of data samples falling within the neighbourhood but this is at the expense of imposing stronger constraint. The standard minimisation of the loss function for the learning machine

is :

$$R_{emp}(\alpha) = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathcal{L}(y_i, f(\mathbf{x}, \alpha)) + \lambda Q(\alpha) \tag{3.20}$$

Where λ is a regularisation parameter which controls the tradeoff between the smoothness of approximation and accuracy of the approximation and $Q(\alpha)$ is the regularisation function that provide smoothness/constraints to $f(\mathbf{x}, \alpha)$. Notice that this is a means to minimise the bound described in section 3.5.1 (i.e. keep the confidence interval fixed and minimise the empirical risk). Given a sequence of positive numbers, the term $Q(\alpha)$ is some function that reflects the capacity of the function $(f(\mathbf{x}, \alpha))$ and will monotonically increase. It is worth mentioning here that there are other regularisation functions that exist such as the squared norm of α; $Q(\alpha) = \| \alpha \|_2^2$ (known as ridge regression) and constraining the model to stay in a small subset of possible models (i.e $\min \sum_{i=1}^{\ell} L(y_i, f(\mathbf{x}, \alpha)$ subjected to $Q(\alpha) < \frac{1}{\lambda}$) (Bellman 1961). It should be noted that $Q(\alpha)$ lies in hypothesis space.

Classical regularisation network theory lacks practical justification when applied to a finite set of data. (Vapnik 1995) justifies the use of regularisation techniques for finite data by considering the approximation function $(f(\mathbf{x}, \alpha))$ for a finite set of data. The function has to be constrained to an appropriately "small" hypothesis space. If the hypothesis space is large, model fitting is good but generalisation performance is poor. This concept is then formulated by Vapnik into the terms of the capacity of a set of functions depending on the training set size. For a small data set, the capacity of the function space is small, whereas it becomes large for a larger training set.

Let us summarise how the RN and SVM can be related before proceeding to describe the implementation of SVM. For the case of the RN, the $Q(\alpha)$ is fixed and λ is unknown. On the other hand for a SVM, the $Q(\alpha)$ is unknown and λ is fixed. In order to implement the SRM bound, the $Q(\alpha)$ must be a fixed prior. This is equivalent to fixing the weighted coefficients $\mathbf{w}$ as described in Eq. 3.14.

3.8 SVM For Pattern Recognition

For a two class separable pattern recognition case, the optimal hyperplane is defined as a hyperplane associated with ($\mathbf{w}$, b) in the feature space that maximises the margin from the closest point without training error. The margin is defined as the minimum distance from the hyperplane to the closest point. The optimal hyperplane is then obtained by maximising the margin given by (Vapnik 1995) :

$$\tau(\mathbf{w}, b) = \frac{2}{\| \mathbf{w} \|} \tag{3.21}$$

Maximising the margin requires minimising $\mathbf{w}$. Minimising this $\mathbf{w}$ is equivalent to implementing a SRM. It reduces the number of possible hyperplanes while minimising the bound on the VC dimension. In order to maximise the margin, minimising a quadratic cost function in $\mathbf{w}$ is appropriate :

$$\phi(\mathbf{w}) = \frac{1}{2} \| \mathbf{w} \|^2 . \tag{3.22}$$

In this way, the optimal weight $\mathbf{w}_{opt}$ obtained will provide the maximum separation between two classes and hence the optimal hyperplane is *unique*. Given a training vector as $\mathbf{x}_i, i = 1, ..., \ell$ with corresponding target $y_i \in \{-1, 1\}$, combining the linear discriminant function of the two different classes can be translated into :

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i = 1, ..., \ell. \tag{3.23}$$

Now, the cost function is a quadratic function and the constraints are linear with respect to $\mathbf{w}$. This constrained optimisation problem is called the *primal problem* and it can be solved using a Lagrange function, given as:

$$\ell(\mathbf{w}, b, \alpha) = \frac{1}{2} \| \mathbf{w} \|^2 - \sum_{i=1}^{\ell} \alpha_i (y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1) \tag{3.24}$$

where α_i are the Lagrange multipliers. The solution to this constrained optimisation problem is the *saddle point* for the function which needs to be minimised with respect to $\mathbf{w}$ and b and maximised with respect to α. In order to obtain an optimal solution for the Lagrange multipliers, the primal problem is transformed to a *dual*

problem and is given as :

$$\max_{\alpha} \mathbf{w}(\alpha) = \max_{\alpha} \left(\min_{\mathbf{w},\, b} \mathcal{L}(\mathbf{w}, b, \alpha)\right). \tag{3.25}$$

Then the dual problem can be solved by differentiating the Lagrangian function with respect to $\mathbf{w}$ and b to be equal to 0 and is given as :

$$\bar{\alpha} = \sum_{i}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j}^{\ell} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i, \mathbf{x}_j) \tag{3.26}$$

subject to the constraint :

$$\alpha_i \geq 0 \ , \ \sum_{i}^{\ell} \alpha_i y_i = 0 \tag{3.27}$$

which can be solved using *Quadratic Programming* (QP) with a linear constraint. In addition, *Karush Kuhn Tucker* (KKT) states that the Lagrange multiplier α and the dual function (Eq. 3.26) must be of a non-zero value. Hence, the linear constraint in Equation (3.23) needs to satisfy this KKT condition, such that :

$$\alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1) = 0 \ \ \forall i, ..., \ell \tag{3.28}$$

hence, only points with $\mathbf{x}_i$ which satisfy

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) = 1 \tag{3.29}$$

will have non-zero Lagrange multipliers. These values are then called *Support Vectors* (SVs) and are those points which are closest to the decision boundary. These represent those points that are most difficult for the machine to assign a class. Furthermore, the SVs represent the sparseness of the training set (due to the dual problem solution) and will be used for prediction. Hence, the optimal hyperplane is given as :

$$f(\mathbf{x}) = (\sum_{i=1}^{n} \alpha_i y_i (\mathbf{x}_i, \mathbf{x}) + b) \tag{3.30}$$

where b is the bias. The bias can be calculated implicitly or explicitly; implicitly because some kernel functions will themselves contain the bias. The explicit bias can be calculated as:

$$b = -\frac{1}{2} \sum_{i \in Svs} \alpha_i y_i (\mathbf{x}, \mathbf{x}_i) \tag{3.31}$$

It has been shown that even with the different approaches in obtaining the bias, both still provide reasonably good results (Gunn 1998). For the case where the b are calculated implicitly, the linear constraint in Eq. 3.27 (i.e. $\sum_i^\ell \alpha_i y_i = 0$) is not required. The optimum weight vector $\mathbf{w}$ can be obtained as :

$$\mathbf{w} = \sum_{i \in Svs} \alpha_i y_i \mathbf{x}_i \tag{3.32}$$

and the margin is derived from Eq. (3.21).

The description above is based on a linear separable case or hard margin, which implies that it is a noise free problem. However, a more realistic case will be a linear non-separable case, as it can accommodate problems with noise and hence allow for the classes to overlap. This can be implemented using the margin disturbed classifier (Taylor 1998) or the soft margin approach (Cortes & Vapnik 1995). The distributed classifier adds a constant factor to the kernel function output whenever the given inputs are identical. On the other hand the soft margin approach defines prior, the size of the training weight as an upper bound. In both cases, the magnitude of the constant factor controls the number of training points that the system weights.
The soft margin approach is described in more detail as follows. Two situations can occur in the margin : when a point falls inside the margin but in the right class (i.e. $0 < \xi \leq 1$) or in the wrong class (i.e. $\xi > 1$) with respect to the optimal hyperplane (see Fig. 3.1). The ξ_i are non-negative scalar variables known as *slack variables* and are used to measure data points which lie within the margin corresponding to its border. In this case, it requires the accommodation of some error in the decision boundary. A non-negative scalar variable, ξ_i, can be added to the linear

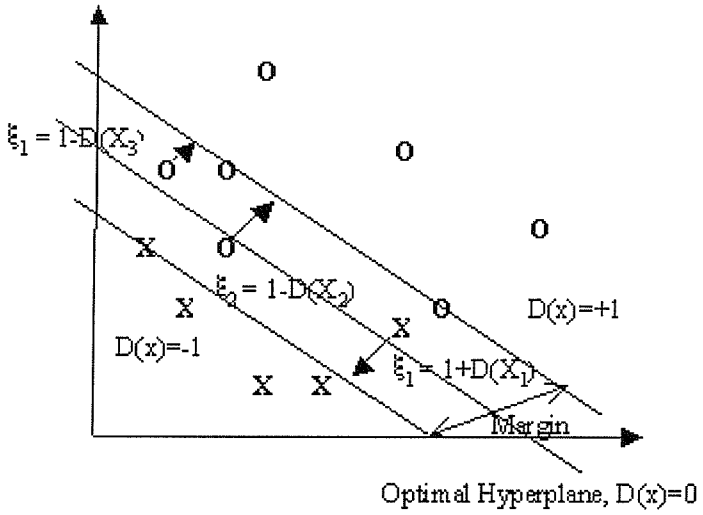


Figure 3.1: SVM Non-Separable case decision boundary, slack variable, ξ and margin. Three points in this figure are non-separable. The subscript 1 and 2 are misclassified while 3 is classified correctly. The ξ measures the errors with respect to their corresponding class hyperplane. The optimal hyperplane is obtained by maximising the margin between the class.

discriminant function in Eq. (3.23), and is rewritten as :

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \; \forall i = 1, ..., \ell \tag{3.33}$$

subject to the constraint :

$$\xi_i \geq 0 \; \forall i = 1, ..., \ell \tag{3.34}$$

The new minimisation problem is given as the cost function :

$$\phi(\mathbf{w}, \xi) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + CG(\sum_{i=1}^{\ell} \xi_i^{\sigma}). \tag{3.35}$$

$G(.)$ is a free function except that it must be a convex function with $G(0) = 0$. For $\sigma = 1$, the number of errors can be counted within the margin and sometimes we address it as the 1 norm loss function. It is possible to set $\sigma = 2$, this becomes a quadratic loss function. However, $\sigma = 1$ is commonly used as it is easy to interpret (Cortes & Vapnik 1995). The C can be considered as "prior knowledge" or a "regulariser" of the data noise as it controls the tradeoff between the complexity of the decision boundary and the number of errors allowed which is known as the

capacity control. Therefore, the Lagrange function for the non-separable case is :

$$\ell(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{\ell} \xi_i - \sum_{i=1}^{\ell} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{\ell} r_i \xi_i \tag{3.36}$$

with $0 \leq \alpha_i \leq C$ and $r_i \geq 0$ as the Lagrange multipliers and we want to minimise with respect to $\mathbf{w}, b$ and ξ. Solving this problem is exactly the same as the separable case except for the constraint of the Lagrange multiplier, α_i, Eq. (3.27) which has an upper limit of C.

What has been described so far is for the case where the decision boundary is linear. Where a non-linear decision boundary is appropriate, the kernel described in section 3.7.2 can then attach to the input and then transform it to the feature space where the non-linear decision is obtained.

3.8.1 SVM for Multi-Class Problems

The above binary classification problem can be extended to k-class classification problems; k-class pattern recognition problems are usually solved using the voting scheme method based on binary classification decision functions. In SVM, the most commonly used is the one-against-the-rest voting schemes (Blanz *et al.* 1996). That is the k^{th} classifier constructs a hyperplane between class k and the $k-1$ other classes. This method requires k binary classifiers to be constructed. For a given test point, a voting scheme (e.g. the winner-takes-all, tree voting) can then be used to assign the class with largest positive output (assuming the output values are real). Another approach is the One-against-One; this approach requires $k(k-1)/2$ hyperplanes to be constructed, separating each class from the other classes and then uses the voting schemes to assign the class for a test point. This approach was extended to incorporate tree voting schemes into the testing phase by (Platt *et al.* 2000). A more natural way to solve the k-class problem is to construct a decision function by considering all classes at once (Weston & Watkins 1998) rather than constructing the combination of binary classification rules. This approach attempts to generalise the binary classification support vectors method with ordering of the constraint for the hyperplane through the piecewise linear

separation by the maximum of k linear functions. This will allow the quality of each hyperplane to be measured individually. Weston show that the results obtained on benchmark data sets suggest that his new approach can reduce the number of support vectors and hence kernel computation. Furthermore with this approach, the problem that the voting scheme may become stuck at for example a draw, is not encountered.

3.8.2 SVM for Regression Estimation

The ideas of SVM classification can be applied to regression problems by introducing a more robust loss function that measures the difference between the target and the predicted values. Whilst there are a few loss functions that SVM regression problems can accommodate, the two most commonly used are ϵ-insensitive and the quadratic loss function defined respectively by :

ϵ-insensitive loss function

$$\mathcal{L}\epsilon(y, f(\mathbf{x})) = \begin{cases} 0 & \text{for } \mid f(\mathbf{x}) - y \mid < \epsilon, \\ \mid f(\mathbf{x}) - y \mid -\epsilon & \text{otherwise} \end{cases} \tag{3.37}$$

where ϵ is a prescribed parameter, that represents the allowance for the insensitivity to the error (that is error within this range is not penalised).

quadratic loss function

$$\mathcal{L}q(y, f(\mathbf{x})) = (f(\mathbf{x}) - y)^2 \tag{3.38}$$

The task of the loss function is therefore to minimise the cost function of :

$$\phi(\mathbf{w}, \xi, \xi^*) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C(\sum_{i=1}^{\ell} \xi + \sum_{i=1}^{\ell} \xi^*). \tag{3.39}$$

where C is a user prescribed parameter (i.e. capacity control), and ξ, ξ^* are slack variables representing the upper and lower errors on the model output respectively. These result in an optimisation process which leads to the well known QP problem given for each function respectively as :

ϵ-insensitive loss function

$$\max_{\alpha,\alpha^*}\mathbf{w}(\alpha,\alpha^*) = \max_{\alpha,\alpha^*}\left\{\begin{array}{l} -\frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(\mathbf{x}_i,\mathbf{x}_j) \\ +\sum_{i=1}^{\ell}\alpha_i(y_i-\epsilon)-\alpha_i^*(y_i+\epsilon) \end{array}\right\} \quad (3.40)$$

subjected to :

$$\begin{aligned} 0 \leq \alpha_i \leq C, \quad i = 1, ..., \ell \\ 0 \leq \alpha_i^* \leq C, \quad i = 1, ..., \ell \\ \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*) = 0 \end{aligned} \quad (3.41)$$

making use of the KKT conditions (i.e. $\alpha^*\alpha = 0$), the support vectors are then one of the Lagrange multipliers with non-zero values. The regression function is then given by $f(\mathbf{x}) = (\mathbf{w}^T\mathbf{x}) + b$ where $\mathbf{w} = \sum_{i\in Svs}(\alpha_i - \alpha_i^*)K(\mathbf{x}_i,\mathbf{x})$ and $b = -\frac{1}{2}\mathbf{w}[K(\mathbf{x}_r,\mathbf{x}_i) + K(\mathbf{x}_s,\mathbf{x}_i)]$ where there are two support vectors for the upper and lower values.

quadratic loss function

$$\max_{\alpha,\alpha^*}\mathbf{w}(\alpha,\alpha^*) = \max_{\alpha,\alpha^*}\left\{\begin{array}{l} -\frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(\mathbf{x}_i,\mathbf{x}_j) \\ +\sum_{i=1}^{\ell}(\alpha_i-\alpha_i^*)y_i - \frac{1}{2C}\sum_{i=1}^{\ell}(\alpha_i^2+\alpha_i^{*2}) \end{array}\right\} \quad (3.42)$$

Making use of the KKT conditions and letting $\beta_i = \alpha_i - \alpha_i^*$. The quadratic optimisation problem can be simplified as :

$$\bar{\beta} = \arg\min_{\beta}\frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{n}\beta_i\beta_j(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{\ell}\beta_i y_i + \frac{1}{2C}\sum_{i=1}^{\ell}\beta_i^2 \quad (3.43)$$

subjected to :

$$\sum_{i=1}^{\ell} \beta_i = 0 \tag{3.44}$$

The regression function is then given by $\mathbf{w} = \sum_{i \in Svs}(\beta_i)K(\mathbf{x}_i, \mathbf{x})$ and $b = -\frac{1}{2}\mathbf{w}[K(\mathbf{x}_r, \mathbf{x}_i) + K(\mathbf{x}_s, \mathbf{x}_i)]$ where there are two support vectors for the upper and lower values.

It is possible to convert the regression task to classification. This can be achieved by letting the target y be set as $+/-$ 1 in Eq. 3.38. However, there is an issue involving computing the parameters involved. For the case of the ϵ-insensitive loss function, we have to select the upper and lower values of the parameter ϵ. On the other hand, using the quadratic loss function, the sparse representation inherent by the Dual representation for the QP (in Eq. 3.25) is unavailable now as all the training samples are taken to be support vectors.

3.9 SVM Parameter Tuning

In the previous section we have described how SVMs are constructed and the types of task they can perform. The next question is how to obtain SVM optimal hyperparameters (e.g. capacity control and kernel parameters). A common method for doing this is estimating the generalisation error by cross-validation methods (such as the Leave-One Out (LOO) (Wahba *et al.* 1999, Chapelle & Vapnik 1999, Joachims 2000, Herbrich 2001)), this is a time and computationally expensive approach. As such, it can be extended to the k-fold cross validation which is more computationally desirable. Much of the work by the above authors on tuning SVM parameter concentrates on minimising the VC-bound (to provide good generalisation error), which is to approximate the VC dimension by $E(R^2A^2(\sigma))$ in Eq. 3.14. (Joachims 2000) uses $\xi - \alpha$, (Chapelle & Vapnik 1999) uses the approximation span rule on the support vectors to estimate the upper bound. The extension of the LOO SVM to provide better generalisation is known as the Adaptive Margin (AM) SVM (Herbrich 2001). This approach provides an automatic tuning to its margin and it will be described in more detail in chapter 4.3.3. (Wahba *et al.* 1999) use

the Generalised Approximation Cross Validation (GACV), which is a computable proxy for the Generalised Comparative Kullback-Liebler distance (GCKL distance). The GCKL is an upper bound for the misclassification rate, given by

$$GCKL(\lambda) = E_{true}\frac{1}{\ell(1-yf)_+} \tag{3.45}$$

where λ is the regularising parameter, ℓ is the number of data, f is the approximated function, E_{true} is the expected true loss, $\tau = 1 - yf$ and $(\tau)_+ = \tau$ if $\tau \geq 0$ otherwise 0. While

$$MisClassification(\lambda) = E_{true}\frac{1}{\ell(-yf)_+} \tag{3.46}$$

where in this case the $\tau = -yf$ and $(\tau)_+ = 1$ if $\tau \geq 0$ otherwise 0. It is note worthy that the λ is associated with the f and is used to obtain the minimised GACV and hence GCKL.

Work by (Weston *et al.* 2000) views tuning the parameter as feature selection. Instead of minimising the bound, one can also use feature selection to provide a good generalisation performance. They use the p-norm for minimising the parameters of the model :

$$\min_{\mathbf{w}} \| \mathbf{w} \|_p \tag{3.47}$$

subject to the constraint of Eq. 3.23. The standard SVM uses a 2-norm for minimising the weight parameters (i.e. $\frac{1}{2} \| \mathbf{w} \|^2$ in Eq. 3.22) which provides an easy solution. (Weston *et al.* 2000) uses the 0-norm (i.e. $\| \mathbf{w} \|_0 = card\{\mathbf{w}_i | \mathbf{w} \neq 0)\}$) which was used directly in the learning machine. Solving this is a non-polynomial (NP) problem. Therefore, an approximation of this NP problem is then studied. Note : this approach has no sparse constraint and also may exhibit local minima (but at least it can be solved using constrained gradient approaches). The usefulness of this approach depends on the problem at hand, for example whether the data has a lot of irrelevant features. It requires a prespecified number of features and therefore can be used as a feature selection algorithm to reduce the number of

irrelevant features. (Duan & Poo 2001) use the $\xi - \alpha$, span rule and cross-validation described with three sets of benchmark data samples and show that although k-fold cross validation is computationally expensive it provides the best estimate for generalisation error.

3.10 Summary

Given a set of data for classification, the goal set by the classical approach is interpreted as setting an accurate probabilistic density estimation of the data. This is not appropriate for a classification system especially with small sample data set. As such, learning machines known as SVMs have been developed based on SLT which effectively describe small data sets. The SLT is the current best known theory developed for small data set learning (Cherkassky & Mulier 1998). This chapter provides a basic understanding of SLT and the different goals set for classical pattern recognition and SLT have been highlighted. The main components that build a learning machine namely, the loss function, induction principle, set of decision functions and an algorithm to build the machine have been described. In accordance with SLT, the complexity of classification is lower than for the regression task, and it should be constructed prior to the regression task. Part of the success of the SVM is its use of the structural risk minimisation (SRM) and kernel functions. The SRM handles the complexity of the model associated with data size and the kernel function handles the dimensionality mapping from input space to feature space. The relationship in SRM between the SVM and the RN has been highlighted as both use kernel methods for learning. Their main difference is the way in which they minimise the risk bound of the parameters involved. The SVM tasks can be extended to multi-class classification and regression estimates and have also been described here. Finally, tuning of the SVM parameters focuses on minimising its expected risk bound, hence providing generalisation to the model.

In a pattern classification machine learning, a common problem is the imbalanced data. The following chapter will describe what modification is required for standard learning machines to be used for imbalanced data.

Chapter 4

SVM For Imbalanced Data

Learning algorithms used in machine learning are usually inappropriate for imbalanced data as they assume no sampling bias and that the misclassification cost for both classes are the same. This chapter therefore describes why imbalanced data is important and how to deal with it for classification problems. The conventional classification performance criterion using the Arithmetic Mean (Amean) is biased towards the majority class for imbalanced data. As such, we introduce the Geometric Mean (Gmean) that is less affected by extreme values. Gmean is the point in the Receiver operating Characteristic (ROC) which is maximised when the classification rate between both classes are balanced. Several SVM extension techniques are then reviewed which offer different ways of dealing with imbalanced data for classification.

4.1 Curse of Imbalanced Data

Imbalanced data in classification can be defined as occurring when the data of one class is heavily represented while the other is under represented. This is a very common problem seen in most practical learning problems, such as fault diagnostics, conditional monitoring, and can be found in many fields such as medical, nuclear processing plants and metallurgy. The data for a given minority/positive case in most diagnostic problems is less than (or under represented) the majority/negative case (or heavily represented). It is often difficult and expensive to obtain the minority information. As such, we have imbalanced data associated with our learning problems. (Provost 2000) has provided a good review as to why learning problems do not perform well with such imbalanced data, since the goal set by most learning

approaches is to maximise the accuracy of classification rate and also the classifier assumes that the training and testing distributions are consistent. This assumption made by most machine learning is for computational convenience. The accuracy of a learning problem in this case is therefore always biased towards the heavily represented class. Also, the distributions between the training and testing are usually prespecified rather than randomly selected.

The simplest way to handle the problem of imbalanced data is to threshold the output. For example, in a artificial neural network (ANN), the output of the model is the posterior probability of the class membership and can be thresholded. This is a more powerful and useful representation for classification than that provided by networks which only provide the discriminant between classes directly (for example Fisher linear discriminant and classical SVM). This scaling technique can be extended to scale its weight update as well. Other techniques such as modified sampling can be used for imbalanced data. The modified sampling techniques' aim is to balance the training data either by upsampling (replicating the minority class) or downsampling (ignoring some cases in the majority class). However, with this approach, the original distribution of data which might be useful for interpretation is likely to be lost unless there is an appropriate criteria for selecting important or redundant data. Another well known sampling technique similar to the Monte Carlo approach is the Bootstrapping technique. This technique does not require assumptions regarding sampling distribution.

Typically, the imbalanced data may require a different Misclassification Cost (MC). The MC due to wrong classification of one class might be more heavily represented than the other class. Usually, the minority class should invoke a higher MC because it is the phenomenon of interest. As such, the MC for the minority is more than the majority class. This will shift the accuracy towards preferentially classifying minority class. The most common example of why the observed (training distribution) may not represent the target distribution is the sampling bias. The sampling bias arises due to the training data being sampled in a way which is not completely random (i.e. a bias from the true distribution). If we consider the case where

we simply select the training data randomly from the actual population, then we may arrive at the situation where one (or more) classes are under-represented for training. For example, the prior distribution of the positive and negative classes is $P(y = +1) = 0.01$ and $P(y = -1) = 0.99$ respectively. If we randomly select the training data with 100 data points, we will have only 99 data points representing the negative class with 1 data point representing the positive class. It will therefore be impossible to achieve good generalisation performance. Instead, we can select the training set in such a way that it is not truly random. For example, in the standard learning machine, the assumption is made that the prior is for balanced classes by choosing $P(y = +1)_{tr} = P(y = -1)_{tr} = 0.5$. By upsampling the positive class and downsampling the negative class, the classes are balanced but the true distribution of classes is no longer guaranteed. This requires us to have a learning procedure to adapt to the true representation of our classification algorithm more appropriately. This analogue of the true distribution between the target for the training and testing classes might be different. As such, it is necessary to compensate for this, therefore, learning from imbalanced data incorporating two different priors for training and testing can be formulated from Bayes theorem as follows : Assuming that the probabilistic likelihood of the data and prior of the train class and test class are the same :

$$\begin{aligned} p_{tr}(\mathbf{x}|y) &= p_{te}(\mathbf{x}|y) \\ p_{tr}(\mathbf{x}) &= p_{te}(\mathbf{x}) \end{aligned} \tag{4.1}$$

and the posterior probabilities of input $\mathbf{x}$ for the test and train class are:

$$\begin{aligned} P_{te}(y|\mathbf{x}) &= \frac{p_{te}(\mathbf{x}|y)P_{te}(y)}{P_{te}(\mathbf{x})} \\ P_{tr}(y|\mathbf{x}) &= \frac{p_{tr}(\mathbf{x}|y)P_{tr}(y)}{P_{tr}(\mathbf{x})} \end{aligned} \tag{4.2}$$

then putting Eq. 4.1 and Eq. 4.2 together, we get :

$$P_{te}(y|\mathbf{x}) = P_{tr}(y|\mathbf{x})\frac{P_{te}(y)}{P_{tr}(y)} \tag{4.3}$$

The optimal prediction from a testing example then becomes :

$$P_{te}(y|\mathbf{x}) = \text{argmin}_{y \in Y} \sum_{j=1, y_j \neq y}^{m} \frac{P_{te}(y_j)}{P_{tr}(y_j)} P_{tr}(y_j|\mathbf{x}) MC(y_j) \tag{4.4}$$

where m is the number of class and $MC(y_i)$ is the cost deal to misclassification. It is important to incorporate MC as the important of misclassification cost of each class can be specified. This provides a natural way to view problems of imbalanced data in a learning problem and will be used in section 4.3.2 to developed the Non-Standard Situation (NSS) SVM.

4.2 Performance Criteria for Imbalanced Data

A confusion matrix is a useful tool for visualising the performance of most classification problems. It consists of the number of points in the data set corresponding to four categories: False Positive (FP), False Negative (FN), True Positive (TP) and True Negative (TN). TP and TN are the correct prediction while FP and FT are the wrong prediction. Table 4.1 is a representation of the confusion matrix for a two class problem. Due to the curse of the imbalance of data, using standard

		Target	
		P	N
Prediction	P	TP	FP
	N	FN	TN

Table 4.1: Confusion Matrix

performance criteria such as the Arithmetic Mean (AMean) to assess the classification rate of the train and test set is not applicable in this case. For example, a system with an AMean of 50% may be dominated by one class providing 90% whilst the other provides 10% classification rate with respect to the majority and minority classes respectively. In imbalanced data applications, the prediction from the minority class is usually more important as explained earlier. As such, the above system cannot be used and an appropriate criterion has to be used.

4.2.1 Receiver Operating Characteristic (ROC) and Geometric Mean (GMean) Analysis

The ROC analysis was initiated from the field of signal detector theory (Egan 1975) and has been extended for use in machine learning systems to compare the relationship between classifiers (Provost & Fawcett 1997). The ROC curve describes the trade-off between sensitivity ($\frac{TP}{TP+FN}$) and specificity ($\frac{TN}{TN+FP}$) values which can be obtained from the confusion matrix. The range of the ROC curve is from 0 to 1 in sensitivity and specificity and the best solution of the classification system can be compared to the worst, with the best on the top left corner and worst on the lower right corner. The ROC curve then allows us to represent simultaneously the classifier performance by two degrees of freedom for a range of possible classification thresholds using the plot of TP and FP. Figure 4.1 shows an illustration of the ROC curve. The advantage of the ROC is that the performance of the classifier is independent of the class distribution (i.e. the classification rate is not affected by the majority class). Furthermore, the ROC captures, in a single graph, the various alternatives that are available to the user as they move their criteria into higher or lower levels.

Another way of evaluating the imbalance of data is by forcing the accuracy between two classes to be balanced (Kubat *et al.* 1998). This is known as the geometric mean (Gmean) and it is less sensitive to skew distribution than the Amean. A simple view of the difference between AMean and GMean is that, given a set of numbers (e.g. the classification rate), we want to represent it with a number. The question now arises as to how these sets of numbers can be combined and be represented by a single number? The ways to combine the set of numbers for the case of the Amean is adding them together while for the case of Gmean the numbers are multiplied together. The GMean is less affected by extreme values than the Amean and it is a useful measure of the central tendency for some positively skewed distributions. The formula for the GMean is defined as :

$$\text{Gmean} = \prod_{i=1}^{m} (B^i)^{1/m} \tag{4.5}$$

where B is a set of positive numbers (which in our case, is the classification rate) and m is the number of classes. Note that the Gmean is equal to AMean when all the B's are equal.

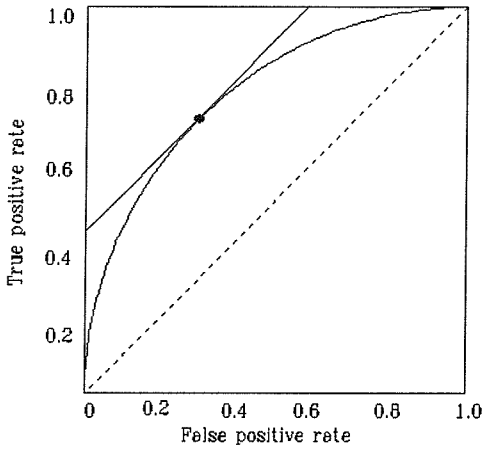


Figure 4.1: Example of a ROC curve showing the plot of TP vs FP. The curve corresponds to different thresholds used for the classifier. The best solution of the system can be compared to the worst with the best on the top left corner and the worst on the lower right corner. The • which forces the classifier to have a balanced classification between both classes corresponds to a typical Gmean in the ROC curve.

4.3 SVM Extension Techniques

The building up of the SVM has been described in Chapter 3. The main advantage of SVM is that it was developed based on the SLT which is best used to describe small sample data sets. SVM was also developed based on classification problems. There is a huge list of applications of SVM in a diverse range of fields such as: image classification, 3-D object orientation, text categorisation, hand written digital recognition etc ; a comprehensive list can be found in Isabelle Guyon's web page (http://www.clopinet.com/isabelle/Projects/SVM/applist.html). However, little attention has been paid to SVM in handling problems of imbalanced data. Early work by (Veropoulos *et al.* 1999), impose a different MC associated with each class. This approach is related to imbalanced data but the problem of sampling bias which occurs with imbalanced data has not been resolved by them. (Lin *et al.* 2000) then extended their work to incorporate both MC sampling bias costs for imbalanced data into SVM and this is called Non-Standard Situation (NSS) SVM, which

was built using the framework of the regularisation network as described in chapter 3.7.3. Other work that relates to imbalanced data in SVM is the classification of the microarray gene (Brown *et al.* 1999). They used the margin distribution approach to deal with imbalanced data and used a regularisation parameter associated with the ratio of the class prior and the operational prior distribution. (Cawley & Talbot 2001) uses this class prior with the soft-margin approach. Both have been shown to produce good results in the problem that they investigated. In other techniques such as Adaptive Margin (AM) SVM (Herbrich 2001) which we investigated for imbalanced data, the margin is adapted automatically to fit each of the training data set. The standard AM SVM is built upon the Leave-One Out (LOO) SVM. To accept more outliers, the AM SVM is incorporated with a regulariser making it the generalised LOO SVM. We then make modifications to the misclassification cost incorporating sampling bias and misclassification cost into the AM SVM for imbalanced data. The following sections describe these SVM extension techniques in more detail.

4.3.1 Control Sensitivity (CS) SVM

The capacity control C (sometimes known as the variance of the noise data) in the SVM is used to control the tradeoff between the complexity of the decision boundary and the network capacity of the number of misclassification errors (i.e. how many errors can be tolerated with the training data). By splitting the C according to the respective classes implies that the MC associated with each class is different. This was originally proposed by (Veropoulos *et al.* 1999) and can be extended to the use of imbalanced data as it effectively incorporates a different cost function for each class. Note : however, in this approach the sampling bias is not incorporated. The standard soft-margin approach for SVM can be extended to the use of Control Sensitivity (CS) SVM, where the Lagrangian in Eq. 3.36 now has different cost functions associated with it. This is used to accommodate the two different cost

functions for each class and is rewritten as :

$$\mathcal{L}(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C^+ \sum_{i=1|y_i=+1} \xi_i + C^- \sum_{i=1|y_i=-1} \xi_i - \sum_{i=1}^{\ell} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{\ell} r_i \xi_i \quad (4.6)$$

with $0 \leq \alpha_{i|y_i=+1} \leq C^+$, $0 \leq \alpha_{i|y_i=-1} \leq C^-$ and $r_i \geq 0$ as the Lagrange multipliers. We can extend the use of the 1-norm (i.e ξ) cost function to the 2-norm (i.e ξ^2) and the dual formulation can now be written as :

$$\mathcal{L}(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{4C^+} \sum_{i|y_i=+1} \alpha_i^2 - \frac{1}{4C^-} \sum_{i|y_i=-1} \alpha_i^2. \quad (4.7)$$

This implementation can be carried out in a standard SVM by adding the $\frac{1}{4C^+}$ and $\frac{1}{4C^-}$ term onto the diagonal of the kernel function with respect to their appropriate classes. This approach can be viewed as implementing an asymmetric margin classifier in order to describe the misclassification risk similar to that of the margin distribution by (Taylor 1998).

CS SVM shows that it is possible to incorporate different cost functions associated with different classes. However, there are now two parameters, C^- and C^+ which need to be pre-specified or tuned and a way of measuring the performance for each combination of the C's is required. It is always difficult to determine the realistic cost for misclassification in each class and hence, the combination of $C's$ can be large which results in a large computational burden.

4.3.2 Non-Standard Situation (NSS) SVM

The sampling bias is a problem that is typically inherent in imbalanced data because the data selected for training needs to be pre-specified (sampled individually) rather than selected randomly, hence violating the random principle of sampling (all samples are equally sampled with equal probability). This leads to the necessity of differentiating the prior distribution for training from that for testing and incorporating the misclassification cost into the loss function for the imbalanced data. (Lin

et al. 2000) developed the SVM for non-standard situations based on the regularisation framework following (Lin 1999) and established the relationship between the standard SVM and the Bayes theorem. Lin showed that the estimates of the SVM from the sampling set is related to Bayes rule as sign$[P(Y = 1|X = \mathbf{x}) - 0.5]$ as the number of data gets to infinity.

For the case of imbalanced data, the MC and the sampling cost can be incorporated into the Bayes theorem as described in section 4.1. For the case of a two class problem, the training and testing posterior probability of the input $\mathbf{x}$ can be written as :

$$P(y|\mathbf{x}) = \frac{\pi_{tr}^{+} f^{+}(\mathbf{x})}{\pi_{tr}^{+} f^{+}(\mathbf{x}) + \pi_{tr}^{-} f^{-}(\mathbf{x})} = \frac{\pi_{te}^{+} f^{+}(\mathbf{x})}{\pi_{te}^{+} f^{+}(\mathbf{x}) + \pi_{te}^{-} f^{-}(\mathbf{x})} = \frac{\pi_{tr}^{+} \pi_{te}^{-}}{\pi_{tr}^{-} \pi_{te}^{+}} \tag{4.8}$$

where "+" and "-"are the respective class, π are the prior probabilities in the output population, the subscript "tr" and "te" denote the training class and testing class respectively, and f is the probability density of the input. This leads to the Bayes rule for SVM to be rewritten for imbalanced data as :

$$\phi(\mathbf{x}) = \text{sign}[P_{tr}(y|\mathbf{x}) - \frac{L(-1)}{L(-1) + L(+1)}] \tag{4.9}$$

where L's are the imbalance modification factor (i.e $L(-1) = MC^{+}\pi_{tr}^{+}\pi_{te}^{-}$ and $L(+1) = MC^{-}\pi_{tr}^{-}\pi_{te}^{+}$). The regularised problem for SVM is then modified to that of minimisation problem of :

$$H(f) = \frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i)[1 - y_i f(\mathbf{x}_i)]_{+} + \lambda \|f\|_2 \tag{4.10}$$

where λ is a regulariser (can be thought to be equal to $\frac{1}{2\ell C}$ in the standard SVM), $[.]_{+}$ is a function such that, $\tau_{+}=\tau$ if $\tau > 0$ otherwise 0 (similar to that of the slack variable ξ of the conventional SVM used in section 3.8 Eq. 3.33). The bias term in the SVM is given by:

$$b = \frac{\sum_{i=1}^{\ell} \alpha_i (L(y_i) - \alpha_i)(y_i - \sum_{i=1}^{\ell} c_j K(x_i, x_j))}{\sum_{i=1}^{\ell} \alpha_i (L(y_i) - \alpha_i)}. \tag{4.11}$$

The solution to the above problem can be solved using the QP similar to that

described in chapter 3.8, except now the imbalance modification factor, L's are required to be incorporated corresponding to the capacity control C in the conventional SVM and the regularising parameter λ needs to be determined. this will be described in more detail in chapter 5.5.

4.3.3 Adaptive Margin (AM) SVM

The conventional SVM machine fixed the margin τ (see Eq. 3.21), to separate between the classes. Rather than fixing this margin, Adaptive Margin (AM) SVM adapts the margin automatically. Making the margin sensitive to each point was first proposed by (Herbrich 2001). This is done by formulating the margin error and the support vectors, α_i to be dependent. This idea was based on the Leave-One-Out (LOO) to provide a good generalisation bound (Weston 1999). The bound on the expected risk in Eq. 3.9 can be obtained from the error of the sparse solutions, which in turn is bounded by the ratio of the number of non-zero coefficients of α to the number of training examples ℓ.

In classical SVM, the best choice of training errors and margin depends on the capacity control, C. In AM SVM, the C is fixed since in LOO-SVM, a soft margin is automatically attached. This is because the algorithm does not attempt to minimise the number of training errors - it minimises the number of points that are classified incorrectly even when they are removed from the linear combination that forms the decision rule (Weston 1999). (Herbrich 2001) extended the work of LOO SVM by generalising it. This was done through adding a regulariser term to the loss margin in the constraint. The generalised LOO uses the learning algorithm to minimise the bound of the error directly through slack variables, ξ, and can be written as:

$$\begin{aligned} \text{minimise} \quad & \sum_{i=1}^{\ell} \xi_i & (4.12) \\ & y_i f(\mathbf{x}_i) \geq 1 - \xi_i + \lambda \alpha_i k(\mathbf{x}_i, \mathbf{x}_i) \\ \text{subject to} \quad & \xi_i \geq 0 \\ & \alpha_i \geq 0. & (4.13) \end{aligned}$$

The significance of the regulariser, λ, at each training point is :

- if λ=0, no effort has been made to make the minimisation function smooth. It is based on the empirical risk.
- if $\lambda \rightarrow \infty$, no effort has been made to reduce the empirical risk. It is equivalent to kernel density estimation in each class.
- if λ=1 , this is the LOO SVM.

The regulariser, λ, can then be used to relax the decision boundary and hence allows the application to find the outliers in the data. Now the margin for separating the class is automatically adapted. Previous approaches required a tradeoff between maximising the margin and misclassification of errors. We then extend this approach to our application for imbalanced data by splitting the ξ into two classes associated with their appropriate loss function (i.e. the imbalanced modification factor in Eq. 4.9). Hence, rewriting Eq. 4.12 to :

$$\text{minimise} \qquad L(-1) \sum_{i=1|y=-1} \xi_i \; + \; L(+1) \sum_{i=1|y=1} \xi_i \tag{4.14}$$

subject to constraint of Eq. 4.13.

4.4 Summary

Typical learning machine algorithms are not readily usable for imbalanced data unless some modification is made. The two important factors for imbalanced data are the specification of the accuracy of misclassification rate and the sampling bias which have to be considered. Also, the Gmean performance criterion is more appropriate for imbalanced data as it is less sensitive to large deviation between two outputs. SVM was developed based on SLT and SLT effectively describes statistical estimation with small samples. SVM was also developed based on classification problems. Several SVM extension techniques that may be suitable for imbalanced data have been described in detail in this chapter. The CS SVM was driven by imposing a misclassification cost for each class. The training and testing distribution is not taken into account here. The NSS SVM then incorporates the two factors for imbalanced data into account. The interesting thing about AM SVM is the LOO error. However, the reason for investigating the AM SVM is due to the fact

that the margin is adapted automatically rather than fixed. Among the techniques reviewed, the work by done (Lin *et al.* 2000) called the NSS SVM seems to be more appropriate for our imbalanced data as it has a sound theoretical background incorporating misclassification cost and sampling into SVM training. The imbalanced modification factor (L's) derived from Lin's NSS SVM is then applied to AM SVM.

Obtaining a good classification for imbalanced data is not the end goal of a good classification system. What would be also desirable is to understand the structure of the derived model for interpretation.

Chapter 5

Model Interpretation for Classification

The ultimate goal of a classification system is the classification rate. However, it is often important to justify how the output is derived from the inputs and which input features are the essential ones. There is often a complex relationship between inputs. The class posterior probability is a common way to assess classification problems that provides model interpretability by specifying how confident we are of selecting the appropriate class. Another way to view interpretability of the model is to decompose the model structure into a simpler form and yet retain the model's performance. This is the approach that we have considered in this thesis to provide model structure interpretability by enforcing sparseness of the model. The first section describes interpretability from the classification point of view given some classification system examples and describes how this can provide interpretability. In this work, the interpretability of our model was attempted within the SVM framework. This starts off with the well known additive model structure. The feature selection techniques used in SVM are also briefly described since it provides interpretability for the model. Prior to describing the SUpport vector Parsimonious ANalysis Of VAriance (SUPANOVA) approach, its main components, namely the use of Spline kernels and the ANOVA decomposition function, are described relating these to the additive model. The original work on SUPANOVA was developed for a regression task. The final section is delegated to the implementation of SUPANOVA for classification with imbalanced data.

5.1 Understanding the Interpretability of a Classification System

In a classification problem, the task is to assign a new input to a number of possible labelled outputs. Learning is then the process of determining the model parameters that provide the output on the basis of a given set of data. The performance of the classification is based on the classification rate, which measures how well the learning algorithm is able to discriminate between the classes. In many practical applications, the output from the classifier which simply discriminates between classes is insufficient. For example, an Artificial Neural Network (ANN) (highlighted in chapter 2.5) is viewed as a "black box" classification tool as it is unable to provide a clear explanation as to its output (i.e. it is difficult to interpret its parameters). It is difficult to convince the end user that this classification is correct unless it can provide some understanding of how this output was derived or at least indicate how confident we are for this output compared to other classes (i.e. class posterior probability). The class posterior probability expresses the quantity of uncertainty in prediction while it helps to facilitate the separation between "inference" and "decision" (Duda *et al.* 2000). As our investigation into the automotive material is a two class problem, this thesis concentrates on understanding the parsimonious representation of the classification model rather than the confidence of the model. In classical work such as linear discriminants, the output provides information about the projection of the input space to a one-dimensional space for classification. The interpretability lies in the parameters of the technique that represents the projection from high-dimensional data onto a line and performs classification in this one-dimensional space (Bishop 1995). The projection maximises the distance between the means of the two classes while minimising the variance within each class. Although the parameters provide information about the projection, the strict assumption about the model (being a multi-dimensional Gaussian distributed model and equal covariance matrix) is not realistic for most practical applications. Other techniques such as tree methods adaptively split the input space into disjointed regions in order to construct a decision boundary. The regions are chosen based on a greedy optimisation procedure where in each step, the algorithm selects the

split that provides the separation of the class according to some cost function (a cost that reflects the misclassification risk). Pruning methods are usually used after growing the tree for model selection. The Classification And Regression Tree (CART) algorithm (Breiman *et al.* 1984) is commonly used for a binary tree split. The binary tree structure produced by CART is easily interpretable for a moderate number of nodes. Each node represents a rule involving one of the input variables hence providing interpretability on how the output of the CART is derived. The main problems with this approach are that it is sensitive to coordinate rotation, the solution may be a local minima (due to the greedy search) and also that the region over which local averaging occurs is highly restrictive (i.e partitioning is by a recursive splitting of hyperrectangular subdomains by a plane perpendicular to a selected input). Other partition methods exist such as the nearest neighbourhood method, which uses the Voronoi partition (that is the distribution is the set of points in the plane which are as close or closer to the centre of that disc than to the centre of any other disc in the distribution). As such, the nearest neighbourhood structure is a piecewise regression model like CART but with less restriction. The decision about the boundary is constructed using the m data point nearest to the point of estimate or voting scheme. The problem with this approach is its computational burden for a large data set. Every set of training data has to be recalculated in order to make a prediction. Techniques such as Learning Vector Quantisation (LVQ) (Kohonen 1990) have been used to combat the computational issue by representing a large data set by a smaller number of prototype vectors. Another technique similar to tree methods is the graphical model. A graphical model has the notation for modularity - that is a complex system can be built by combining simpler parts (Murphy 2001). The graphical model uses the theory of the graph and probability. Probability theory links the simpler parts together to provide a whole system which is consistent and also interfaces models to data. The Graphical model theory provides transparent interfaces of models with highly interacting sets of variable as well as their data structure which leads to an easier understanding of the original high dimensional model. A common type of graphical model in machine learning is

the direct graphical model sometimes known as a Bayesian network (Pearl 1998). In probabilistic reasoning, random variables represent an event or an object. The aim is to compute their joint probabilities given the random variable of the current state of the world, however making each and every combination is combinatorially expensive. As such, the Bayesian network recognises that certain random variable pairs may be uncorrelated once information concerning some other random variables is known. This allows us to reduce the chain rule size by eliminating the conditional independence for probabilistic terms while explicitly keeping the joint probability.

However, most techniques described so far are based on setting the initial goal that requires good density estimation. This has been described in the earlier chapter 3.5.2 as a misconception between conceptual needs and technical interpretation. There are many other classification techniques yet to be discussed here, for example Bayesian Neural networks which use the evidence framework to select important features. This is done via adding a hyperparameter into each input feature which provides information about the importance of each input feature in the model. Neuro-fuzzy networks use linguistic explanations for modelling. All this work may provide us with an understanding of the output of the classification which is an important issue for the classification problem.

5.2 Interpretability in SVM via Model Structure

A complex model is typically difficult to interpret. This inability to interpret the model can lead a complex model to be described as a "black box" system. It is therefore important to be able to provide a simpler or parsimonious model to yield interpretability of the model structure. This principle was stated by Occam, that design should take into account the simplicity of the model in addition to good predictive performance. The bound on the expected risk from the SVM can be used as a guide to feature selection in SVM, hence providing an interpretable model. Work done by (Weston *et al.* 2000) uses the 0-norm on the weights to provide direct feature selection for SVM. This approach of feature selection is to reduce the number of features used and also preserve or improve the discriminative ability of

the classifier at the same time. It is important because it affects the running time requirements and interpretation issues imposed by the problems. In this thesis, the model structure approach has been focussed upon to provide interpretability to the model by decomposing the model into simpler terms, hence providing interpretability for a high dimensional input problem.

The additive model is an attractive framework that has been used to establish the generalisation of linear models (Hastie & Tibshirani 1990). This additive model was used to avoid the dimensionality problem. (Rasmussen 1996) has shown that the additive model has many successes when used in the learning machine community. His work concentrates on the predefined nature of additive models and determines whether it can capture the properties of the physical data. Additive models are useful because they are a superposition of one-dimensional functions. As such the effect of different variables can be examined separately. With such properties, this model is attractive as it provides easy interpretation of the model. A simple additive model is:

$$f(\mathbf{x}) = \sum_{i=1}^{N} f_i(x^i) \tag{5.1}$$

where N is the number of input dimension, $\mathbf{x}$ is the input vector and f_i are the univariate functions. A sparse representation of the model structure can be enforced on the additive model. This has been used in the signal processing community to decompose any signal into a linear expansion of waveforms (waveforms are discrete time signals with specified length) (Mallat & Zhang 1993). A large number of basis functions that were linear superpositions were built and weighted coefficients were associated with each basis function. Picking out the important basis function or sparse representation from this large number of basis functions (known as the *Dictionary*) requires an enforcement of the weight associated with each basis function. This is related to a learning problem by minimising the following expected cost function :

$$E[\mathbf{a}] = \mathcal{L}(f(\mathbf{x}), \sum_{i=1}^{J} a_j \phi_j(\mathbf{x})) + \lambda \parallel \mathbf{a} \parallel_0 \tag{5.2}$$

Where $\mathcal{L}$ is the loss function between the approximation and the sparse representation, 0-norm counts the number of non-zero values of $\mathbf{a}$, λ is a parameter that measures the tradeoff between sparsity and approximation (or is chosen as proportional to the noise) and J is the number of basis functions, $\phi_j(\mathbf{x})$. Therefore, a large value of λ implies a more sparse representation or more coefficient a's become zero. The loss function needs to be inferred from the given data set and is known as the empirical risk minimisation (as described in chapter 3.5.1). Typically, it uses an empirical approach with a convex loss function (i.e. $\min \frac{1}{\ell} \sum_{i=1}^{\ell} (y_i - \sum_{i=1}^{J} a_j \phi_j(\mathbf{x}))^2$). Note : this minimisation assumes that the true function or the target y is corrupted by additive noise. The problem with this additive model is that the 0-norm of the coefficient is a non-polynomial hard problem to solve because it requires a search through all the combinations (Chen *et al.* 1999). Therefore, an alternative is to use a greedy method to estimate the cost function or different norms to enforce a sparse representation. The early work uses the greedy method known as "matching pursuit". It starts with an initial approximation scheme with a square error loss, and the basis functions are added iteratively to the model. It is note worthy that this is similar to that of a Radial Basis Network for given Gaussian basis function and the same number of basis functions. The use of 2-norm for minimizing coefficients was implemented in the Method of Frames (MOF) (Daubechies 1992) in wavelets. This approach has computational advantages, however, sparsity is not preserved. (Chen *et al.* 1999), used the 1-norm (which is the summation of the absolute value of the coefficients) instead of an approximation to the 0-norm or 2-norm, and this is known as Basis Pursuit De-Noising (BPDN). The computation cost for BPDN is still expensive, even with linear programming as obtaining its goal minimum requires a computation of all terms in the dictionary term.

In SVM, the kernel is a tool used to map the input dimensional to a high non-linear feature space. Attempts to decompose the associated kernels used in SVM is to provide a sparse kernel and hence, an interpretable model. Here, we have described how a sparse kernel can be obtained by enforcing different 1-norms to the kernel coefficient. Several kernels are described in Chapter 3.7.2. Not all kernels

fit nicely into this additive model framework. Next, we describe how the SVM can obtain its model interpretability using the unique properties of the spline kernel and the Reproducing Kernel Hilbert Space (RKHS) enabling their tensor products to be produced.

5.3 Spline Kernels and ANOVA Decomposition Functions

Splines are good for modeling due to their ability to approximate arbitrary functions, shown by (Wahba 1990). It provides a natural and flexible approach to density estimation which has been shown to couple well with data that are sparse. Splines are not parametric in a function form, but they can be written as a linear combination of basis functions that usually have a polynomial representation. B-splines are computationally advantageous and favorable when a rule base is described (Brown & Harris 1994) and are widely used in neuro-fuzzy networks. However, (Gunn 1999) has observed experimentally that they have the tendency to oscillate. While infinite splines have no oscillation problem, there are no scales involved. Hence, no parameter has to be determined, making it very attractive. This motivates the use of spline kernels within the ANOVA framework, as the ANOVA decomposition would produce a magnitude of such parameters which need to be determined. The simple first order spline kernel with infinite nodes, which passes through its origin, is a piecewise cubic with knots located at a subset of the data given as in Eq. 3.19.
Kernels can be constructed from their tensor products of other kernels. As such, extended kernel functions can be constructed from the additive sum in terms of a Mercer theorem (described in more detail in the next section). This enables the learning problem with an additive spline model to become :

$$\begin{aligned} F(\mathbf{x}) &= \sum_{\ell} \mathbf{w}_\ell (\sum_{j=1}^{N} K(x_\ell^j, x^j)) \\ &= \sum_{j=1}^{N} (\sum_{\ell} \mathbf{w}_\ell K(x_\ell^j, x^j) \\ &= \sum_{j=1}^{N} f_j(x^j) \end{aligned} \tag{5.3}$$

Where ℓ is the number of samples, N is the number of input dimensions, $\mathbf{w}$ is the weight associated with each univariate spline kernel. This then forms a learning problem which is an additive model where $f(\mathbf{x})$ is estimated through adding each $f(x)$ associated with a univariate term. A special case of the additive model focuses on building up the model with a univariate (i.e $f(x_1, ..., x_N) = f_0 + f_1(x1) + ... + f_N(x_N)$ satisfying $f_N(0) = 0$ for all variables of N). The interactive model was not considered in this special case of the additive model (Chen *et al.* 1999) as such, it suffers from approximation errors. Let us consider a 2 dimensional integral to illustrate their interactive terms :

$$\begin{aligned} f_0 &= f(0,0) \\ f_1(x_1) &= \int f(x_1, x_2)dx_2 - f_0 \\ f_2(x_2) &= \int f(x_1, x_2)dx_1 - f_0 \\ f_{12}(x_1, x_2) &= f(x_1, x_2) - \int f(x_1, x_2)dx_2 - \int f(x_1, x_2)dx_1 + f_0 \\ f(x_1, x_2) &= f_0 + f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2) \end{aligned} \tag{5.4}$$

This decomposition can be viewed as a functional version of the statistical methodology Analysis Of Variance (ANOVA). The curse of dimensionality will exist in this instance when the order of interaction increases. In most instances, we are interested only in the low interaction terms since they can be more easily interpreted. Hence, a term that enforces a sparse representation of the model (e.g. BPDN) can be incorporated to provide a interpretable model. Putting the flexibility of the spline and the decomposition of the ANOVA function into additive components together with an enforcing term for sparse representation leads to the SUpport vector Parsimonious ANOVA (SUPANOVA) (Gunn 1999).

5.4 SUpport vector Parsimonious ANOVA (SUPANOVA)

The ANOVA kernel has been used by (Stitson & Weston 1996) and has shown good performance. ANOVA kernels have been used in this thesis, electing the sparse ANOVA kernel will produce a parsimonious model, which has a sparse structural representation, and yet is flexible enough to retain the model representation (i.e.

preserve generalisation as well). Understanding the model structure provides a good understanding of the selection of inputs. This is an important issue in any learning problem and it has often been neglected in classification problems or is considered in terms of class posterior probability (e.g. in ANN applications). The sparse model provides easy interpretation with a smaller number of interactive terms (through decomposition of the model). The ANOVA kernel can be incorporated into the SVM framework with an enforcing term to choose a parsimonious model and can written as :

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i \sum_{j} a_j K_j(\mathbf{x}^i, \mathbf{x}); \text{ subjected to } a_j \geq 0 \tag{5.5}$$

where the kernel, K_j is associated with a weighted term, a_j, and j is the number of basis functions. The enforcing term for transparency can then be introduced by careful selection of each weighting term for each kernel (sparse selection). As has been noted in section 5.2, there are several enforcing terms (i.e norms) that can be employed. It has been argued that the BPDN is more appropriate in this case, leading to modification of the constraint of a_j in the above equation to a 1-norm (similar to Eq. 5.2 except that it uses 1-norm). The spline ANOVA kernel uses the infinite spline as it is flexible and has no scale term to be determined. Furthermore, the additive representation of the ANOVA model structure is advantageous as the higher order interactive terms can be ignored, hence leaving small subsets of ANOVA which can be easily visualised. This provides a parsimonious model as opposed to neural networks where the structure of the network is in itself very difficult to interpret. The ANOVA kernels that can be used in SVM must satisfy the Mercer condition, and is stated as :

- if K1 and K2 are positive definite then K1 + K2 and K1 x K2 are positive definite.

Then a multivariate ANOVA kernel can then be written using the tensor product of a univariate plus a bias term,

$$\begin{aligned} K(u,v) &= \prod_{i=1}^{N}(1+k(u_i,v_i)) \\ &= 1+\sum_{i=1}^{N}k(u_i,v_i)+\sum_{i<j}^{N}k(u_i,v_i)k(u_j,v_j)+....+\prod_{i=1}^{N}k(u_i,v_i) \quad (5.6) \end{aligned}$$

Note that each of the additive terms has its own property, since k_1 and k_2 can be expressed individually and it is also similar to that expressed in Eq. 5.4 for the case of a 2-dimensional problem. In built up ANOVA kernels, a univariate is required to satisfy $f_N(0) = 0$ for all variables. This means that the univariate terms will pass through zero and the bivariate and other higher terms will also be constrained to be zero along their axes. As a result, this parsimonious model will favour smaller order terms rather than higher ANOVA terms.

The approach to obtain the model structure is different from the CART algorithm. CART uses a greedy search to provide flexible basis functions using a partitioning approach. However, they may be entrapped locally. Here, we are using the full model and we look into the subsets rather than at the subsets to build the full model. However, the potential problem with any additive model is when the model itself contains high dimensional interactions, whereby the transparency would not be apparent. However the interactive terms can be restricted by a regulariser, λ (Eq. 5.2), although this may provide an interpretable model at the expense of structural integrity.

5.5 SUpport vector Parsimonious ANOVA (SUPANOVA) for Imbalanced Classification

In dealing with imbalanced data, incorporating a modified class dependent misclassification cost function and sampling bias is required for the SVMs described in Chapter 4. The misclassification cost for each class can be implemented in the capacity control, C (i.e. C^+ and C^- for the respective class) as in Control Sensitivity SVM (Veropoulos *et al.* 1999). This can be extended to imbalanced data by imposing a heavy penalty on a skewed class. As such, the sampling bias needs to

be incorporated as given in the case of (Lin *et al.* 2000) for non standard situation (NSS) SVM. The difference lies in the fact that (Lin *et al.* 2000) use the regularisation network and we are using the classical SVM approach. The differences are highlighted here :

- Standard SVM

$$C\sum_{i=1}^{\ell}\xi_i + \frac{1}{2} \parallel \mathbf{w} \parallel^2 \tag{5.7}$$

- Regularisation Network SVM

$$\frac{1}{\ell}\sum_{1=1}^{\ell} L(y_i)\xi_i + \lambda \parallel f \parallel_K^2 \tag{5.8}$$

Comparing the above two equations :

$$C = \frac{L(y_i)}{2\ell\lambda} \tag{5.9}$$

where $L(+1) = MC^{-}\pi_{tr}^{-}\pi_{te}^{+}$ and $L(-1) = MC^{+}\pi_{tr}^{+}\pi_{te}^{-}$ are the imbalanced modified factor given in Eq. 4.9 where MC are the misclassification costs. It should be noted that the ratio of the L is affected by the λ. The exact value of $L(1)$ and $L(-1)$ is not important as opposed to its ratio as the optimum decision is based on the sign of the posterior of the training minus the L's ratio (see Eq. 4.9). This is a reflection of the threshold imposed for the decision boundary. The geometric means (Gmean) were then used to obtain the true classification rate of the positive and negative class of the classifier. This was found to be the best approach to deal with imbalanced data although our previous work was based on using the misclassification cost alone and finding the two different misclassification costs through trial and error (Lee *et al.* 2001*b*, Lee *et al.* 2001*d*, Lee *et al.* 2001*a*). This approach provides a more natural way to allow for the imbalanced data in SVM.

The above described approaches deal with the problem of imbalanced data. This work is extended to provide an interpretable model using the SUPANOVA. A fast way of converting the SUPANOVA from regression applications to classification

problems was described in (Lee *et al.* 2001*e*). This was done by altering the model selection from being based upon Mean Square Error to classification rate. This thesis however, reformulates the regression task to a classification task by changing the quadratic (regression) to a hinge loss function for regularisation. The SUPANOVA technique allows for an additive decomposition of low dimensional kernel models to be recovered, enhancing model visualization. This is a difficult task and is decomposed into 4 stages similar to those of the regression, except that the loss function and the model selection are changed. It is noteworthy that, prior to proceeding, the form of the univariate kernel must be chosen. There are many kernels that can be employed, such as Radial basis functions, polynomials, splines. However, there are additional parameters within many of these kernels that must be determined, therefore, whilst they provide increased flexibility for the model, a significant additional cost is introduced. A spline kernel has been used here as it does not require any additional parameters to be determined and it does have the ability to approximate any function with arbitrary accuracy (Wahba 1990).

The stages involved in SUPANOVA for classification of imbalanced data are :

1. Model Selection

   a good generalisation estimate from the SVM based on GMean provides the value of the two different imbalanced modified factor L's for each class as described in Eq. 5.9.

2. ANOVA basis selection;

   using the values of L's in model selection, Lagrange multipliers, $0 < \alpha \leq L(y_i)$ are obtained. The decision function (below) is decomposed into all its possible sub-components assuming all the a's to be 1.

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i y_i \sum_{j}^{m} a_j K_j(\mathbf{x}_j, \mathbf{x}) \tag{5.10}$$

   where $\alpha_i > 0$ are the Lagrange multipliers, y_i are the targets, a_j are the weighted model coefficients ℓ, n is the number of training patterns and m is the number of additive kernels used in the model.

3. Sparse ANOVA selection;

   this reduces the number of model coefficients, $a_j \geq 0$ from stage 2 by a 1-norm imposed on the additive model coefficients. The solution to a quadratic loss function is then given as :

$$\begin{aligned} &\min_a \parallel y - \Phi a \parallel_2^2 + \lambda \parallel a \parallel_1, \\ &= \min_a a^T \Phi^T \Phi a + (\lambda \vec{1} - 2y^T \Phi^T) a \quad \text{subject to } a_i \geq 0 \end{aligned} \tag{5.11}$$

   equation and for the case of the hinge loss function by :

$$\min_a \parallel y - \Phi a \parallel_{1,h} + \lambda \parallel a \parallel_1 \tag{5.12}$$

$$\min_a \begin{bmatrix} \lambda \vec{1} \\ L(+1) \\ L(-1) \end{bmatrix}^T \begin{bmatrix} a \\ \xi \\ \xi^* \end{bmatrix} \text{ subject to } \begin{bmatrix} -diag(y^+)\Phi & -I & 0 \\ -diag(y^-)\Phi & -I & 0 \\ -I & 0 & 0 \\ 0 & -I & 0 \\ 0 & 0 & -I \end{bmatrix} \begin{bmatrix} a \\ \xi \\ \xi^* \end{bmatrix} \leq \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

   where y_i is the target, Φ is the ANOVA basis matrix obtained, λ is the structural regulariser and $\rightarrow$ denotes a vector of size 2^N and ξ and ξ^* is the slack variable that measures the distance of a point from the optimal hyperplane corresponding to its respective class (i.e. $max(1 - y_i \Phi a, 0)$). Hence providing interpretability through the additive kernel function.

4. Parameter selection

   using only those coefficients selected in stage 3, reconstruct a new model using.

$$f(\mathbf{x}) = \text{Sign}(\sum_{i=1}^{\ell} \alpha_i y_i \sum_{j}^{m} a_j K_j(\mathbf{x}_j.\mathbf{x})) \tag{5.13}$$

5.6 Summary

Class posterior probability and model structure are two different ways to provide interpretability to a classification model. Several model structures used in classification such as CART and graphical models have been described. However, such an approach requires a good estimate of the model's posterior density leading to the

curse of dimensionality. A way to deal with this problem is to use the SLT. This is the basis of the ANOVA kernel used in SVM which decomposes the structure model. The conventional Support Vector kernel uses the full kernel and has a complex interpretation. The sparseness in the kernel is enforced in the kernel coefficients and can be obtained using the p-norm on the coefficients, hence enforcing sparseness of the model structure which is the kernel in the feature space. Furthermore, the contraction of the ANOVA kernel has a constraint where all the univariates must pass through its origin which also means that all other higher order terms are constrained to be zero along their axes. As such, this favours selection of the lower order terms. Not all kernels fit nicely into these frameworks. The spline kernel is flexible in approximating arbitrary functions and has no scale parameter and is a good choice. The SUPANOVA was then developed for decomposing the model structure of the kernel in SVM for regression. In this thesis, we extend its use for classification of imbalanced data using 4 stages similar to that of regression except that the loss function is altered to a hinge loss function and the appropriate misclassification cost with the training and testing target distribution is incorporated. The performance selection is also altered based on the geometric mean which is less sensitive to big differences between the classification rate between each class. This approach provides an enforced sparseness of the kernels in the feature space to provide model structure interpretability. As such, the important input features distinguishing between classes can be recovered.

Chapter 6

Data Analysis

The importance of understanding why fatigue crack initiation occurs in component materials used in the automotive industry has been highlighted in Chapter 2. In this thesis, we investigate fatigue crack initiation in automotive camshafts and plain journal bearing linings. The problem that we first encounter is that of a small set of data which is imbalanced. Difficulties involved in dealing with imbalanced data and how to deal with these difficulties are looked at in Chapters 3 and 4. In order to understand the model selected, Chapter 5 describes the decomposed model produced using SUPANOVA. This chapter is divided into two main sections dedicated to the analysis of each set of data. With the camshaft data, we attempt to use several SVM extension techniques to deal with imbalanced data. Then, by using this model (i.e. the best model), we attempt to provide model interpretability. The features selected are then compared with the metallurgists' understanding of the mechanics of the system. The same approach is then applied to the Al-Si-Sn plain journal bearing lining fatigue data.

6.1 Automotive Camshaft Material - Austempered Ductile Iron (ADI)

6.1.1 Model Specification

The ADI materials data set for the automotive camshaft application contains a total of 2923 examples of which 116 samples are crack initiation sites ("Crack" class) while 2807 samples did not act as crack initiation sites ("No Crack" class). These data were obtained from a FBT of ADI which has been described in chapter 2.2. A set of

nine measurements relating to the spatial distributions and measures of the object (graphite nodules) were obtained from the tessellation. This set of nine features describes the prior domain knowledge of the microstructural distribution, e.g. the morphology of the secondary particles. The features measured for each graphite nodule which are used for learning are generated from the following measurements:

1. Object area, (O.A)
2. Object aspect ratio, ($O.A_r$)
3. Object angle, (O.Ang)
4. Cell area surrounding the object, (C.A)
5. Cell aspect ratio, ($C.A_r$)
6. Cell angle, (C.Ang)
7. Local area fraction, (L.A.F)
8. number of near neighbours, (N.N.N)
9. nearest neighbour distance, (d_{Min})
10. mean near neighbour distance, (d_{Mean})
11. nearest neighbour angle, (N.N.Ang)

See also Fig. 2.3 from chapter 2.

Prior to using the different approaches to classify the graphite nodules, the input features are normalised. This will ensure that the input feature is restricted to a unit domain and so provides no bias for each feature. Here we normalise the data to be between 0 and 1. Upon normalising, the data is ready to be partitioned into training and testing sets. The emphasis here is to use at least 75% of the "crack" class for training as it is the minority class and understanding why cracks are initiated is our main interest. Due to the extensive analysis time required to compute for a large set of data, 700 samples from the "no crack" class were randomly selected for the classification exercise. As such a set of imbalanced data with "crack" samples = 90, and "no crack" samples = 700 were used as our training sets. The rest of the data from both classes are then used for testing. The selection of the training set in each case is then repeated five times with random selection of the data each time. This was designed to assess the effect of data selection on the models produced. The average Geometric mean (Gmean) was then used to assess the overall performance of each technique and the Gmean variance is used to measure the confidence in the model selected as it reflects the dependency of the model on the data set selected for training and testing.

6.1.2 Classical Approach Results

Work done previously by (Hockley *et al.* 1999) uses simple averaging techniques (i.e. comparison of means and standard deviations - see table 2.1) and visualisation of the histogram plots to assess the difference between crack initiating graphite nodules and those that do not act as crack initiators. We have first extended the use of the simple linear discriminant technique described in chapter 2.6 to the data. Although this is a linear model, it provides feature selection by maximising the class separation. Furthermore, a linear classifier is less sensitive to noisy data and no complicated optimisation is required as discussed in chapter 3.5.2. Table 6.1 shows the results obtained from the Fisher Linear Discriminant (FLD) features using all nine features and just three features (the O.A, x_1, the C.A, x_4 and the L.A.F x_5). These three features were selected as important by prior analysis based on simple averaging techniques by Hockley et al. The results show that the FLD model is biased in both cases towards the "no crack" class (i.e. the classification rate is dominated by the "no crack" class). Although, we have identified the successful prediction of the "crack" class as being important, we also need to consider the tradeoff for the "No Crack" class. For example, the model is useless if it can classify 99% of the "Crack" class correctly but only 1% classification for the "no crack" class. Our target was set through discussion with the metallurgists as achieving a successful classification rate of at least 70% for both classes. The geometric mean provides a more suitable measurement of successful prediction for which the performance in predicting both classes is high only when they are reasonably equally well predicted.

Table 6.1 also demonstrates that using the Arithmetic mean (Amean) technique for measuring classification performance for imbalanced data is inappropriate. The Amean technique does not reflect the difference between the classification rate of the "crack" class (TP) and the "no crack" class (TN)(e.g. the Amean for using all the 9 features and using the 3 features are fairly similar but the difference between both classes' classification rate using all 9 features is 0.51 (i.e. 0.88-0.37) while for the 3 features is 0.77 (i.e. 0.99-0.22)). Therefore the use of Gmean clearly shows that

the results from using all 9 features gives a better representation of the skewness in prediction between the two classes. Lastly, the results show that using more features could enable us to obtain a better classification performance based on either Amean or Gmean. This shows that the 9 features contain more information than just using the three features identified by simple inspection, although the Gmean variance is higher than when using the 3 features.

No. of Feature	TP	TN	Amean	Gmean	Gmean(variance)
F9, All	0.37	0.88	0.62	0.57	0.0412
F3, x_1, x_4, x_5	0.22	0.99	0.61	0.49	0.0312

Table 6.1: Result from Fisher Linear Discriminant (FLD). TP and TN denote the true classification rate for the "crack" and "No Crack" class respectively. This model is biased towards the TN class, the Gmean is less sensitive to a skew distribution of the classification rate and it can be seen that using all nine features obtained a less skewed result and a better overall classification rate.

Linear Discriminant analysis provides an initial statistically based analysis prior to generalising with more complex nonlinear modelling approaches. The linear approach may miss key features of the data which can only be represented using nonlinear approaches. Furthermore, the FLD provides a description of the data rather than predicting unseen data that might also be useful. Last but not least, it requires density estimation of the input which may not be appropriate for a given limited number of data, as described in chapters 2.6 and 3.1. As such, a non-linear approach known as SVM has been investigated. SVM have gained success in recent years for many classification and regression problems (Burges 1998, Smola 1998). SVM was developed from the *"Statistical Learning Theory"* (SLT) which was thought to effectively describe statistical estimation with small samples (Cherkassky & Mulier 1998). Another important feature of SVM is its substitution of the kernels. This eliminates the problem of the input dimensionality that FLD has. Next, we will describe the results obtained using the SVM and its extension techniques described in chapter 4.3.

6.2 SVM Results

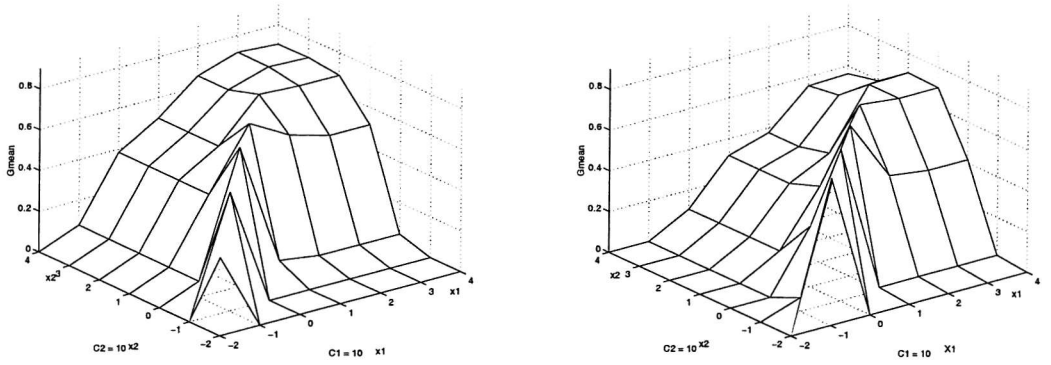
Our initial approach was to use the standard SVM, Adaptive Margin (AM) SVM and SVM for regression to deal with the imbalanced data set. The results are summarised in Table 6.2. Two kernels, namely the spline and Radial Basis Function (RBF) were used. Spline kernels were used because of their flexibility and there is no free parameter to be determined. Furthermore, it can be easily incorporated with a parameter to enforce sparseness in the model interpretability (as discussed in chapter 5.3). On the other hand, RBF kernels require the width (σ) to be tuned. The capacity Control, C, was sampled logarithmically on [0.01, 10000] for both classes (i.e. the "crack" and "no crack" class). The results from the standard SVM show that with C of both classes allowed to be rather high (i.e. C^+ and C^- are 1000) the spline kernel obtains a Gmean classification rate of 0.58 with a variance of 0.0312. On the other hand, the RBF with σ=0.5 obtains a similar Gmean of 0.59 with a variance of 0.0354. We have used σ = 0.1,0.5,1.0 and see that σ =0.5 gives the best results. Again, the C is rather high here, indicating that a large amount of smoothness is required.

Classification Performance		TP	TN	GMean
Approaches	Kernels			(Variance)
FLD	-	0.37	0.88	0.57 0.0412
SVM	Spline	0.36 $C^+ = 1000$	0.94 $C^- = 1000$	0.58 0.0312
	RBF (σ=0.5)	0.39 $C^+ = 1000$	0.90 $C^- = 1000$	0.59 0.0354
AMSVM	Spline	0.32 $\lambda = 0$	0.91 $\lambda = 0$	0.54 0.0241
	RBF (0.5)	0.32 $\lambda = 0$	0.92 $\lambda = 0$	0.53 0.0416
Regression SVM	Spline	0.38 $C^+ = 10000$	0.91 $C^- = 10000$	0.59 0.0158

Table 6.2: Summary of the best test results obtained by averaging the set of five random data set selection samples with Fisher Linear Discriminant (FLD) techniques and standard SVM with various extension techniques for the imbalanced data set. TP and TN are the true classification rate for "crack" and "no crack" classes respectively.

Rather than keeping the margin fixed in the classical SVM, we make the margin automatically adapt to its data. The margin can be relaxed by having a regularising parameter λ and this was varied from 0 to 10. This was the approach taken by

Adaptive Margin (AM) SVM, also known as Generalised LOO SVM described in chapter 4.3.3. The results indicate that no smoothness (i.e. λ=0) is necessary in order to obtain a good solution. This approach did not provide enough flexibility for imbalanced data as it only gave a Gmean of 0.54. The advantage of this approach is that the optimal parameter (λ) can be obtained from the training set alone (i.e it is the generalised LOO method). Next, in the regression SVM case, we allow the target y to be a step function and alter the model selection for a regression task (i.e. mean squared error) to the misclassification cost. This provides a quick way of converting the regression problem in SVM into a classification problem. A quadratic loss function (see chapter 3.8.2) is used in this case as there are no parameters to be tuned. However, a sparse representation is no longer available. A fairly similar result was obtained with that of the standard SVM but the variance is lower with a higher regulariser required (i.e. C's = 10000). This may be due to the loss function and because the model selection reflects a different cost. No attempts were made to use the RBF as the above results show similar performance, and also it is an extra parameter requiring tuning making it computationally less efficient. In summary, our initial attempts show that all the approaches (FLD and SVM) have a bias towards the "no crack" class with a classification success rate of at least 0.88, whereas the "crack" class has a far lower classification success rate of 0.32-0.39. Also, a high capacity (i.e. large smoothness) is required for the case when the margins are fixed (i.e. standard SVM and SVM for regression). However, when the margin is automatically adapted, it shows that no smoothness is required to obtain a good solution. This may show the flexibility of the adaptive margin approach. The variance of the FLD on Gmean is higher than that of the SVM approaches, although the average Gmean values are comparable. This leaves doubts as to the stability of the classification based on classical FLD as it appears to be more susceptible to data set selection than the SVM approaches. The testing results for modified SVM approaches dealing with imbalanced data are summarised in Table 6.3. Our first attempt to deal with imbalanced data is to make it almost balanced. As such, we further downsampled the majority class (i.e. "no crack") from 700 to 120: Note: we



(a) Balanced data set (b) Imbalanced data set

Figure 6.1: The plots of the GMean result of the CSSVM result used for balanced and imbalanced data. C1 and C2 denote the "crack" and "no crack" class capacity control and x1 and x2 denotes the order of the C's value. The plots show that for the case of imbalanced data, the capacity control has to be penalised differently in order to obtain good results. This is demonstrated by the higher value of the Gmean for the balanced case lying on the diagonal axis while for the imbalanced case it is off the diagonal axis.

simply took the first 120 from the 700 previously randomly selected data. The set of C's used are the same as in the previous set. Results have shown improvement, increasing the Gmean result by at least 0.09. Also, the variance is greatly improved. However, the results obtained between the 2 kernels are very different (in terms of both classification rate and capacity control) compared to using the imbalanced data set. Identifying the kernel to be used is therefore an important issue in this approach. Although the results of the reduced set of data provide good results, the classification rate for the TP (crack) is less than 0.70 which is less than we have identified as a good classification rate (through discussion with the fatigue experts who provided the original data). Furthermore, the importance of the kernel used is highlighted therefore making it another tuning parameter to be considered. Next, we attempt to use the CS SVM (described in chapter 4.3.1) on imbalanced data in order to impose different misclassification costs. The capacity control is now two dimensional rather than one dimensional (i.e we have capacity control sampled logarithmically on $[0.01, 10000]^2$, producing 49 models for selection). The Gmean was observed to have out-performed the SVM using balanced data and the variance is also smaller (i.e we are more confident in our model which is less susceptible to data set selection) with both classification rates well above 0.70. We observe that

the ratio between the misclassification cost is 10 times (i.e. $\frac{C^+}{C^-} = \frac{1}{0.1} = 10$. This could coincide with the ratio of the data sets used (i.e. $\frac{crackclass}{nocrackclass} = \frac{700}{90} \approx 8$). Figures 6.1 a and b, show 2D plots of the Gmean results for the balanced and imbalanced data set. It can be seen that for the case of the balanced data set, the highest value of Gmean lies along the diagonal axis (i.e. $C^+ = C^-$) while for the case of the imbalanced data this shifts towards the crack class.

The use of the spline kernel and the RBF (σ=0.5) kernels for the case of imbalanced data seem to provide similar solutions. As such, we will concentrate on using the spline alone. It will become clearer that the use of this kernel provides a further advantage when dealing with model interpretability as described in chapter 5.3, which has formed the basis for SUPANOVA. The analogue of using two different C's is derived by imposing a different misclassification cost. A misclassification cost ratio of 8 seems to be unrealistic compared with values quoted in the literature of 2-5 at most. The Non-Standard Situation (NSS) SVM approach provides a natural way of dealing with imbalanced data. It incorporates the imbalanced data into 2 important components : the misclassification cost (MC) and sampling bias (π) as described in chapter 4.1 and 4.3.2. The associated values for this approach used in this case can be shown as :

$$\pi_{tr}^- = \frac{700}{790} \qquad \pi_{tr}^+ = \frac{90}{790} \tag{6.1}$$

$$\pi_{te}^- = \frac{2107}{2133} \qquad \pi_{te}^+ = \frac{26}{2133} \tag{6.2}$$

$$\begin{aligned} L(+1) &= MC^- \pi_{tr}^- \pi_{te}^+ \\ &= 0.0108 MC^- \end{aligned} \tag{6.3}$$

$$\begin{aligned} L(-1) &= MC^+ \pi_{tr}^+ \pi_{te}^- \\ &= 0.1125 MC^+ \end{aligned} \tag{6.4}$$

substituting the imbalanced modified factor L's for the case of standard SVM (see Eq. 5.9) and assuming the smoothness parameter to be ($\lambda = 10^{-5}$), the C of the

SVM is modified to :

$$\begin{aligned} C^{+} &= \frac{L(-1)}{2n\lambda} \\ &= \frac{0.1125MC^{+}}{2(790)(10^{-5})} \\ &= 7.12MC^{+} \end{aligned} \tag{6.5}$$

$$\begin{aligned} C^{-} &= \frac{L(+1)}{2n\lambda} \\ &= \frac{0.0108MC^{-}}{2(790)(10^{-5})} \\ &= 0.69MC^{-} \end{aligned} \tag{6.6}$$

Notice that the magnitude of the ratio between the capacity control is controlled by λ, a smoothness parameter.

The MC for the crack class was varied between [1 and 2] and the λ is varied between $[10^{-4}, 10^{-5}, 10^{-6}]$. Our results show that with no heavy MC imposed on the "crack" class (i.e. using only the factor of sampling bias), a Gmean of 0.74 with variance of 0.0106 and the classification rate of both classes are at least 0.74. When the misclassification cost of the "crack" class was directly imposed twice, the classification rate of the crack class increased but the Gmean is reduced. As for the case of the varying λ (which is the regulariser parameter in the RN, see eq. 4.10), we find that there is no significant effect on their results. However, the results shown here are based on $\lambda = 10^{-5}$. The idea of imposing MC and sampling bias was extended to the AM SVM and regression SVM. The results from the extended AM SVM indicate that the LOO SVM (i.e. λ=1) provides a good solution. However, a slight increase in allowing generalisation for the crack class (i.e. λ=4) improved the Gmean and its variance. The extended regression also performed much better than the original regression SVM with a Gmean of 0.72.

In conclusion, the above results show that most of our extended approaches do provide a reasonably good Gmean performance. Therefore, for imbalanced data, it

Classification Performance		TP	TN	GMean (Variance)
Approaches	Kernels			
SVM balanced data	Spline	0.68 $C^+ = 1000$	0.76 $C^- = 1000$	0.72 0.0107
	RBF (σ=0.5)	0.56 $C^+ = 1$	0.83 $C^- = 1$	0.68 0.0187
CSSVM With L1-norm Error	Spline	0.71 $C^+ = 1$	0.78 $C^- = 0.1$	0.74 0.0078
	RBF (σ=0.5)	0.76 $C^+ = 1$	0.79 $C^- = 0.1$	0.75 0.0040
NSS SVM	Spline ($\lambda = 10^{-5}$,MC=1)	0.74 $C^+ = 7.12\text{x}MC$	0.75 $C^- = 0.68$	0.74 0.0106
	Spline ($\lambda = 10^{-5}$,MC=2)	0.85 $C^+ = 7.12\text{x}MC$	0.60 $C^- = 0.68$	0.71 0.0071
Extended AMSVM	Spline ($\lambda = 1$,MC=1)	0.74 $C^+ = 7.12\text{x}MC$	0.73 $C^- = 0.68$	0.73 0.0069
	Spline ($\lambda = 4$,MC=1)	0.76 $C^+ = 7.12\text{x}MC$	0.73 $C^- = 0.68$	0.75 0.0036
Extended SVM (Reg)	Spline (MC=1)	0.71 $C^+ = 7.12\text{x}MC$	0.72 $C^- = 0.68$	0.72 0.0078

Table 6.3: Summary of the best test results by averaging the results of five random selection data set samples using different techniques to handle the problems of imbalanced data. TP and TN are the true classification rates for "crack" and "no crack" classes respectively.

is necessary to incorporate the sampling bias and if necessary, a higher misclassification cost for the minority class. The ratio of the factors for imbalanced data for CS SVM for the "crack" class and the "no crack" class is 10 while in NSS SVM it is 10.47 (i.e. $\frac{7.12}{0.68}$). They are fairly similar (however, a good result requires fine tuning) and are the key factors required for this imbalanced set of data. The development of the NSS SVM may provide a rough guide to the ratio of the factors for imbalanced data. In this instance, it has been shown that the heuristic approach of the CS SVM outperforms the NSS SVM with its somewhat lower variance for the GMean. As such, it was this model that we used as a basis for further work although it must be acknowledged that the decision has been made based on slight differences only) . This model structure was decomposed for model interpretation purposes. It is noteworthy that although the modified AM SVM can be used, it is however computationally expensive (i.e. with the kernels involved in the constraint

Eq. 4.18).

6.2.1 Results and Discussion for Model Interpretability

The previous attempts using a set of five randomly selected data set samples provided an assessment of which is the more appropriate approach for imbalanced data. Having identified the most acceptable performance (CS SVM) (i.e. rating the factors for taking imbalanced data into account (associated with misclassification cost and the sampling bias)), the SUPANOVA approach was then used on the CS SVM to generate model interpretability. In this case, we now increase the number of randomly selected training data sets to ten. The parameters that we use for the model structural regulariser λ are set in the range of [0.05,0.1-1] with increments of 0.1. The number of input components selected as represented was based on its occurrence more than five times out of the ten in the models generated on the randomly selected sample sets. In this approach, we use the hinge loss function which provides a more natural way of dealing with classification as the loss function for the MC and the model selection are the same. Our initial work (Lee *et al.* 2001*c*) did not incorporate the modification factor L into the ANOVA basis (Eq. 5.12). Thus, there is an inconsistency in the loss function between the CS SVM and the SUPANOVA. As such, a better result is obtained when both loss functions are consistent. The complete description of the SUPANOVA for imbalanced data classification can be found in chapter 5.5. By using the full kernel, we obtain a Gmean of 0.74 and a variance of 0.0202. An increase in λ reduces the number of components selected. However the Gmean variance increases. The results show that when $\lambda = 0.4$, a sum of 6 subcomponents provide a reasonable Gmean and Gmean variance. With the increase values of λ, the number of subcomponents selected reduced to only one and the Gmean results are bad. Tables 6.4 and 6.5 show the summary of the modelling results and the input components selected by the SUPANOVA for the ADI material.

The plots from Fig. 6.2 show the trends of the 6 components obtained from our SUPANOVA model and are described as follows :

Approaches	TP	TN	GMean Variance	Components
SUPANOVA Classifications	0.72 $C^+ = 1.0$	0.77 $C^- = 0.1$	0.74 0.0202	512 λ=0
	0.71 $C^+ = 1.0$	0.76 $C^- = 0.1$	0.73 0.0247	13 λ=0.05
	0.70 $C^+ = 1.0$	0.77 $C^- = 0.1$	0.72 0.0282	6 λ=0.4

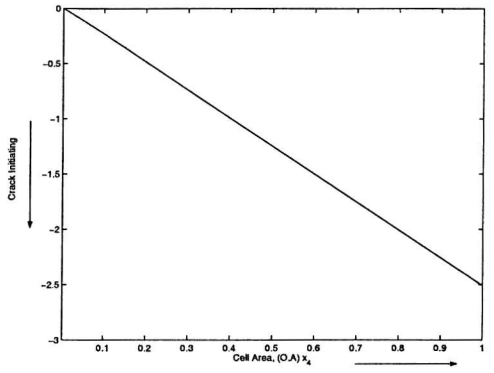
Table 6.4: Summary of results from SUPANOVA. These results are based on averaging 10 randomly sampled data sets and the number of input components identified are based on occurrence more than 5 times out of 10. Note: the λ here are used to enforce sparseness of the components rather than acting as a regulariser parameter as in SVM.

Components	Occurrence	Consistency	Remarks
bias	10	YES	-
C.A, x_4	10	YES	As C.A increases, cracks are likely to initiate
L.A.F, x_5	10	YES	As L.A.F increases, cracks are likely to initiate
N.N.N, x_6	6	YES	As N.N.N increases, cracks are likely to initiate
O.Ang $\bigotimes$ d_{min} $x_3 \bigotimes x_7$	8	YES	As both components increase, cracks are unlikely to initiate
d_{Mean} $\bigotimes$ N.N.Ang $x_8 \bigotimes x_9$	6	YES	As both components increase to a threshold, cracks start to initiate

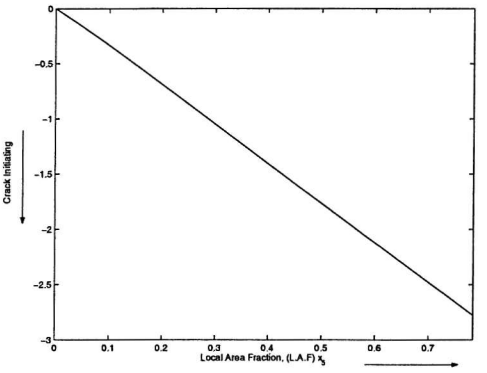
Table 6.5: SUPANOVA components selected and their occurrence in the classification task. $\bigotimes$ denotes Tensor product. "Consistency" refers to similar trends observed in the SUPANOVA terms.

- Univariate: Large cell area (C.A) (100% selection), high local area fraction (L.A.F) (100% selection) and a large number of near neighbours (N.N.N) (60% selection) are all shown to identify a graphite nodule that initiates a fatigue crack.

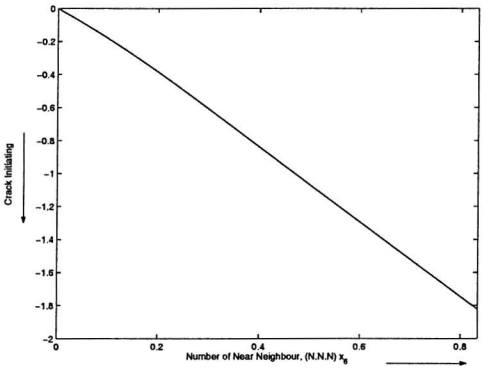
  This can be interpreted as large graphite nodules of high L.A.F (i.e. with local clustering from a lot of N.N.N) acting as fatigue initiation sites. The fact that the classification has identified C.A rather than object area (O.A) explicitly is intriguing. Due to the FBT process, the cell area (C.A) is directly linked to the O.A (as the cell is defined as always larger than the O.A). The condition that both increased C.A and L.A.F (i.e. $\frac{O.A}{C.A}$) give rise to preferential crack initiation can be satisfied by considering a large O.A as identifying a crack



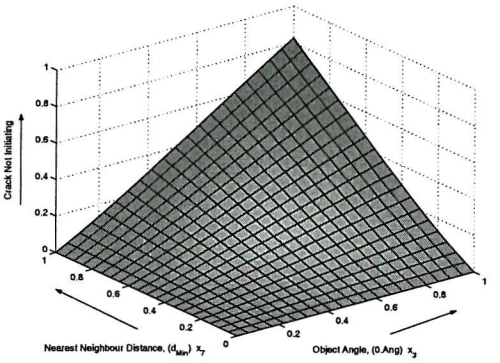
(a) Cell Area



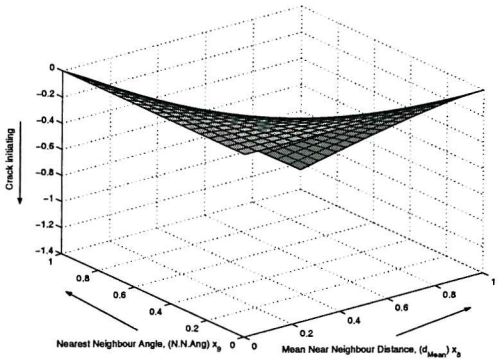
(b) Local Area Fraction



(c) Number of Near Neighbour



(d) Object Angle and Nearest Neighbour Distance.



(e) Mean Near Neighbour Distance and Nearest Neighbour Angle.

Components	Frequency
Crack	
C.A (high)	10/10
L.A.F (high)	10/10
N.N.N (high)	6/10
N.N.Ang (high) and d_{Mean} (high)	6/10
No Crack	
d_{Min} (high) and O.Ang (high)	8/10

Summary table

Figure 6.2: An example of plots with the components selected versus the output of SUPANOVA for classification with imbalanced data. Bias and 5 other components being selected as significant factors causing fatigue crack initiation. The tessellation measurements (already normalised) form the x-axis and x-y axes, whilst on the y-axis or z axis, the scales values act as an indicator of crack initiation (i.e a negative value denotes a crack initiation and positive value denotes a crack not initiating).

initiating nodule. The interdependency of the features measured by the FBT is a function of the geometry of 2 phase microstructures. In real life cases many of these input parameters vary and despite the imposed transparency

afforded by the SUPANOVA decomposition, some ambiguity as to the key features causing fatigue initiation remains. As detailed later in chapter 7, this can be further explored by producing simulated microstructures where the input features can be varied in a more systematic way and the resultant predictions produced by the classification models assessed. So, in considering the univariate, a reasonable initial interpretation would be that large graphite nodules of high L.A.F (i.e. with local clustering provided by a larger than average N.N.N) will act as crack initiators.

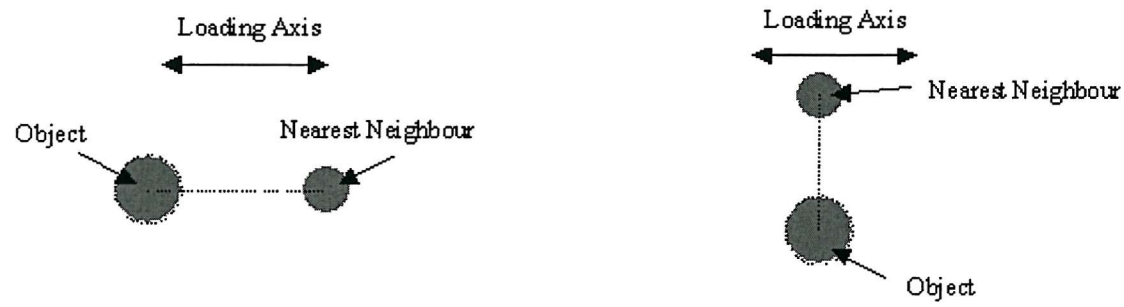
- Bivariate: Two bivariates have been identified: Object Angle (O.Ang) and nearest neighbour distance (d_{min}) (80% selected) and Mean near neighbour distance (d_{mean}) and nearest neighbour angle (N.N.Ang) (60% selected) (See Fig. 6.2d and e).

  The O.Ang defines the angle between the object major axis and the loading axis, the larger the O.Ang the closer the object's major axis is to perpendicular to the loading axis (see Fig. 6.3). In the case of the graphite nodules the aspect ratio is close to 1 (i.e. the mean and standard deviations for "crack" class is 1.30 and 0.28, and for "no crack" class is 1.40 and 0.38 respectively) as the nodules are reasonably spherical, so a link to O.Ang is initially surprising. Considering the bivariate dependence between O.Ang and d_{min} it appears that for a far away nearest neighbour (N.N), a high O.Ang (i.e. object major axis perpendicular to the loading axis) leads to a graphite nodule not initiating a crack.

Figure 6.3: The alignment between the object angle and the loading axis.

The N.N.Ang is defined as the angle between the loading axis and the line connecting the centre of the N.N object of interest. Thus, if N.N.Ang is

high the N.N lies perpendicular to the load axis, and if the N.N.Ang is low, the N.N is parallel to the object of interest in line with the loading axis - as shown in Figure 6.4. Now, considering the bivariate dependence between d_{mean} and N.N.Ang, it appears that with increasing d_{mean} (i.e. further apart near neighbours) and increase in N.N.Ang (i.e. the N.N. particle becoming closer to being perpendicularly aligned to the object of interest) leads to an increased likelihood of the graphite nodule initiating fatigue.



a.) Nearest Neighbour Angle Small

b.) Nearest Neighbour Angle Large

Figure 6.4: The alignment of the nearest neighbour angle with respect to the object and its loading axis.

The d_{mean} and d_{min} relate to the spacing between the graphite nodule interfaces - averaged for all near neighbours and for the N.N respectively. The d_{mean} value depends on many factors, including N.N and their spacing, and can be considered to reflect clustering (although not unambiguously), and a high d_{mean} may be considered to reflect a relatively unclustered situation, which may allow the positioning of the N.N. to be more influential in affecting the central graphite nodule. The N.N particle appears therefore to have two possible effects (from inspection of the SUPANOVA trends) if it is far away and the central graphite nodule's major axis is perpendicular to the loading axis, then the nodule is not likely to initiate a fatigue crack. If the average N.N spacing is high (again, relatively unclustered) then the N.N particles positioning may be more influential and if it is perpendicularly aligned above or below the central graphite nodule, this appears to promote cracking.

Attempting to consider all these variables, it seems that large nodules in a locally clustered environment with many near neighbours are likely to initiate cracks. If the situation is less clustered, then a N.N particle aligned perpendicularly above the central graphite nodule may promote cracking. If the O.Ang is perpendicular to the loading axis and yet the N.N is distant then the graphite nodule is unlikely to initiate a fatigue crack.

We can try to explain these trends in the following way. The graphite nodules have a significantly lower effective Young's modulus than the surrounding matrix, decohere easily and may be considered to act as holes in a mechanical sense. The predominantly spherical nature of the nodules indicates that size increase will not increase the local stress concentration factor, although the larger graphite nodules will give a larger sampling volume of potential initiation points. Local clustering around such larger graphite nodules (as identified by the classifier) may be expected to superimpose local particle stress fields, raising the peak stress levels. The more complex bivariate relationships are somewhat harder to assess. The O.Ang defines the angle between the loading axis and the major axis of the nodule and if this is high the major axis of the nodule is closer to perpendicular to the tensile axis (which might be expected to promote cracking). However this, combined with a relatively furthest N.N might be expected to minimise superimposition of local particle stress fields, and hence make these nodules less likely to act as crack initiation sites. Given the low aspect ratio of the nodules (they are effectively spherical) correlations with O.Ang are surprising. When the near neighbours are relatively far away or fewer in number, then the positioning of the N.N appears important, with a perpendicularly oriented N.N making a nodule more likely to crack. The cracks initiate perpendicular to the loading axis and so superposition of local stress fields between the N.N in a perpendicular orientation may promote cracking. The two univariate components (i.e. the C.A and L.A.F) tally with the finding of (Hockley *et al.* 1999). Here, we have picked up an extra component, that is the N.N.N which may be an important component. With SUPANOVA, we are also able to pick up higher order interactions that are not easily identifiable through simple means of

visualisation. These interpretable classification results allow us to start to assess the relationships that give rise to crack initiation and hence eventually to identify optimised microstructures with good fatigue resistance for the camshaft application. Further assessment of these results are considered in chapter 7 - where the model predictions for simulated particle distributions are examined. an analysis of scales (and distributions in scale) would be valuable, along with comparison against various mechanical analyses that exist of 2 phase materials, however, a thorough investigation of these points is beyond the scope of the current work.

6.3 Automotive Plain Journal Bearing Lining Material - Aluminium-Silicon-Tin (Al-Si-Sn)

6.3.1 Model Specification

The plain journal bearing lining fatigue initiation assessment used a total of 10 observation regions which were selected randomly from microstructure containing crack initiation sites as discussed in section 2.3. The total number of cell was 2938, with silicon (Si) being identified as the primary initiating phase. The cells produced by the FBT were initially divided into three populations: initiating Cells (with 163 cases); bordering Cells (with 810 cases surrounding the initiating sites) and background Cells (with 1965 cases). Results by (Joyce 2001) comparing the mean and standard deviation show that there is little significant difference between the bordering and background cells (see table 2.2). As such, the bordering and the background cells are considered here to belong to the same class. Once again, we have a two class classification problem as in the case of the previous camshaft application. We now have a total of 2938 examples of which 163 samples are crack initiation sites ("Crack" class) while 2775 samples do not act as crack initiation sites ("No Crack" class). There are altogether 11 features used for this example (as seen in Table 2.2). The two extra features extracted from the FBT are the Cell Aspect Ratio ($C.A_r$) and Cell Angle (C.Ang). In this application, the tensile axis was vertical, so a large O.Ang corresponds to particles aligned parallel to the loading axis. Note: this difference for the case of ADI, where the tensile axis was horizontal.

The data are normalised between 0 and 1. The data are partitioned into the training and testing sets consisting of 1200 ("no crack") and 120 ("crack") data in the training sets, and the rest are used for the testing set. These are randomly selected and the data set partitioning was repeated 10 times to provide good generalisation for our model. We have again chosen to use 75% of the crack data for training. The reason for using more of the data collected for the "no crack" class compared with the ADI application (1200 compared to 700) is because we are combining two populations in the Al-Si-Sn case (background and bordering). If we randomly took 700 samples, then they might predominantly come from either class. Hence the need to ensure reasonable representation from both classes. Again, the average Gmean (based on averaging 10 models) was then used to assess the overall performance of each technique and the Gmean variance is used to measure the confidence in the model selected as it reflects the dependency of the classification on the data set selected for training and testing.

6.3.2 Results and Discussion

The previous application of the CS SVM and the NSS SVM on ADI showed good prediction results. These approaches were then extended to this new set of data (Al-Si-Sn). A set of capacity controls similar to those used in the ADI case were used with the spline kernel. The results show that with ($C^+ = 10$, $C^- = 1$) a GMean of 0.72 was obtained. A set of regularisers $\lambda = [10^{-4}, 10^{-5}, 10^{-6}]$ with $MC^+ = [1, 2, 3, 4, 5]$ were used in the NSS SVM. The imbalanced factors L were calculated and the corresponding capacity control obtained as $C^+ = 3.35MC^+$ and $C^- = 0.92MC^-$ for the case when $\lambda = 10^{-5}$. Table 6.6 summaries the results of the CS and NSS SVM. Results show that a misclassification cost of 3 is required to be imposed on the "crack" class in order to obtain a Gmean of 0.70. The imbalanced modification factor for the NSS SVM is 10.92 while for the CS SVM is 10. The variance obtained is similar to that of the CS SVM. As such, the CS SVM model was again selected for the SUPANOVA to provide model interpretability. In both applications we have assessed, we have shown that the NSS SVM can provide a rough guide for the value of the imbalance modification factor L. However, fine

tuning may be required. The model structure regulariser λ we obtained here varied from 0-20 and results show that $\lambda = 15$ yields the best result as shown in Table 6.7. A set of 6 components were obtained with reduced Gmean of 0.70 compared to 0.72, but with a slightly lower Gmean variance. The fact that a higher model regulariser is required is due to the fact that the capacity control used here is larger than the case of the ADI (i.e. ADI - $[C^+ = 1.0, C^- = 0.1]$ and Al-Si-Sn - $[C^+ = 10, C^- = 1.0]$). The components picked up and consistency of the input components being selected are presented in table 6.8.

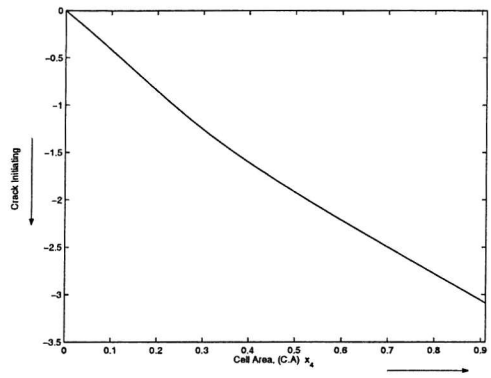
Approaches	TP	TN	GMean Variance
CS SVM	0.73 $C^+ = 10$	0.71 $C^- = 1$	0.72 0.0185
NSS SVM MC=1	0.27 $C^+ = 3.35$xMC	0.87 $C^- = 0.92$	0.40 0.0439
NSS SVM MC=3	0.69 $C^+ = 3.35$xMC	0.70 $C^- = 0.92$	0.70 0.0184

Table 6.6: Summary of the test results for Al-Si-Sn results from CS and NSS SVM. This shows that a misclassification penalty of 3 must be imposed for the crack class in the NSS SVM in order to obtain a good classification. The Gmean of the CS SVM is better than the NSS SVM.

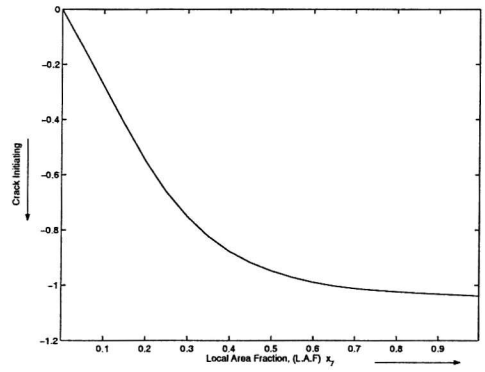
Approaches	TP	TN	GMean Variance	Components
SUPANOVA Classifications	0.73 $C^+ = 10$	0.71 $C^- = 1$	0.72 0.0185	2048 λ=0
	0.69 $C^+ = 10$	0.71 $C^- = 1$	0.70 0.0136	6 λ=15

Table 6.7: Summary of the test results for Al-Si-Sn results from SUPANOVA for classification. These results are based on averaging the predictions based on 10 randomly sampled data sets and the number of components identified are based upon occurrence more than 5 times out of 10.

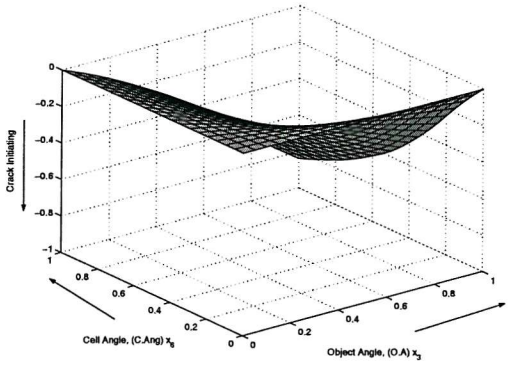
Figure 6.5, shows the plots of selected examples of the input components. The univariate plots of the C.A and L.A.F. show that as both values increase, the chances of crack initiation also increase. The occurrence of the two univariates are 100% (i.e. 10 models out of 10 selected these two components). There is a slightly different trend in one of the L.A.F plots (as shown in Fig. 6.6a) as it is a concave shape trend (lowest at 0.40). However, the C.A area trend consistently indicates that crack initiation occurs as C.A. gets larger. The bivariate plot for the O.Ang



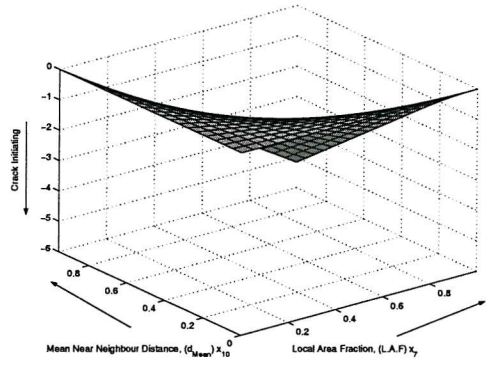
(a) Cell Area



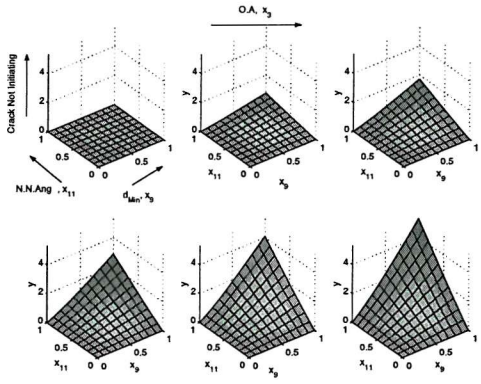
(b) Local Area Fraction



(c) Object Angle and Cell Angle



(d) Local Area Fraction and Mean Near Neighbour Distance



(e) Object Angle and Nearest Neighbour Distance and Nearest Neighbour Angle.

Components	Frequency
Crack	
C.A (high)	10/10
L.A.F (high)	10/10
N.N.N (high)	6/10
C.Ang (high) and O.Ang (high)	8/10
d_{Mean} (high) and L.A.F (high)	6/10
No Crack	
d_{Min} (high) and O.Ang (high) and N.N.Ang (high)	8/10

Summary table

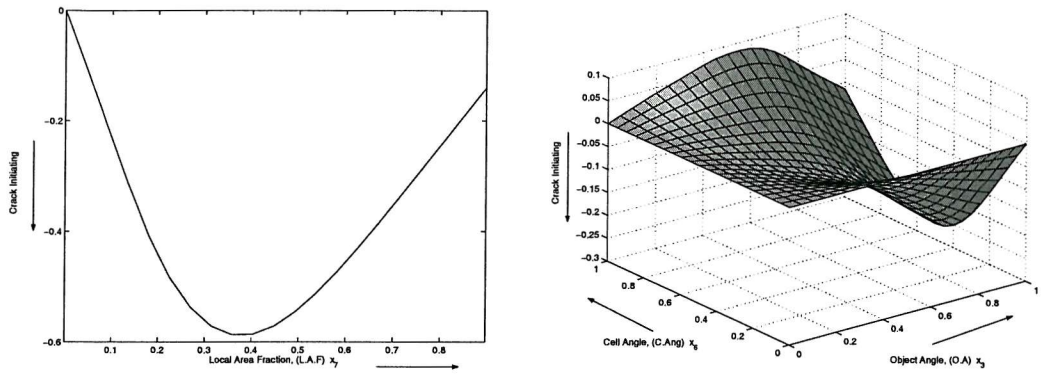
Figure 6.5: An example of plots with the input components selected versus the output SUPANOVA for classification with imbalanced data. Bias and 5 other components have been selected as significant factors causing fatigue crack initiation. The tessellation measurements (already normalised) form the x-axis and x-y axes, whilst on the y-axis or z axis, the scales values act as an indicator of crack initiation (i.e a negative value denotes a crack initiation and positive value denotes a crack not initiating).

Components	Occurrence	Consistency	Remarks
bias	10	YES	-
C.A x_4	10	YES	As C.A. increases, cracks likely to initiate
L.A.F x_7	10	NO	As L.A.F. increases, cracks likely to initiate
O.Ang $\bigotimes$ C.Ang $x_3 \bigotimes x_6$	8	NO	Simply (varying functions) difficult to explain
L.A.F $\bigotimes$ d_{mean} $x_7 \bigotimes x_{10}$	6	YES	As both components increase, cracks likely to initiate
O.Ang $\bigotimes$ d_{Min} $\bigotimes$ N.N.Ang $x_3 \bigotimes x_9 \bigotimes x_{11}$	8	YES	As the three components increase, Cracks unlikely to initiate

Table 6.8: SUPANOVA components selected, their occurrence rated out of 10 and consistency in classification task. $\bigotimes$ denotes Tensor product. "Consistency" refers to similar trends observed in the SUPANOVA terms.

and C.Ang shows a complex trend which indicates a large O.Ang and C.Ang are likely to initiate a crack. The selection of the occurrence of this component is 80%. However, one of the plots (shown in Fig. 6.6b) has a different shape but it too shows that as the O.Ang (increases independent of C.Ang), the chances of crack initiation increase. The next bivariate plot selected in Fig 6.5d is the L.A.F. vs d_{Mean}. A hyperplane of concave shape is seen along the diagonal of both increasing axes (i.e. as both features increase). This indicates that there is a threshold value (i.e. approximately 0.5 for both directions) for these two features. Beyond this threshold, cracks are more likely to initiate. The confidence about the importance of this bivariate component selected is less as it only has an occurrence of 60%. The trivariate relationship selected consists of O.Ang, d_{Min} and N.N.Ang with an occurrence of 80%. This plot simply indicates that as all three features increase, the chances of crack initiation decreases.

In summary, an increase of both C.A and L.A.F indicates that the object is large (as discussed in the ADI section). A large Si particle is more likely to cause cracks because of the increase of local matrix strain around the large hard/stiff particles. (It is noteworthy that in the Al-Si-Sn case, the initiating particles have a significantly higher stiffness than the surrounding matrix, in contrast to the ADI case.) The bivariate plot of the O.Ang and C.Ang implies that the Si particles (which have a slightly higher Mean $O.A_r$ (1.49 compared to ADI 1.40) when locally



(a) Local Area Fraction (b) Object Angle and Cell Angle

Figure 6.6: The inconsistency of the trends obtained for Local area fraction and also the Object Angle and Cell Angle as opposed to those obtained from Figure 6.5

aligned parallel to the loading axis are more likely to cause crack initiation.

The other bivariate function obtained is the L.A.F and the d_{Mean}, cracks being likely to initiate as both increase. This may describe a relatively large Si particle which is relatively isolated (i.e. large d_{Mean}, but large O.A to still give large L.A.F). There will be little stress/strain shielding from neighbouring Si particles. Therefore, cracks are perhaps more likely to be initiated if both components increase. Finally, the trivariate plot indicates that as the O.Ang, d_{Min} and N.N.Ang increase, crack initiation is unlikely (shown schematically in Fig. 6.7). The large d_{Min} can perhaps be considered to indicate a situation where any local stress/strain field overlapping between the N.N is minimised hence making crack initiation unlikely.

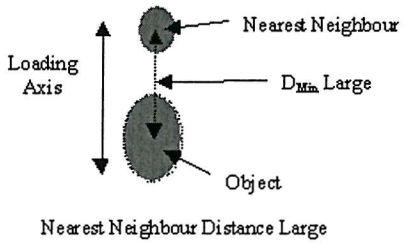


Figure 6.7: Schematic representation of the extreme of the trivariate function (i.e. O.Ang, d_{Min} and N.N.Ang) that indicates crack initiation unlikely.

Further investigation of these possibilities of crack initiation using Finite Element Analysis (FEA) simulations of idealised examples is necessary and ongoing within the Materials Research Group. It is also necessary to try and systematically vary the parameters selected by the SUPANOVA decomposition for *self-consistent*

particle distributions to assess those features of particle distributions which will give rise to increased fatigue initiation. The simulated particle distributions described in Chapter 7 have been used for both the ADI and Al-Si-Sn SUPANOVA classification models to further enhance the interpretability offered and to provide indications of more fatigue initiation resistant microstructures.

6.4 Summary

The present data can be dealt with in the standard SVM by incorporating a factor for imbalanced data that was derived from imposing different misclassification costs for each class and sampling bias. The NSS SVM provides us with a rough guide of the ratio required for the capacity control used in SVM between both classes. However, a better result can be obtained with fine tuning. The interpretability of the model was provided by decomposing the model structure. This was used in the original SUPANOVA for regression tasks. Here, we extend its use for the case of imbalanced data for classification. There were six important components selected in each data set investigated out of the possible 512 (ADI) and 2048 (Al-Si-Sn). A larger regulariser, λ, is required to obtain the smaller set of components required. Finally, the sets of components selected show qualitative correlation with known metallurgical factors as important factors which initiate fatigue cracks. These components selected are inter-related. As such, it is necessary to vary them systematically in order to enhance our understanding. This is done via the simulated particle distributions which will be described in the following chapter.

Chapter 7

Simulated Data Analysis

The important input components needed to classify fatigue initiation sites have been picked up by our SUPANOVA model. However, the components selected are inter-related and it is difficult to unambiguously determine the key particle distribution characteristics that promote failure. This chapter describes the particle simulations that were used to systematically vary the key input components within self-consistent particle distributions. The results from these simulated data sets enhance the interpretability offered by the SUPANOVA model and provide indications of more fatigue initiation resistant microstructures in both ADI and Al-Si-Sn bearing lining alloys. The first section describes the selection and justification of the simulated data sets. This is then followed by a detailed description of the procedure required for the specification of the simulated data. The descriptive enhancement offered by the simulated data for the ADI and Al-Si-Sn SUPANOVA models is investigated. Finally, the results obtained from the SUPANOVA model and the simulated distributions are described.

7.1 Selection and justification of the simulated data sets

Varying individually each component of the FBT features is a very difficult task as all the components are inter-related. However, we can vary various parameters such as the object shape, size, angle and distribution which can be fixed/varied in a systematic fashion to allow clearer visualisation of the trends identified by the SUPANOVA technique. Let us summarise the components identified for the two automotive materials selected from the SUPANOVA model. For the case of the

ADI: Cell Area (C.A), Local Area Fraction (L.A.F), Number of Near Neighbours (N.N.N), Object Angle (O.Ang) Vs Nearest Neighbour distance (d_{Min}) ; Mean Near Neighbour Distance (d_{Mean}) Vs Nearest Neighbour Angle (N.N.Ang) are selected as important components. For the Al-Si-Sn: C.A, L.A.F, O.Ang Vs Cell Angle (C.Ang); L.A.F Vs d_{Mean} and O.Ang Vs d_{Min} Vs N.N.Ang are selected.

In considering all these variables, we have attempted to identify a consistent set of particle distributions which vary these parameters systematically. By varying the inputs systematically, although variables remained within each individual input range, the input combinations may cover high dimensional input space where there was no original training data. As such, these simulations are further interrogations of the model produced. This attempt is summarised in Table 7.1 for ADI and in Table 7.2 for the Al-Si-Sn case. To identify these distributions we have adopted the following notation: e.g. ARCE-θ where the first letter indicates the material used (A stands for ADI, B for Al-Si-Sn), the second letter indicates the object distribution (R stands for random, C for clustered), the third letter indicates the object area (O.A) (C stands for constant, V for varying), and the last letter indicates the shape of objects at angle θ (C stands for circular, E-θ stands for ellipse shapes at angle θ to the loading axis). If we consider the univariates chosen for ADI (Fig. 6.2a-c), we need to try and assess whether the dependence on C.A and L.A.F in fact reflects a large object size. By fixing the objects to be circles of equal size and varying their distribution (random compared with clustered) we can vary the C.A and L.A.F independently of object size and angle (ARCC and ACCC). We can then bring in the effect of object size by taking the random and clustered distributions, but now varying the object size (ARVC and ACVC). The N.N.N is hard to vary independently, but considering the 4 distributions already identified, it is likely that we will get the largest N.N.N for small objects which are locally clustered.

If we now go on to consider the two bivariate functions (Fig. 6.2d-e), we want to assess O.Ang Vs d_{Min} and d_{Mean} Vs N.N.Ang. To assess the bivariate function, first we need to consider ellipses (where we have taken the mean aspect ratio) with the two extremes of O.Ang imposed on the particle distribution. In our case, we

have aligned the ellipses parallel to and perpendicular to the loading axis. By considering both random and clustered distributions of constant size ellipses with parallel or perpendicular alignment (ACCE-0^o and ARCE-0^o, and ARCE-90^o and ARCE-90^o respectively) we should be able to assess the bivariate function in Fig. 6.2d. To vary d_{Mean} and N.N.A simultaneously (Fig. 6.2e) a further consideration of ARCC and ACCC may be helpful as the clustered distribution will have a smaller d_{Mean} from the parent particles (methodology detailed later). N.N.Ang is difficult to systematically vary. A final "complex" set of particle distributions of varying sized ellipses at varying angles in both random and clustered distributions have also been considered to provide an overview of the simulated data set ARVE-θ and ACVE-θ.

If we consider the Al-Si-Sn system as summarised in Table 7.2 and Fig. 6.5a-e a similar set of particle distributions can be considered that will also assess the various input dependencies revealed by the SUPANOVA decomposition. In all, 10 simulated particle distributions have therefore been considered for each case: circular objects of constant and varying size in random and clustered distributions, e.g. ARCC, ACCC, ARVC and ACVC, constant sized ellipses at 90^o and parallel to the loading axis in random and clustered distributions: e.g. ARCE-0^o, ACCE-0^o, ARCE-90^o and ACCE-90^o. Finally, 2 more complex particle distributions have been considered consisting of ellipses of varying size and object angle in both random and clustered distributions (e.g. ARVE-θ and ACVE-θ). These last 2 distributions are more realistic and give an overview of the model predictions for fatigue initiation sites. The procedure to produce these 10 simulated particle distributions is now described in the following section.

Components	Shapes	Related to	Simulated parameters	Notes
Cell Area (C.A) & Local Area Fraction (L.A.F)	Circular	Object Size + Object distribution	1. Fix Object Area (O.A) + Random Distribution (ARCC) (APP A1, fig 1)	Clustering may be used as a tool to vary the L.A. F (if objects are close L.A.F likely to be high – independent of object size)
			2. Fix O.A + Clustered Distribution (ACCC) (APP A1, fig 2)	
			3. Vary O.A + Random Distribution (ARVC) (APP A1, fig 3)	
			4. Vary O.A + Clustered Distribution (ACVC) (APP A1, fig 4)	
Number of Near Neighbours (N.N.N)	Circular	Object distribution	5. Fix O.A + Random Distribution (ARCC)	It is very difficult to simulate this independently. Depends on clustering and size of object.
			6. Fix O.A + Clustered Distribution (ACCC)	
			7. Vary O.A + Random Distribution (ARVC)	
			8. Vary O.A + Clustered Distribution (ACVC)	
Object Angle (O.Ang) VS Nearest Neighbour Distance (d_{Min})	Ellipse (With angle between loading axis)	Object distribution + Object Angle	9. Fix O.A + Random Distribution + Angle 90 (ARCE-90°) (APP A1, fig 5)	Two extremes of O.Ang have been considered. Clustered – should have a smaller d_{Min}.
			10. Fix O.A + Random Distribution + Angle 0 (ARCE-0°) (APP A1, fig 6)	
			11. Fix O.A + Clustered Distribution + Angle 90 (ACCE-90°) (APP A1, fig 7)	
			12. Fix O.A + Clustered Distribution + Angle 0 (ACCE-0°) (APP A1, fig 8)	
Mean Near Neighbour Distance (d_{Mean}) Vs Nearest Neighbour Angle (N.N.Ang)	Circular	Object distribution	13. Fix O.A + Random Distribution (ARCC)	N.N.Ang is hard to independently vary systematically. Clustered – should have a smaller d_{Mean} established from the parents. The d_{Mean} could be affected by the N.N.N.
			14. Fix O.A + Clustered Distribution (ACCC)	
			15. Vary O.A + Random Distribution (ARVC)	
			16. Vary O.A + Clustered Distribution (ACVC)	
Overall view	Ellipse	Object Size + Object Distribution	17. Vary O.A + Vary object Angle +Random Distribution (ARVE-θ) (APP A1, fig 9)	Provides overview of the simulated data set
			18. Vary O.A + Vary object Angle + Clustered Distribution (ACVE-θ) (APP A1, fig 10)	

Table 7.1: Description of particle distributions produced to assess the input components identified by the SUPANOVA decomposition for the ADI cases (see Fig. 6.2 a-e). The notation used here is as follows: e.g. ARCE-θ where the first letter indicates the material used (A stands for ADI), second letter stands for object distribution (R stands for random, C stands for clustered), third letter stands for object area (C stands for constant, V stands for varying), and the last letter stands for shape of objects at angle θ (C stands for circular, E-θ stands for ellipse shapes at angle θ to the loading axis). The particle distributions of this simulated data can be referred to in Appendix A.

Components	Shapes	Related to	Simulated parameters	Notes
Cell Area (C.A) & Local Area Fraction (L.A.F)	Circular	Object Size + Object distribution	1. Fix Object Area (O.A) + Random Distribution (BRCC) (APP B1, fig 1)	Clustering may be used as a tool to vary the L.A.F (if objects are close, L.A.F is likely to be high – independent of object size)
			2. Fix O.A + Clustered Distribution (BCCC) (APP B1, fig 2)	
			3. Vary O.A + Random Distribution (BRVC) (APP B1, fig 3)	
			4. Vary O.A + Clustered Distribution (BCVC) (APP B1, fig 4)	
Object Angle (O.Ang) VS Cell Angle (C.Ang)	Ellipse (With angle between loading axis)	Object distribution + Object Angle	5. Fix O.A + Random Distribution + Angle 90 (BRCE-90°) (APP B1, fig 5)	Two extremes of the O.Ang have been considered. Cell Angle depends on the object distributions.
			6. Fix O.A + Random Distribution + Angle 0 (BRCE-0°) (APP B1, fig 6)	
			7. Fix O.A + Clustered Distribution + Angle 90 (BCCE-90°) (APP B1, fig 7)	
			8. Fix O.A + Clustered Distribution + Angle 0 (BCCE-0°) (APP B1, fig 8)	
L.A.F Vs Mean Near Neighbour Distance (d_{Mean})	Circular	Object Size + Object distribution	9. Fix O.A + Random Distribution (BRCC)	L.A.F is related to object size while d_{Mean} distribution is affected by the number of near neighbours hence, object distribution must be varied.
			10. Fix O.A + Clustered Distribution (BCCC)	
			11. Vary O.A + Random Distribution (BRVC)	
			12. Vary O.A + Clustered Distribution (BCVC)	
O.Ang VS Nearest Neighbour Distance (d_{Min}) Vs Nearest Neighbour Angle (N.N.Ang)	Ellipse (With angle between loading axis)	Object distribution + Object Angle	13. Fix O.A + Random Distribution + Angle 0 (BRCE-90°)	It is hard to independently vary systematically. Therefore, by fixing the object angle to the 2 extreme values, we investigated the effect of d_{Min} and N.N.Ang. The d_{Mean} distribution could reflect the number of near neighbours, hence, object distribution considered.
			14. Fix O.A + Random Distribution + Angle 90 (BRCE-0°)	
			15. Fix O.A + Clustered Distribution + Angle 0 (BCCE-90°)	
			16. Fix O.A + Clustered Distribution + Angle 90 (BCCE-90°)	
Overall view	Ellipse	Object Size + Object Distribution	17. Vary O.A + Vary object Angle +Random Distribution (BRVE-θ) (APP B1, fig 9)	Provides an overview of the simulated data set
			18. Vary O.A + Vary object Angle + Clustered Distribution (BCVE-θ) (APP B1, fig 10)	

Table 7.2: Description of particle distributions produced to assess the input components identified by the SUPANOVA decomposition for the Al-Si-Sn cases (see Fig. 6.5 a-e). The notation used here is as follows: e.g. BRCE-θ where the first letter indicates the material used (B stands for Al-Si-Sn), second letter stands for object distribution (R stands for random, C stands for clustered), third letter stands for O.A (C stands for constant, V stands for varying), and the last letter stands for shape of objects at angle θ (C stands for circular, E-θ stands for ellipse shapes at angle θ to the loading axis). The particle distributions of this simulated data can be referred to in Appendix B.

7.2 Procedure and Specification for Simulated Data

The features selected by the SUPANOVA for imbalanced data can be explored and visualised further with the help of the simulated data. Chapter 2 has described the use of a particle simulation created by (Yang *et al.* 2000, Yang *et al.* 2001). We have identified a set of model particle distributions for both the ADI and Al-Si-Sn applications. The procedure and specification to produce these are now described in more detail.

1. A 2 dimensional rectangular field with nominal width of 1014 pixels/units and height of 653 pixels/units was specified.
2. In order to provide a realistic simulated data set, the Volume Fraction (VF) and hence the average area fraction (AF) of secondary phase particles found in the original data set and the simulated data set must be consistent. The secondary phase particles for the ADI are graphite nodules and for the case of the plain journal bearing are Si particles, both of which are roughly spherical in shape. Once the AF is known, a number of objects can be specified to fit into the simulated image area with the appropriate radius. In the case of the ADI, the average radius of the particles was then used to calculate the number of particles with respect to the nominal width specified. In the case of the Al-Si-Sn alloy, the average radius of the particles was much smaller, and so the nominal image area was effectively reduced to 101.4 by 65.3 pixel/unit to keep the number of particles considered in the simulation to a reasonable number. However, if the particles are very small in the simulation, then significant rounding errors due to pixel resolution will be present. As such, the scale is effectively magnified by ten times. The procedure for calculating the number of particles in each case is given briefly below.

   ADI

$$\begin{aligned} A_F &= \frac{\sum O.A}{\sum C.A} \\ &= 0.0913 \end{aligned}$$

Average Radius, R = 13 μ/units

Number of Objects in the field = $\frac{1014 \times 653 \times 0.0913}{\Pi \times R^2}$=113

Al-Si-Sn

$$A_F = \frac{\sum O.A}{\sum C.A}$$
$$= 0.0723$$

Average Radius, R = 1.28 μ/units (We magnified it by 10 times, therefore taken as 12.8 μ/units)

Number of Objects in the field = $\frac{1014 \times 653 \times 0.0723}{\Pi \times R^2}$=92

3. Once the number of particles in the image area has been defined, the size, shape and distribution of the objects can be varied as follows :
   - Circular objects with uniform or exponential size distribution (defined around a mean value)
   - Elliptical objects with uniform or exponential size distribution (defined around a mean value)

     Note: For a given area, the circle can be converted to an ellipse shape. This is given as : area of circle ($\Pi X R^2$) = area of Ellipse ($\Pi X A X B$) where A and B are the length of the major axis and length of the minor axis of an ellipse. A/B is the aspect ratio of the ellipse which is a feature obtained from the FBT. The mean A_r for ADI = 1.41 and for the Al-Si-Sn alloy = 1.49. Therefore, with this information available an object with a circular shape can be converted to a ellipse.
   - Elliptical objects with varying angles of the particle major axis to the loading axis
4. The centroid of the objects are then generated in the form of a random or clustered distribution. It is important to note that a strict constraint is imposed that the objects generated should not overlap (for a given shape, size and orientation) with each other.

   Random - The centroids of the objects are generated using a random number generated with a repeatable sequence.

Clustered - A set of "parent" centroids are generated which are at least 200 units/pixels away from each other. In the ADI - 11 parents were defined , in the Al-Si-Sn - 9 parents.

The average number of "children" associated with each parent can be calculated based on the number of parents that have been specified. In ADI - 10 children per parent, in Al-Si-Sn = 10 children per parent, thus allowing approximately 110 particles in total in the ADI and 90 particles in total in the Al-Si-Sn

The centroid of each child is based on the variance of the x and y co-ordinates specified with reference to its parent. For both the ADI & the Al-Si-Sn - x-axis variance =80, y-axis variance = 50.

Note : this clustering is a global clustering and is defined as such based on work done by (Yang *et al.* 2001). They systematically defined a global clustering effect from d_{mean} measurements obtained from the standard deviation divided by its mean (i.e. $COV_{dmean} = \frac{S.D\ d_{Mean}}{Mean\ d_{Mean}}$). If this value is greater then 0.36±0.02, this unambiguously indicates a clustered distribution.

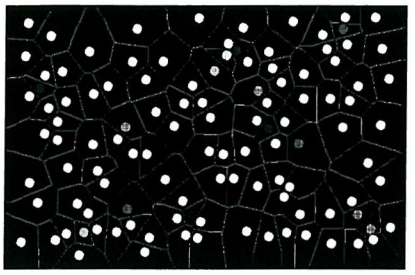
5. The O.Ang can be fixed to be parallel, perpendicular or random with respect to the (horizontal) loading axis.
6. Upon obtaining the relevant parameters (i.e. for circular objects - x and y coordinates and object radius; for ellipses - x and y coordinates, A and B chord lengths of the ellipse) these values are then digitised to produce the simulated images.
7. The tessellation analysis was then applied to the simulated image. The edge objects are eliminated as they are insufficiently defined in terms of near neighbours, C.A etc. However, it is important to ensure that the remaining AF remains within ±10% of its original value. Furthermore, for the case of random and clustered distributions of the objects, the value of the COV_{dmean} must be retained.

8. The cells generated from the tessellation were then labelled for later identification.
9. The tessellated information from this simulated data were then used as a test set for the 10 SUPANOVA classification models produced on the ten data set training-testing random positions. Prior to using it as a testing set, the data is normalised against the original data using its mean.
10. The consistency of initiation site selection was then assessed for each simulated distribution, and a clearer understanding of the importance of each component selected by the SUPANOVA approach achieved (as discussed in the following section).

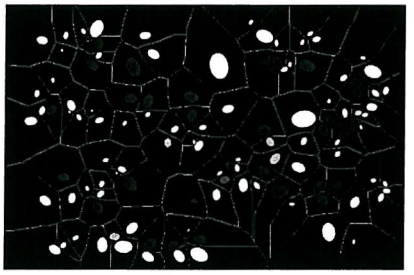
7.3 Use of Simulated Data to Enhance Visualisation

The two extreme examples of the simulated data cases can be seen in Fig. 7.1. The complete set of figures referred to in this section have been collated in Appendix A (APP A for ADI) and Appendix B (APP B for AL-Si-Sn). APP A1, Figures 1-10 show the simulated particle distributions and their associated tessellation cells in ADI. The consistency of initiation site identification by the 10 SUPANOVA models is given by the degree of contrast for a given particle, i.e. a white particle is never identified as initiating a crack (0/10) whereas a dark grey particle is always identified as a crack initiator (10/10). Those that were selected less than 5 times were not considered (allocated to the 0/10 group) and also, the boundary objects are not considered as they do not provide a full set of feature information. A similar set of figures were obtained for the case of the Al-Si-Sn (APP B1, Fig. 1-10). Tables 7.3 and 7.4 summarise the mean and standard deviations for each tessellation feature for all the simulated data sets for ADI and Al-Si-Sn respectively. It includes each of the simulated distributions and the breakdown for the "crack" and "no crack" classes for the original data distribution. The "crack" and "no crack" population distributions for each univariate of interest have also been systematically compared in histogram form for each simulated particle distribution. Appropriate bivariate plots of the two classes have also been considered. The results of these comparisons

are summarised in table 7.5 a,b,c (for ADI) and 7.6 a,b,c,d (for Al-Si-Sn) and are also discussed below.



a.) ADI, random object distribution, constant area, circular shapes.



b.) ADI, clustered object distribution, varying object area ellipse shapes at angle θ to the loading axis.

Figure 7.1: a & b are examples showing the two extreme cases for the simulated data set. a.) is a simple simulated data set where the object shapes are round (hence no O.Ang effect), O.A is fixed, with object distribution random. b.) is a more complex simulated data set where the object shapes are now ellipses (hence, O.Ang is a variable), O.A is varied, object distribution clustered.

7.3.1 Automotive Camshaft - ADI

Univariate Discussion (C.A,L.A.F,N.N.N)

The class means and histogram comparisons for ARCC (R is random distribution, C is constant O.A, and the next C is circular object) indicate that the "crack" class tends in fact to have a smaller C.A and a larger L.A.F (see Table 7.3 and APP A2.1, Fig. 1 and 2). If the clustered version of this particle distribution ACCC is now considered, a similar trend is observed. It should be noted that although differences in the mean values of these univariates are observed, the standard deviations (S.D) are relatively high. More initiation sites (10/10 cf. 0/10) are also predicted in the clustered than the random distribution (i.e APP A1, Fig. 1 (ARCC- 17%) and APP A1, fig 2 (ACCC-33%)). If we now vary O.A for both the random and clustered distributions (ARVC and ACVC) we can see that the "crack" class tends to have a larger O.A, C.A and L.A.F (APP A2.1, Fig. 3,4,5). In the more clustered case, the trend with C.A is less clear cut (APP A2.1, Fig. 6), however the mean and S.D values indicate that the "crack" class tends to have a larger value (table 7.3). More initiation sites are again predicted for the clustered case (APP A1, Fig. 3 (ARVC-28%) and APP A1, Fig. 4 (ACVC-36%)) although it is less clear cut. Considering these 4 simulations for the case of the N.N.N, given that the O.A is

fixed and the object distribution is random (ARCC), it is difficult to assess the class distribution (APP A2.1, Fig. 7). The "crack" class appears to be associated with either lower or higher values of the N.N.N compared to the "no crack" class. However, when the object is clustered (ACCC) the "crack" class is observed to have more N.N.N (APP A2.1, Fig. 8). If we now vary the O.A (ARVC and ACVC), a clearer view can be seen as the "crack" class has more N.N.N (APP A2.1, Fig. 9). In summary, assessing the simulated data to consider the univariate dependencies revealed by SUPANOVA shows that for the basic microstructural parameters covered by these simulations, a large C.A is not a good "crack" indicator. A large L.A.F is a better indicator for ADI fatigue crack initiation. This is more clearly shown when the O.A is varied (ARVC and ACVC). The effect of the N.N.N is shown more in the clustered object distribution (ACCC and ACVC) although this effect is less clear cut, once O.A is also varied, and the L.A.F effect outweighs it to some extent.

Bivariate Discussion (O.Ang and d_{Min}, d_{Mean} and N.N.Ang)

Now, let us consider the bivariate case for the O.Ang and the d_{Min}. The object shape now has changed to an elliptical shape so as to incorporate the two extreme angles involved in the O.Ang (i.e parallel (ARCE-0-ACCE-0) and perpendicular (ARCE-90 and ACCE-90) to the loading axis). Considering the case when the O.Ang is set perpendicularly (ARCE-90 and ACCE-90) to the loading axis, it was observed that the "crack" class has a smaller d_{Min} (APP A2.1, Fig. 10). This tallies with our SUPANOVA model which indicates that a large O.Ang with large d_{Min} is unlikely to initiate cracks. This observation was not clear for the case when the O.Ang is parallel (ARCE-0 and ACCE-0) to the loading axis (APP A2.1, Fig. 11, 12). Comparing the object clustering effect, we observed that more crack initiation was seen when the O.Ang is set parallel to the loading axis (APP A1, Fig. 6 (ARCE-0 (51%)) , APP A1, Fig. 8 (ACCE-0 (56%)) compared to APP A1, Fig. 5 (ARCE-90 (23%)) , APP A1, Fig. 7 (ACCE-90 (38%))).

The d_{Mean} and N.N.Ang is very difficult to vary systematically. However, the d_{Mean} is related to the object distribution. As such, the previous four sets of

simulated data (ARCC,ACCC,ARVC,ACVC) were further used in this bivariate analysis. Given the O.A is fixed and the objects are randomly distributed (ARCC), the "no crack" class tends to lie on the lower side of the hyperplane as d_{Mean} and N.N.Ang increase proportionally (APP A2.3, Fig. 13). This is reflected in our SUPANOVA model where a threshold value is seen beyond which cracks will start to initiate (Fig. 6.2e). When the object distribution is clustered (ACCC), the "no crack" class tends to lie in the middle value of the d_{Mean} with a relatively small N.N.Ang (APP A2.3, Fig. 14). When the O.A is varied (ARVC and ACVC), it becomes more difficult to assess the trends (APP A2.3, Fig. 15).

Validation Data set Discussion

The final two simulated data sets used were the ARVE-θ (Object distribution random, varying O.A and ellipse in shape at θ angle w.r.t the loading axis) and ACVE-θ (Object distribution clustered, varying O.A and ellipse in shape at θ angle w.r.t the loading axis). This resembles more closely the original data set. Large O.A, C.A, L.A.F and N.N.N and small d_{Min} tends to initiate cracks (APP A2.4, Fig. 16,18,19,20,21). The effect of the O.Ang shows that the crack class has a smaller O.Ang (APP A2.4, Fig. 17, 24). It is difficult to assess the effect of the mean near neighbour distance (APP A2.4, Fig. 22, 25). For the case where the objects are clustered, the "crack" class d_{Mean} is either relatively high or low. This might reflect the threshold we obtained from the SUPANOVA model (Fig. 6.2e). It is also observed that the mean value for the "crack" class is fairly similar for the case of the object being randomly distributed but higher for the case when the object is clustered (table 7.3). However, the value for the S.D is high in both cases for the "crack" class. Assessing the N.N.Ang becomes more difficult when the O.A is set to vary (APP A2.4, Fig. 23,26). Here, again, we see that when the object distribution is clustered more cracks are initiated (APP A1, Fig. 9 (ARVE-θ (37%)) and APP A1, Fig. 10 (ACVE-θ (41%))). The univariate analysis from this simulated data tallies with that of the model produced. This is reflected clearly when the O.A varies. However, the S.D values for the "crack" class are high for the components, O.A, C.A and L.A.F. This may indicate that there are also other factors affecting

the crack initiation other than those of the univariates such as bivariates and other combined object distribution effects. The bivariates are somewhat difficult to assess since the O.Ang and object size were varied simultaneously (APP A2.4, Fig. 27, 28).

ADI Conclusion

In summary, the simulated data set provides further understanding of the SU-PANOVA model, as shown in Table 7.5a-c. For example, when the O.A is varied, better understanding is obtained of the role of O.A, C.A and L.A.F (see point no. 3,4,7,8 in table 7.5a and Fig. 7.2). We can also assess the effect of the object distribution (see points 1,2,3,4 (table 7.5a) and 1,2,3,4 (table 7.5b) and also Fig. 7.3). This serves as an example for the univariate components. For the bivariate case, by fixing the O.A, a better understanding of the relationship between the d_{Mean} and N.N.Ang is provided (see points 5,6 (table 7.5b) and Fig. 7.4). Also, by fixing the O.Ang to the two extreme cases (i.e. parallel or perpendicular to the loading axis) a better understanding of the effect of O.Ang is obtained (points 1,2 (table 7.5b) and Fig. 7.5), with more initiation being found for the case where the ellipses major axes are parallel to the loading axis.

Particles Distribution		O.A	$O.A_r$	O.Ang	C.A	L.A.F	N.N.N	d_{Min}	d_{Mean}	N.N.Ang		
		Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	No. of objects	COV_{dMean}
Origin	COM	549.88 (1071.70)	1.40 (0.37)	0.78 (0.41)	6022.60 (4973.30)	6.71 (7.42)	5.76 (1.87)	17.36 (17.02)	57.03 (24.09)	0.75 (0.40)	2923	0.42
	Crack	2326.89 (2549.83)	1.30 (0.28)	0.69 (0.45)	12340.84 (7628.74)	15.87 (10.13)	7.60 (2.21)	16.23 (16.36)	64.81 (21.08)	0.77 (0.47)	116	
	NoCrack	476.45 (890.87)	1.40 (0.38)	0.79 (0.41)	5761.52 (4653.25)	6.34 (7.03)	5.68 (1.82)	17.40 (17.04)	56.71 (24.16)	0.75 (0.46)	2807	
ARCC	COM	540.00 (0)	1.05 (0)	0.78 (0)	5665.92 (2150.07)	11.08 (4.64)	5.61 (1.11)	18.63 (11.91)	54.24 (17.93)	0.75 (0.41)	80	0.33
	Crack	540.00 (0)	1.05 (0)	0.78 (0)	4683.26 (2953.82)	15.68 (7.54)	5.93 (1.82)	11.92 (10.05)	46.42 (28.03)	0.81 (0.43)	14	
	NoCrack	540.00 (0)	1.05 (0)	0.78 (0)	5874.37 (1902.99)	10.10 (3.04)	5.55 (0.90)	20.05 (11.85)	55.90 (14.75)	0.73 (0.41)	66	
ACCC	COM	536.00 (0)	1.01 (0)	0.78 (0)	5507.90 (3159.27)	13.00 (7.10)	5.69 (1.20)	12.58 (9.02)	51.65 (25.22)	0.81 (0.47)	78	0.49
	Crack	536.00 (0)	1.01 (0)	0.78 (0)	5468.08 (4769.19)	16.95 (9.82)	6.00 (1.39)	10.71 (8.33)	49.50 (36.45)	0.79 (0.44)	26	
	NoCrack	536.00 (0)	1.01 (0)	0.78 (0)	5528.98 (1882.00))	10.91 (3.82)	5.53 (1.06)	13.58 (9.29)	52.78 (16.88)	0.82 (0.49)	52	
ARVC	COM	497.11 (415.62)	1.05 (0.05)	0.80 (0.36)	5966.34 (2759.12)	8.25 (5.64)	5.65 (1.25)	21.09 (13.13)	59.60 (19.37)	0.85 (0.48)	81	0.33
	Crack	989.39 (364.85)	1.03 (0.02)	0.79 (0)	7631.44 (2718.32)	14.24 (5.76)	6.70 (1.06)	17.86 (12.61)	60.71 (18.86)	0.74 (0.52)	23	
	NoCrack	301.90 (233.45)	1.06 (0.05)	0.78 (0)	5306.04 (2504.02)	5.88 (3.41)	5.24 (1.06)	22.37 (13.22)	59.16 (19.72)	0.89 (0.45)	58	
ACVC	COM	530.82 (505.15)	1.05 (0.04)	0.75 (0.34)	5429.42 (3554.30)	11.75 (9.54)	5.53 (1.54)	16.61 (14.53)	51.82 (26.15)	0.83 (0.47)	76	0.5
	Crack	987.14 (553.61)	1.03 (0.02)	0.79 (0)	6080.02 (4246.01)	20.23 (9.75)	6.46 (1.64)	12.99 (13.15)	45.88 (23.56)	0.95 (0.43)	28	
	NoCrack	264.63 (186.30)	1.06 (0.05)	0.78 (0)	5049.91 (3066.44)	6.81 (4.77)	4.98 (1.19)	18.72 (15.00)	55.28 (27.19)	0.76 (0.49)	48	
ARCE-90	COM	554.95 (0.39)	1.44 (0)	1.51 (0)	5777.49 (2093.07)	11.13 (4.69)	5.68 (1.40)	18.68 (13.68)	54.83 (18.12)	0.87 (0.42)	77	0.33
	Crack	555.00 (0)	1.44 (0)	1.51 (0)	5235.91 (2946.71)	13.77 (6.55)	6.22 (1.93)	7.96 (6.80)	55.70 (25.77)	0.90 (0.44)	18	
	NoCrack	554.95 (0.39)	1.44 (0)	1.51 (0)	5942.72 (1754.06)	10.33 (3.66)	5.51 (1.17)	21.95 (13.61)	54.57 (15.34)	0.86 (0.42	59	
ARCE-0	COM	553.95 (0.32)	1.52 (0)	0.00 (0)	5666.50 (2609.62)	12.03 (5.7)	5.77 (1.09)	16.86 (12.75)	54.55 (18.39)	0.75 (0.48)	82	0.34
	Crack	553.98 (0.15)	1.52 (0)	0.00 (0)	5673.08 (3231.99)	13.34 (7.19)	6.00 (1.08)	17.09 (15.14)	55.54 (25.41)	0.93 (0.41)	42	
	NoCrack	553.95 (0.32)	1.52 (0)	0.00 (0)	5659.58 (1778.35)	10.66 (3.04)	5.53 (1.06)	16.63 (9.81)	53.52 (13.52)	0.56 (0.48)	40	
ACCE-90	COM	555.00 (0)	1.44 (0)	1.51 (0)	5735.28 (3467.91)	13.38 (7.61)	5.86 (1.13)	15.78 (12.22)	51.49 (25.36)	0.92 (0.43)	77	0.49
	Crack	554.97 (0.19)	1.44 (0)	1.51 (0)	5079.43 (4397.40)	17.94 (9.81)	5.93 (1.25)	10.74 (8.79)	46.41 (32.45)	1.05 (0.41)	29	
	NoCrack	555.00 (0)	1.44 (0)	1.51 (0)	6131.51 (2738.82)	10.62 (3.96)	5.52 (1.03)	18.82 (13.05)	54.57 (19.66)	0.83 (0.43)	48	
ACCE-0	COM	553.79 (1.20)	1.52 (0.02)	0.00 (0)	5487.24 (3378.21)	14.71 (9.54)	5.63 (1.27)	15.51 (13.03)	49.46 (27.79)	0.73 (0.40)	78	0.56
	Crack	553.39 (1.57)	1.51 (0.03)	0.00 (0)	5297.37 (4195.71)	17.77 (11.46)	5.89 (1.45)	14.39 (13.59)	47.68 (33.73)	0.72 (0.38)	44	
	NoCrack	553.79 (1.20)	1.52 (0.02)	0.00 (0)	5732.96 (1892.05)	10.75 (3.53)	5.29 (0.91)	16.96 (12.31)	51.77 (17.58)	0.75 (0.44)	34	
ARVE-θ	COM	581.1 (535.8)	1.45 (0.08)	0.8 (0.44)	6133.76 (2981.07)	9.56 (7.20)	5.78 (1.34)	19.09 (14.73)	58.46 (19.93)	0.84 (0.42)	78	0.34
	Crack	1083.62 (563.08)	1.44 (0.05)	0.75 (0.44)	7927.56 (3028.83)	15.49 (8.02)	6.62 (1.29)	14.90 (14.14)	59.38 (22.96)	0.88 (0.47)	29	
	NoCrack	283.69 (180.87)	1.45 (0.10)	0.83 (0.44)	5072.11 (2410.36)	6.06 (3.45)	5.29 (1.10)	21.58 (14.64)	57.92 (18.13)	0.81 (0.40)	49	
ACVE-θ	COM	555.47 (2.58)	1.47 (0.04)	0.89 (0.45)	5868.18 (2123.59)	10.75 (4.24)	5.67 (1.13)	18.11 (12.91)	55.97 (16.64)	0.82 (0.46)	85	0.43
	Crack	907.63 (541.30)	1.43 (0.06)	0.62 (0.42)	6791.68 (3674.81)	16.12 (8.80)	6.69 (1.51)	12.11 (12.65)	55.16 (25.51)	0.89 (0.47)	35	
	NoCrack	248.96 (193.49)	1.46 (0.12)	0.90 (0.42)	4203.78 (2367.22)	7.07 (5.03)	5.10 (1.30)	14.94 (12.02)	48.44 (18.67)	0.84 (0.47)	50	

Table 7.3: Summary of the mean and standard deviation (S.D) values of the simulated data sets and the original data set "Origin" for the ADI. COM denotes the complete set of simulated data. "crack" and "no crack" class denotes the breakdown of their class distributions. NOTE: the values of the boundary cells are not considered here and some slight rounding errors may appeared, this is due to conversion from simulated data to FBT data.

Components	Simulated parameters	Results	Summary
Cell Area (C.A) & Local Area Fraction (L.A.F)	1. Fix O.A + Random Distribution (ARCC)	• Crack class tends to have a smaller C.A (APP A2.1, fig 1) and larger L.A.F (APP A2.1, fig 2)	• Our simulation has shown consistently that purely considering a large C.A as an indicator for crack initiation, is not a true reflection of the situation. This may be partly due to the high standard deviation (S.D) observed (table 7.3). • However, it is true that as the L.A.F gets larger a crack is more likely to initiate. • Higher consistency (i.e. more 10/10 cases) of larger O.A causing crack initiation were pick up when the object size is varied. • Clustering tends to result in more crack initiation sites occurring.
	2. Fix O.A + Clustered Distribution (ACCC)	• Crack class tends to have a smaller C.A and larger L.A.F • More Cracks were initiated for clustered case (17% Random (APP A1, fig 1) Vs 33% Clustered (APP A1,fig 2))	
	3. Vary O.A + Random Distribution (ARVC)	• Crack class has larger O.A, C.A and L.A.F (APP A2.1, fig 3,4,5 respectively).	
	4. Vary O.A + Clustered Distribution (ACVC)	• Crack class has larger O.A • C.A is difficult to assess (APP A2.1, fig 6), but the mean values show that the crack class tends to have larger C.A (table 7.3) • Crack class has a larger L.A.F • More cracks were initiated for clustered case (28% Random (APP 1, fig 3) Vs 36% Clustered (APP 1, fig 4))	
Number of Near Neighbours (N.N.N)	5. Fix O.A + Random Distribution (ARCC)	• It is difficult to assess which class distribution is more significant, but the crack class appears to have either fewer or more N.N.N than the no crack class (APP A2.1, fig 7). The mean value for both class are similar but the crack class has higher S.D value (table 7.3).	• It is interesting to see that when the O.A is varied, the significance of the N.N.N in the crack class is highlighted more easily. This might be due to the effect of the significance of the large objects and hence the larger L.A.F. • Crack class has more N.N.N. Another indication of clustering effect.
	6. Fix O.A + Clustered Distribution (ACCC)	• Crack class has more N.N.N. The crack class has higher mean and S.D values (APP A2.1, fig8).	
	7. Vary O.A + Random Distribution (ARVC))	• Crack class has more N.N.N and this becomes more apparent when the O.A is varied (APP A2.1, fig 9)	
	8. Vary O.A + Clustered Distribution (ACVC)	• Crack class has more N.N.N and this become more apparent when the object size is varied.	

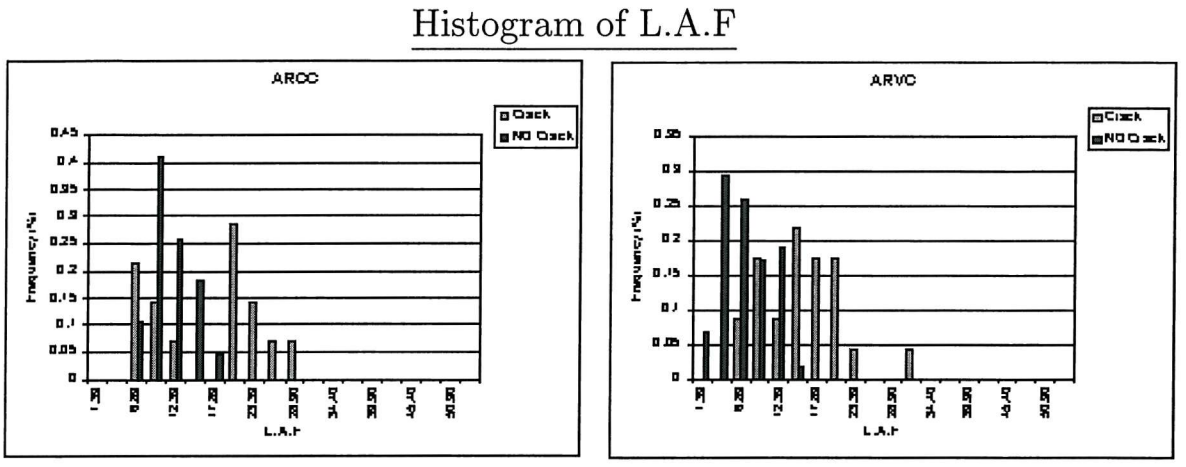
Table 7.4a: Summary of results for ADI obtained from the simulated data set produced to enhance model interpretability.

Components	Simulated parameters	Results	Summary
Object Angle (O.Ang) VS Nearest Neighbour Distance (d_{Min})	1. Fix O.A + Random Distribution + Angle 90 (ARCE-90°)	• The d_{Min} for crack class is smaller (APP A2.2, fig 10) • Crack initiation sites are 23% of particles (APP A1, fig 5).	• When the O.Ang is perpendicular to the loading axis, crack initiation is less likely compared to when it is parallel. Furthermore at 90° angle, the mean and S.D value of the d_{Min} for the no crack class is higher than the crack class (table 7.3). • Our SUPANOVA model suggests that large O.Ang, with large d_{Min} will not initiate cracks. This is reflected here for self-consistent particles distributions.
	2. Fix O.A + Random Distribution + Angle 0 (ARCE-0°)	• The mean value for d_{Min} for both classes are very similar (see table 7.3, crack value is 17.09 and no crack value is 16.63 and also APP A2.2, fig.11) • Crack initiation site are 51% of particles (APP A1, fig 6)	
	3. Fix O.A + Clustered Distribution + Angle 90 (ACCE-90°)	• The d_{Min} for the crack class is smaller • Crack initiation site are 38% of particles (APP A1, fig 7)	
	4. Fix O.A + Clustered Distribution + Angle 0 (ACCE-0°)	• The mean value for d_{Min} for both classes are very similar (see table 7.3, crack value is 14.39 and no crack value is 16.96 and also APP A2.2, fig.12) • Crack initiation site are 56% of particles (APP A1, fig 8)	
Mean Near Neighbour Distance (d_{Mean}) Vs Nearest Neighbour Angle (N.N.Ang)	5. Fix O.A + Random Distribution (ARCC)	• The no Crack class tends to lie on smaller value of N.N.Ang (APP A2.3, fig 13)	• For a given fixed O.A with object distribution being random, a distinction can be made between the crack and no crack classes (at least from the bivariate plots) with the "no crack" class tending to lie on smaller value of N.N.Ang. For the case of when the object are clustered, the "no crack" class tending to lie on middle values of d_{Mean} with small N.N.Ang.
	6. Fix O.A + Clustered Distribution (ACCC)	• The no Crack class tends to lie on middle values of d_{Mean} with small N.N.Ang (APP A2.3, fig 14)	
	7. Vary O.A + Random Distribution (ARVC)	• It is difficult to assess the effect due to varying O.A (APP A2.3, fig. 15).	
	8. Vary O.A + Clustered Distribution (ACVC)	• It is difficult to assess the effect due to varying O.A	

Table 7.4b: Continued from Table 7.5a.

Components	Simulated parameters	Results	Summary
Overview	1. Vary O.A + Vary object Angle +Random Distribution (ARVE-θ)	• Crack has large O.A (APP A2.4, fig 16) • Crack class has smaller O.Ang (APP A2.4, fig 17) • Crack has large C.A (APP A2.4, fig 18) • Crack has large L.A.F (APP A2.4, fig 19) • Crack has large N.N.N (APP A2.4, fig 20) • Crack has smaller d_{Min} (APP A2.4, fig 21) • The crack class d_{Mean} is either on the large value or the small value side). Its S.D is also higher than the no crack class (table 7.3). While the no crack class are more centered around the middle value of the d_{Mean}. (APP A2.4, fig 22). • Difficult to assess the N.N.Ang (APP A2.4, fig 23). • Crack initiation sites are 37% of particles (APP A1, fig 9).	• Varying the O.A leads to more consistenct identification of initiation and also easier identification that a large object, large C.A and large L.A.F initiate a cracks. However, their S.D are higher than the no crack class which indicates others factors such as those of the bivariates and the clustering effect also contribute to crack initiation. • The univariate tallies with that of the model produced. • The bivariate is more difficult to visualise as now the complexity between each features varies (e.g. APP A2.4, fig 27 and 28). • Clustering causes more crack initiation.
	2. Vary O.A + Vary object Angle + Clustered Distribution (ACVE-θ)	• Crack has large O.A • Crack class has smaller O.Ang (APP A2.4, fig 24) • Crack has large C.A • Crack has large L.A.F • Crack has large N.N.N • Crack has smaller d_{Min} • Difficult to assess the d_{Mean} (APP A2.4, fig 25). The mean values indicate that crack class has larger d_{Mean} with high S.D (table 7.3). • Difficult to assess the N.N.Ang (APP A2.4, fig 26). • Crack initiation sites are 41% of particles (APP A1, fig 10).	

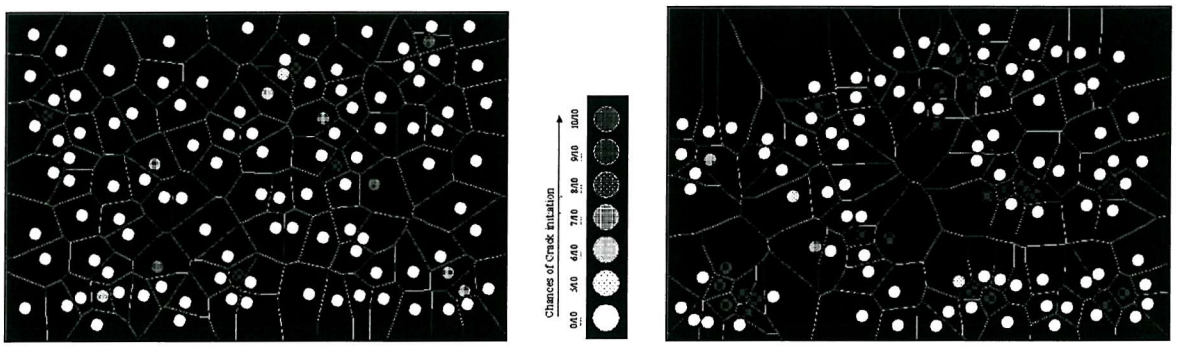
Table 7.4c: Continued from Table 7.5a.



a.) ARCC - O.A are FIXED

b.) ARVC - O.A are VARIED

Figure 7.2: The histogram of L.A.F. When the O.A are fixed (a), it is difficult to see the effects of "crack" initiation as compared to the case when the O.A is varied (b). From (b), the "crack" class appears to have a positive correlation with L.A.F. Similar trends were observed for C.A and O.A.

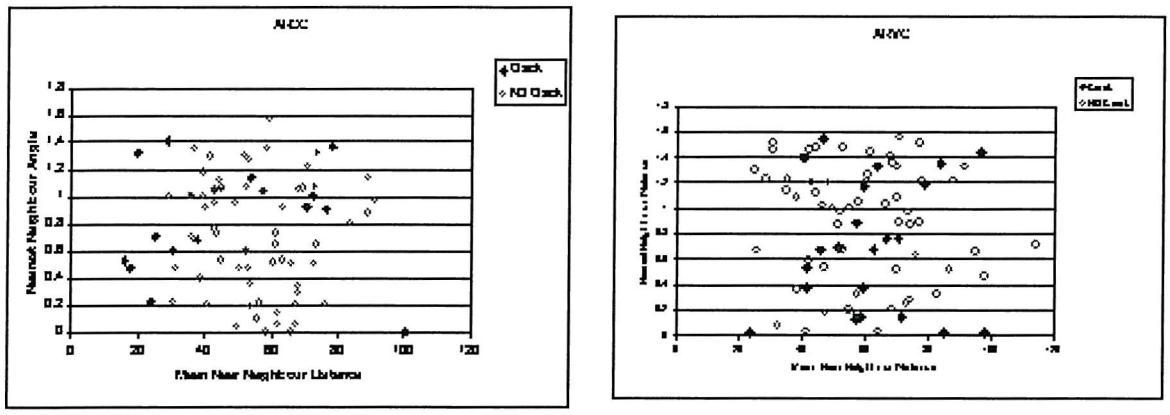


a.) ADI, random object distribution, constant area, circular shapes.

b.) ADI, clustered object distribution, constant area, circular shape.

Figure 7.3: Given that the O.A are fixed and the object shape is circular (i.e. no effect of O.Ang), it appears that the clustered (b) object distribution has more "crack" initiations than the random (a) case.

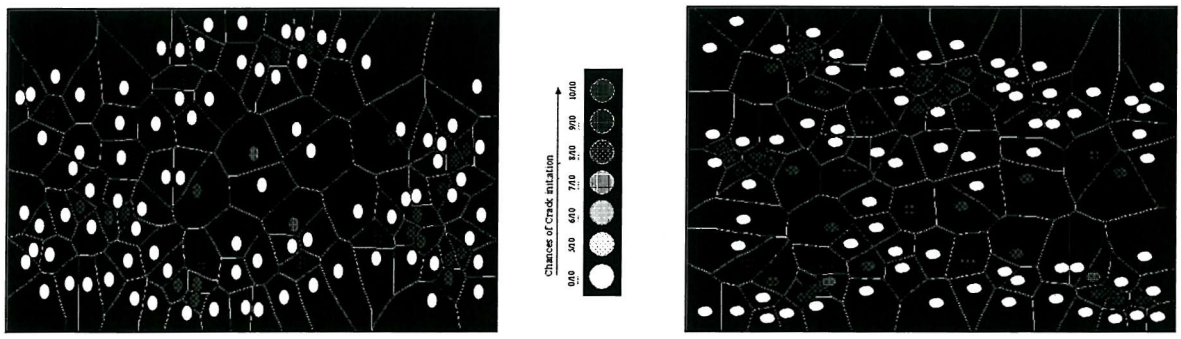
Bivariate plots of N.N.Ang Vs d_{Mean}



a.) ARCC - O.A are FIXED

b.) ARVC - O.A are VARIED

Figure 7.4: The bivariate plot of N.N.Ang and d_{Mean}. When the O.A is fixed (a), the "no crack" class lies on the right hand side of the hyperplane (i.e. if you draw a diagonal line between (0,0) and 120,1.4). Given the O.A as fixed and the objects are randomly distributed (ARCC), the "no crack" class tends to lie on the lower side of the hyperplane.This implies that as d_{Mean} and N.N.Ang increase proportionally cracks are unlikely to initiate. When the O.A is varied it became difficult to see the trends.



a.) ADI, fixed object area, clustered object distribution, constant object area, ellipse shapes at angle 90° to the loading axis.

b.) ADI, fixed object area, clustered object distribution, constant object area, ellipse shapes parallel to the loading axis.

Figure 7.5: Shows the effect of O.Ang. When the O.A is fixed and the object distribution is clustered, more "crack" initiations are observed when the O.Ang is parallel (b) to the loading axis than the case when it is perpendicular (a). Similar trends were observed for the case when the object distribution was random.

7.3.2 Automotive plain bearing lining -Al-Si-Sn

Univariate Discussion (C.A,L.A.F)

Again, let us consider the univariates case (i.e. C.A and L.A.F) first. For a given fixed O.A with random or clustered distribution, (i.e. BRCC & BCCC), the crack class C.A and L.A.F tends to fall either on low or high values (APP B2.1, Fig. 1,2,3,4 compared to the "no crack" class). Further analysis of the mean and S.D values indicates that the C.A appears to be larger for crack initiating Si particles (table 7.4). For L.A.F, the analysis of the mean and SD is less clear (i.e. for the BRCC distribution, they are fairly similar for both classes while in BCCC, the "crack" class is higher). However, it is clear that when the objects are distributed randomly, there is less crack initiation compared to clustered distributions (APP B1, Fig. 1 (BRCC - 10%) and Fig. 2 (BCCC - 30%)). Upon the O.A being varied (i.e. BRVC and BCVC), it can be seen that large O.A, C.A and L.A.F tends to initiate cracks (APP B2.1, Fig. 5,6,7). However, the effect of object distribution on "crack" initiation is less obvious (APP B1, Fig. 3 (BRVC - 33%) and Fig. 4 (BCVC - 35%)).

Bivariate Discussion (O.Ang and C.Ang, L.A.F and d_{Mean})

Now, let us consider the bivariate analysis for O.Ang versus C.Ang. The elliptical shapes with two extreme O.Ang situations were used in this case, the object being perpendicular to the loading axis (i.e. BRCE-90 and BCCE-90) and parallel to the loading axis (i.e. BRCE-0 and BCCE-0). When the O.Ang is set to be parallel to the loading axis, the "crack" class has large C.Angs (APP B2.2, Fig. 8) and more cracks are observed (APP B1, Fig. 5 and 7) than when the O.Ang is perpendicular to its loading axis (APP B1, Fig. 6 and 8). When the object is perpendicular to its loading axis, the effect of C.Ang is difficult to assess as the "crack" class is distributed on its low and high values (APP B2.2, Fig. 9). This effect coincides with the SUPANOVA model (Fig. 6.5c and Fig. 6.6b) as we can see that the "crack" class tends to be on the high side of the O.Ang and is well distributed along the C.Ang (i.e in Fig. 6.5c the C.Ang is large and in Fig. 6.6b the C.Ang is small). When the objects are clustered (BCCE-90), the effect of C.Ang is

more difficult to assess (APP B2.2, Fig 10). The effect of crack initiation related to the object distribution in this case is not clear (i.e. APP B1, Fig. 5 (BRCE-0 (61%)), APP B1, Fig. 7 (BCCE-0 (62%)) compared to APP B1, Fig. 6 (BRCE-0 (18%)), APP B1, Fig. 8 (BCCE-90 (22%))). Next, consider the bivariates of L.A.F versus d_{Mean}. Given that the O.A is fixed (BRCC and BCCC), we observed that the "crack" class lies in the region where it has a small L.A.F with high d_{Mean} or large L.A.F with low d_{Mean} (APP B2.3, Fig. 11). This might be a reflection of the threshold effect from our SUPANOVA model (see Fig. 6.5d). Once the O.A is varied (i.e. BRVC and BCVC), the trends for the "crack" class are clearer and it has now shifted upwards with higher d_{Mean} and high L.A.F (APP B2.3, Fig. 12). We observed further that there is an inverse correlation (as might be expected) between the L.A.F and d_{Mean} which is approximately exponential for both classes.

Trivariate Discussion (O.Ang and d_{Min} and N.N.Ang)

Now consider the trivariates for the O.Ang, d_{Min} and N.N.Ang. In this simulation the O.A and the O.Ang is fixed, to simplify the analysis. The O.Ang is again fixed as perpendicular or parallel to the loading axis. As such, we investigated the other two components (d_{Min} and N.N.Ang) using the bivariate plots. It was observed that when the O.Ang is parallel to the loading axis (BRCE-0, BCCE-0) the "crack" class tends to have a small d_{Min} (APP B2.4, Fig. 13 and 15) ranging from 0-2.5. This is further shown in APP B2.4, Fig. 16 indicating that the "crack" class has a smaller N.N.Ang. On the other hand, when it is perpendicular to the loading axis (APP B2.4, Fig 14, 17) the d_{Min} are well distributed. The analysis when the O.Ang is perpendicular to the loading axis (i.e. BRCE-90 and BCCE-90) show that the "crack" class has larger d_{Min} . The model from SUPANOVA (Fig. 6.5e) indicates that as O.Ang, d_{Min} and N.N.Ang gets larger, cracks are unlikely to initiate. Within our 0^o particle simulation this can be seen to be true, although crack initiations are more prevalent for the distribution as a whole.

Validation Data set Discussion

The overview of this simulation set was a distribution with an elliptical shape object with varying O.A, O.Ang, random (BRVE-θ) or clustered (BCVE-θ). It

was observed that in this more realistic distribution which is closer to the original distribution, the "crack" class has large O.A, O.Ang, C.A, C.Ang, L.A.F, more N.N.N and smaller N.N.Ang. Although the O.Ang for the clustered object is difficult to assess in its histogram plot (APP B2.5, Fig. 23), the mean and S.D values show that "crack" class value is higher (Table 7.4). When the object distribution is random (BRVE-θ), the effect of d_{Min} (APP B2.5, Fig. 18) and d_{Mean} (APP B2.5, Fig. 19) is difficult to distinguish even by considering their mean and S.D values. However, when the object distributions are clustered (BCVE-θ) the value of these two features becomes small for the "crack" class. It is also worth noting that in this instance, the $C.A_r$ is now fairly similar for both classes as the objects are clustered (APP B2.5, Fig. 24). The effect on numbers of crack initiation sites of object distribution is not clear here (APP B1, Fig. 9 (BRVE-θ, random 28%) and Fig. 10 (BCVE-θ, clustered 24%). However, it implies that O.A and hence object size are important for crack initiation.

Let us now consider the case of bivariate and trivariate components. The bivariate plot for both (random or clustered object distribution) shows that the "crack" class has large C.Ang value above $0.8 \approx 46^o$ (APP B2.5, Fig. 20 and 25). For the case of the bivariates between d_{Mean} and L.A.F, it can be seen that the classes can be separated by an approximated curve with "crack" class on the higher side of the curve (i.e. low d_{Min}, and high L.A.F). On further observation, when the objects are clustered (APP B2.5, Fig. 26) the "crack" class tends to have large L.A.F compared to the randomly distributed population (APP B2.5, Fig. 21). The variation in trivariate components (O.Ang, N.N.Ang and d_{min}) are considered via the bivariate plots between the N.N.Ang and d_{Min} for the two fixed O.Ang conditions. For randomly distributed objects (BRVE-θ) and large O.Ang, the bivariate plots of N.N.Ang and d_{Min} show that the "crack" class tends to have smaller N.N.Ang (APP B2.5, Fig. 22) mostly below value of $1 \approx 58^o$). This tallies with our SUPANOVA model which indicates large O.Ang, large d_{Min} and large N.N.Ang make crack initiation unlikely. For the case when the objects are clustered (BCVE-θ), the effect of O.Ang is difficult to distinguish between classes (as discussed earlier).

However, the bivariate plot of N.N.Ang and d_{Min} show the "crack" class tends to have smaller d_{Min} (APP B2.5, Fig. 27).

Al-Si-Sn Conclusion

In summary, once again, we see that by fixing the O.A, we see the importance of the object distribution (i.e. clustered distribution is likely to have more cracks (Table 7.6a points 1 and 2). By varying the O.A, the clustering effect is overshadowed by the object size, hence L.A.F becomes more important. As such, large O.A, C.A and L.A.F (table 7.6a, points 3 and 4) are likely to initiate cracks. Furthermore, with the O.A being a varying parameter, we see that there is an exponential relationship between L.A.F and d_{Mean} (table 7.6b points 1-4 and Fig. 7.6). The O.Ang is assessed by varying between the two extreme values (i.e. perpendicular (BRCE-90, BCCE-90) or parallel (BRCE-0, BCCE-0)) to the loading axis. Results show that as the O.Ang is parallel to the loading axis, more crack initiation is observed (table 7.6a points 5 and 7) as compared to those perpendicular to the loading axis. Also, when the O.Ang is parallel to the loading axis, it was observed that the d_{Min} was low in order to initiate cracks (table 7.6b, points 5 and 7 and Fig. 7.7) when the O.Ang are large.

Particles Distribution		O.A	$O.A_r$	O.Ang	C.A	$C.A_r$	C.Ang	L.A.F	N.N.N	d_{Min}	d_{Mean}	N.N.Ang	No.of Objects	COV_{dMean}
		Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)	Mean (S.D)		
Origin	COM	5.23 (6.44)	1.49 (0.47)	0.88 (0.41)	72.23 (49.49)	1.67 (1.40)	0.78 (0.45)	7.15 (5.72)	5.70 (1.45)	2.14 (1.88)	6.47 (2.66)	0.82 (0.46)	2938	0.41
	Crack	12.17 (11.54)	1.49 (0.36)	0.90 (0.41)	113.65 (58.60)	1.49 (0.42)	0.88 (0.43)	10.48 (6.42)	6.39 (1.39)	2.61 (1.87)	7.59 (2.50)	0.73 (0.44)	163	
	NoCrack	4.82 (5.75)	1.49 (0.48)	0.87 (0.41)	69.80 (47.81)	1.68 (1.43)	0.78 (0.45)	6.95 (5.62)	5.66 (1.44)	2.12 (1.79)	6.40 (2.65)	0.82 (0.46)	2775	
BRCC	COM	5.36 (0)	1.01 (0)	0.81 (0)	71.02 (30.24)	1.54 (0.38)	0.82 (0.44)	9.16 (4.25)	5.67 (1.18)	2.38 (1.64)	6.40 (2.14)	0.68 (0.42)	56	0.33
	Crack	5.36 (0)	1.01 (0)	0.81 (0)	81.03 (48.63)	1.82 (0.42)	1.15 (0.37)	9.23 (5.31)	6.50 (1.38)	1.81 (2.23)	7.89 (3.31)	0.47 (0.40)	6	
	NoCrack	5.36 (0)	1.01 (0)	0.80 (0)	69.98 (28.14)	1.52 (0.37)	0.78 (0.44)	9.15 (4.18)	5.59 (1.14)	2.44 (1.58)	6.25 (1.97)	0.70 (0.42)	58	
BCCC	COM	5.36 (0)	1.01 (0)	0.81 (0)	67.88 (50.42)	1.55 (0.48)	0.71 (0.46)	11.97 (7.14)	5.75 (1.19)	1.84 (1.33)	5.80 (3.14)	0.95 (0.42)	63	0.54
	Crack	5.36 (0)	1.01 (0)	0.81 (0)	89.32 (79.15)	1.40 (0.31)	0.92 (0.36)	13.98 (10.84)	5.84 (1.42)	1.91 (1.69)	6.80 (4.84)	0.77 (0.47)	19	
	NoCrack	5.36 (0)	1.01 (0)	0.81 (0)	58.63 (27.42)	1.62 (0.53)	0.62 (0.48)	11.10 (4.67)	5.70 (1.09)	1.81 (1.17)	5.36 (1.94)	1.02 (0.38)	44	
BRVC	COM	4.80 (4.24)	1.06 (0.05)	0.76 (0.36)	65.76 (33.29)	1.62 (0.46)	0.81 (0.47)	7.36 (5.33)	5.68 (1.32)	2.15 (1.76)	6.53 (2.06)	0.68 (0.45)	65	0.32
	Crack	9.15 (4.37)	1.03 (0.02)	0.81 (0)	81.53 (36.66)	1.55 (0.46)	0.82 (0.47)	12.59 (5.31)	6.14 (1.08)	1.91 (1.45)	6.64 (2.06)	0.75 (0.47)	22	
	NoCrack	2.58 (1.72)	1.07 (0.06)	0.80 (0)	57.68 (28.62)	1.66 (0.45)	0.81 (0.48)	4.68 (2.73)	5.44 (1.39)	2.27 (1.90)	6.48 (2.08)	0.64 (0.44)	43	
BCVC	COM	4.70 (3.93)	1.06 (0.05)	0.77 (0.39)	68.82 (42.33)	1.65 (0.43)	0.86 (0.42)	9.04 (8.47)	5.77 (1.45)	1.98 (1.37)	6.21 (2.91)	0.75 (0.40)	62	0.47
	Crack	8.22 (4.46)	1.03 (0.02)	0.81 (0)	91.42 (51.42)	1.51 (0.34)	0.82 (0.45)	13.83 (11.16)	6.73 (1.49)	1.79 (1.47)	7.02 (3.73)	0.68 (0.33)	22	
	NoCrack	2.76 (1.61)	1.07 (0.06)	0.80 (0)	56.39 (30.55)	1.72 (0.46)	0.88 (0.40)	6.40 (5.01)	5.25 (1.15)	2.08 (1.32)	5.77 (2.29)	0.79 (0.44)	40	
BRCE-0	COM	5.55 (0.01)	1.44 (0.01)	1.51 (0)	72.90 (35.79)	1.43 (0.32)	0.92 (0.38)	9.85 (5.40)	5.81 (1.17)	2.09 (1.82)	6.27 (2.12)	0.85 (0.43)	57	0.34
	Crack	5.55 (0.01)	1.44 (0.01)	1.51 (0)	69.71 (38.53)	1.40 (0.28)	1.08 (0.30)	10.85 (6.22)	5.51 (1.01)	1.60 (1.72)	6.02 (2.51)	0.66 (0.40)	35	
	NoCrack	5.55 (0.01)	1.44 (0.01)	1.51 (0)	77.98 (31.13)	1.48 (0.36)	0.66 (0.35)	8.27 (3,28)	6.27 (1.28)	2.87 (1.74)	6.68 (1.97)	1.15 (0.29)	22	
BRCE-90	COM	5.54 (0.01)	1.52 (0)	0.00 (0)	72.91 (34.46)	1.52 (0.37)	0.64 (0.39)	9.57 (5.06)	5.67 (1.11)	2.07 (1.65)	6.46 (2.26)	0.69 (0.43)	61	0.35
	Crack	5.54 (0.01)	1.52 (0)	0.00 (0)	115.10 (49.46)	1.54 (0.47)	0.55 (0.40)	7.87 (8.26)	5.81 (1.25)	3.20 (2.13)	8.61 (2.82)	0.66 (0.43)	11	
	NoCrack	5.54 (0.01)	1.52 (0)	0.00 (0)	63.63 (21.62)	1.52 (0.35)	0.66 (0.39)	9.95 (4.08)	5.64 (1.08)	1.83 (1.44)	5.99 (1.83)	0.69 (0.43)	50	
BCCE-0	COM	5.55 (0)	1.44 (0)	1.51 (0)	71.07 (39.13)	1.55 (0.40)	0.88 (0.46)	10.00 (4.76)	5.77 (1.10)	1.97 (1.62)	6.16 (2.55)	0.84 (0.50)	61	0.41
	Crack	5.55 (0)	1.44 (0)	1.51 (0)	68.83 (40.15)	1.61 (0.45)	1.13 (0.33)	10.46 (4.90)	5.76 (1.20)	1.69 (1.63)	6.02 (2.63)	0.66 (0.46)	38	
	NoCrack	5.55 (0)	1.44 (0)	1.51 (0)	74.77 (37.97)	1.46 (0.26)	0.48 (0.33)	9.25 (4.52)	5.78 (0.95)	2.43 (1.58)	6.37 (2.44)	1.14 (0.41)	23	
BCCE-90	COM	5.54 (0.01)	1.51 (0.01)	0.00 (0)	73.20 (57.40)	1.54 (0.39)	0.69 (0.42)	11.91 (8.02)	5.69 (1.15)	2.07 (2.01)	6.03 (3.23)	0.77 (0.45)	59	0.55
	Crack	5.54 (0.01)	1.51 (0.01)	0.00 (0)	139.41 (88.30)	1.45 (0.21)	0.67 (0.35)	9.54 (11.78)	6.38 (1.33)	4.04 (2.99)	9.57 (4.74)	0.96 (0.51)	13	
	NoCrack	5.54 (0.01)	1.51 (0.00)	0.00 (0)	54.49 (23.31)	1.57 (0.42)	0.69 (0.44)	12.59 (6.62)	5.50 (1.03)	1.51 (1.18)	5.04 (1.90)	0.72 (0.42)	46	
BRVE-θ	COM	4.80 (4.94)	1.52 (0.11)	0.79 (0.50)	67.26 (30.37)	1.44 (0.29)	0.69 (0.48)	7.48 (6.32)	5.72 (1.24)	2.30 (1.68)	6.44 (2.29)	0.92 (0.46)	64	0.36
	Crack	10.04 (6.21)	1.49 (0.05)	0.97 (0.47)	91.82 (27.44)	1.40 (0.17)	1.05 (0.46)	12.09 (7.63)	6.39 (0.92)	2.38 (1.56)	6.60 (2.09)	0.81 (0.46)	18	
	NoCrack	2.75 (2.06)	1.54 (0.12)	0.71 (0.50)	57.65 (25.91)	1.45 (0.33)	0.55 (0.42)	5.67 (4.69)	5.46 (1.26)	2.27 (1.74)	6.38 (2.38)	0.97 (0.46)	46	
BCVE-θ	COM	4.67 (4.96)	1.53 (0.11)	0.67 (0.45)	59.53 (34.09)	1.63 (0.45)	0.75 (0.47)	8.80 (6.85)	5.61 (1.21)	1.71 (1.55)	5.77 (2.45)	0.84 (0.44)	66	0.42
	Crack	11.30 (5.95)	1.53 (0.04)	0.72 (0.46)	69.54 (25.47)	1.61 (0.42)	0.98 (0.44)	16.28 (5.40)	6.25 (1.39)	1.84 (0.89)	5.08 (1.47)	0.79 (0.47)	16	
	NoCrack	2.55 (1.72)	1.53 (0.12)	0.66 (0.45)	56.33 (36.06)	1.63 (0.46)	0.68 (0.46)	6.40 (5.41)	5.40 (1.09)	2.07 (1.61)	5.99 (2.66)	0.86 (0.43)	50	

Table 7.5: Summary of the mean and standard deviation (S.D) values of the simulated data sets and the original data set "Origin" for Al-Si-Sn. COM denotes the complete set of simulated data. "Crack" and "no crack" class denotes the breakdown of their class distributions. NOTE: the values of the boundary cells are not considered here and some slight rounding errors may appear, this is due to conversion from simulated data to FBT data.

Components	Simulated parameters	Results	Summary
Cell Area (C.A) & Local Area Fraction (L.A.F)	1. Fix O.A + Random Distribution (BRCC)	• The crack class appears to have low and high values of the C.A (APP B2.1, fig1) . The mean and S.D value of the crack class is larger (table 7.4). • The crack class appears to have low and high values of the L.A.F (APP B2.1, fig2) . The mean value of both class are fairly similar with crack class having a higher standard deviation (S.D) (table 7.4).	• Given a fixed O.A, the effect of the cell area and L.A.F is difficult to asses. However, object clustering tends to initiate more cracks. • When the O.A is varied, the clustering effects seems to be shielded by the effect of the L.A.F. • Higher consistency (i.e. more 10/10 cases) of larger O.A causing crack initiation were picked up when the object size is varied.
	2. Fix O.A + Clustered Distribution (BCCC)	• Similar observation as for BRCC was made for the case of the C.A (APP B2.1, fig 3). • Similar observation as for BRCC was made for the case of the L.A.F (APP B2.1, fig4). However, the mean and S.D values are observed to be higher for the crack class. • Given that the O.A are fixed, more cracks were initiated for the clustered case (i.e. Random (BRCC) – 10% (APP B1, fig 1) and Clustered (BCCC) – 30% (APP B1, fig 2)	
	3. Vary O.A + Random Distribution (BRVC)	• Crack class has large O.A, C.A and L.A.F (APP B2.1, fig 5,6,7).	
	4. Vary O.A + Clustered Distribution (BCVC)	• Crack class has large O.A, C.A and L.A.F similar to that observed in BRVC. • The number of cracks initiated is not significantly different between both simulations when the O.A varies (i.e. Random (BRVC) – 33% (APP B1, fig 3) and Clustered (BCVC) – 35% (APP B1, fig 4).	
Object Angle (O.Ang) VS Cell Angle (C.Ang)	5. Fix O.A + Random Distribution + Angle 0 (BRCE-0°)	• The crack class has large C.Ang (APP B2.2, fig 8). • Crack initiation sites are 61% of particles (APP B1, fig 5).	• When the O.Ang is parallel to the loading axis, crack initiation is more likely to occur than when it is perpendicular. • The effect on crack initiation of the object distribution is not clear (i.e. BRCE-0 – 61% , BCCE-0 – 62% and BRCE-90 – 18% , BRCE-90 –22%) • The effect seen here that crack class tends to have either low or high values of the C.Ang is reflected in our SUPANOVA model (fig 6.5c).
	6. Fix O.A + Random Distribution + Angle 90 (BRCE-90°)	• The crack class appears to have more low and high values of the C.Ang (APP B2.2, fig 9) . • Crack initiation sites are 18% of particles (APP B1, fig 6).	
	7. Fix O.A + Clustered Distribution + Angle 0 (BCCE-0°)	• The crack class has large C.A. • Crack initiation sites are 62% of particles (APP B1, fig 7).	
	8. Fix O.A + Clustered Distribution + Angle 90 (BCCE-90°))	• It is difficult to assess which class distribution is significant in C.Ang (APP B2.2, fig 10). The mean and S.D values for both classes are fairly similar (table 7.4). • Crack initiation sites are 22% of particles (APP B1, fig 8).	

Table 7.6a: Summary of results for Al-Si-Sn obtained from the simulated data set produced to enhance model interpretability.

Components	Simulated parameters	Results	Summary
L.A.F Vs Mean Near Neighbour Distance (d_{Mean})	1. Fix O.A + Random Distribution (BRCC)	• There appears to be an inverse relationship between L.A.F and d_{Mean} as the crack class appears to lie on a slightly higher line (APP B2.3, fig 11) (i.e. for a given L.A.F, d_{Mean} slightly higher)	• The relationship between the L.A.F and d_{Mean} can be seen to be an inverse exponential (APP B2.3, fig 11 and fig 12). • Given a fixed O.A, the crack class either lies on the region where it has small L.A.F with large d_{Mean} or high L.A.F with low d_{Mean}. • When the O.A is varied, both d_{Mean} and L.A.F for the crack class has now shifted to higher values. This is due to the increase in large L.A.F, which is likely to be attributed to larger objects.
	2. Fix O.A + Clustered Distribution (BCCC)	• Similar to that observed in BRCC.	
	3. Vary O.A + Random Distribution (BRVC)	• The d_{Mean} for the crack classes has now shifted to higher values (APP B2.3, fig 12)	
	4. Vary O.A + Clustered Distribution (BCVC)	• Similar to that observed as in BRVC.	
O.Ang Vs Nearest Neighbour Distance (d_{Min}) Vs Nearest Neighbour Angle (N.N.Ang)	5. Fix O.A + Random Distribution + Angle 0 (BRCE-0°)	• The bivariate plot between N.N.Ang and d_{Min} shows that the crack class tends to have low d_{Min} (APP B2.4, fig 13).	• When the O.Ang is set parallel to the loading axis, the crack class has low d_{Min} (range from 0-3). • On the contrary when the O.Ang is set perpendicular to the loading axis, the clustered objects show that the crack class has d_{Min} are well distributed. • From the SUPANOVA model, it can be seen that as the O.Ang, d_{Min} and N.N.Ang get larger, crack initiation is unlikely.
	6. Fix O.A + Random Distribution + Angle 90 (BRCE-90°)	• The bivariates plots between N.N.Ang and d_{Min} show that the crack class d_{Min} are well distributed (APP B2.4, fig 14).	
	7. Fix O.A + Clustered Distribution + Angle 0 (BCCE-0°)	• The bivariate plot between N.N.Ang and d_{Min} shows that the crack class again has low d_{Min} (APP B2.4, fig 15). Further observation of the histogram plot of N.N.Ang distribution shows that the no crack has larger values (APP B2.4, fig 16).	
	8. Fix O.A + Clustered Distribution + Angle 90 (BCCE-90°)	• The bivariates plots between N.N.Ang and d_{Min} show that the crack class has large d_{Min} (APP B2.4, fig 17).	

Table 7.6b: Continued from Table 7.6a.

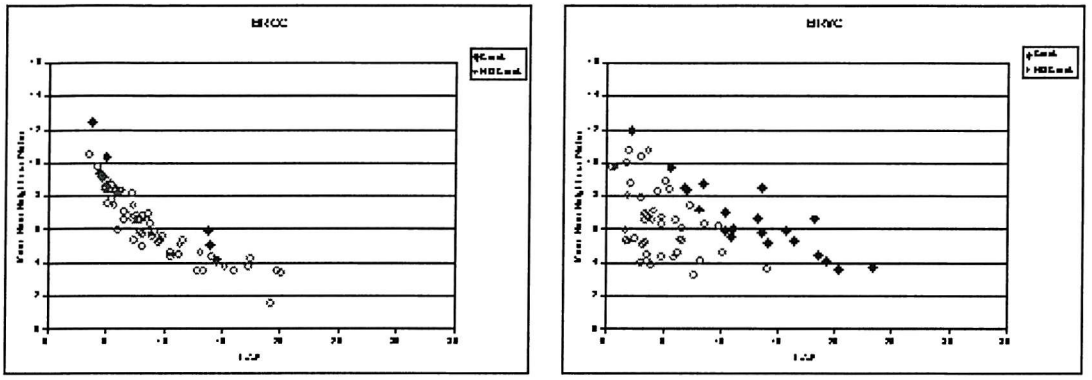
Components	Simulated parameters	Results	Summary
Overview	1. Vary O.A + Vary object Angle +Random Distribution (BRVE-θ)	• Crack class has a large O.A • Crack class has a large O.Ang • Crack class has a large C.A • Crack class has a small Cell Aspect Ratio ($C.A_r$) • Crack class has a large C.Ang • Crack class has a large L.A.F • Crack class has more near neighbours (N.N.N) • It is difficult to assess both class distributions of the d_{Min} (APP B2.5, fig 18) even with the mean and S.D values (table 7.4). • It is difficult to assess both class distributions of the d_{Mean} (APP B2.5, fig 19) even with the mean and S.D values (table 7.4). • Crack class has smaller N.N.Ang. • 28% crack initiation sites observed (APP B1, fig 9) • The bivariate plot for C.Ang and O.Ang show that the crack class tends to have large C.Ang (APP B2.5, fig 20) • The bivariate plot for d_{Mean} and L.A.F tends to show that as the d_{Mean} decreases, the L.A.F increases and this relationship occurs at higher d_{Mean} values for the case of the crack class (APP B2.5, fig 21) • The bivariate plot for N.N.Ang and d_{Min} show that the crack class tends to have smaller N.N.Ang (APP B2.5, fig 22).	• It is clear that the crack class has large O.A, O.Ang, C.A, C.Ang, L.A.F, more N.N.N and a smaller N.N.Ang. • When the object distribution is random, the d_{Min} and d_{Mean} are difficult to assess. However, when the object distribution is clustered (BCVE-θ), the crack class has smaller d_{Min} and smaller d_{Mean}. This also produces a similar $C.A_r$ for class. • Upon varying the O.A, the clustered distribution does not necessarily have more crack initiations observed. This implies that the O.A is a more important factor than any clustering effect (BRVE-θ - 28% and BCVE-θ - 24%) • The bivariates for the C.Ang and O.Ang show that the crack tends to have large C.Ang (value above 0.8 which is approximately 45°) and between d_{Mean} and L.A.F an inverse relationship at higher d_{Mean} values for the case of the crack class. • The bivariate plot for N.N.Ang and d_{Min} for the random object distribution show that the crack class tends to have smaller nearest N.A.Ang while for the case of the clustered object distribution , the crack class was observed to have small d_{Min}.

Table 7.6c: Continued from Table 7.6a.

Components	Simulated parameters	Results	Summary
	1. Vary O.A + Vary object Angle + Clustered Distribution (BCVE-θ)	• Crack class has a large O.A • It is difficult to assess both class distributions for the O.Ang (APP B2.5, fig 23). The mean values suggest that the crack class has a larger O.Ang (table 7.4). • Crack class has a large C.A • It is difficult to assess both class distributions for $C.A_r$ (APP B2.5, fig 24) even the mean and S.D values are fairly similar (table 7.4). • Crack class has a large C.Ang • Crack class has large L.A.F • Crack class has more N.N.N • Crack class has smaller d_{Min} • Crack class has smaller d_{Mean} • Crack class has smallerN.N.Ang. • 24% crack initiation sites (APP B1, fig 10) • The bivariate plots for C.Ang and O.Ang show that cracks tend to have large C.Ang (APP B2.5, fig 25) • The bivariate plots for d_{Mean} and L.A.F show that as the d_{Mean} decreases, the L.A.F increases and this relationship occurs at higher d_{Mean} values for the case of the crack class (APP B2.5, fig 26) • The bivariate plots for N.N.Ang and d_{Min} show that the crack class tends to have smaller d_{Min} (APP B2.5, fig 27).	• See Above

Table 7.6d: Continued from Table 7.6a.

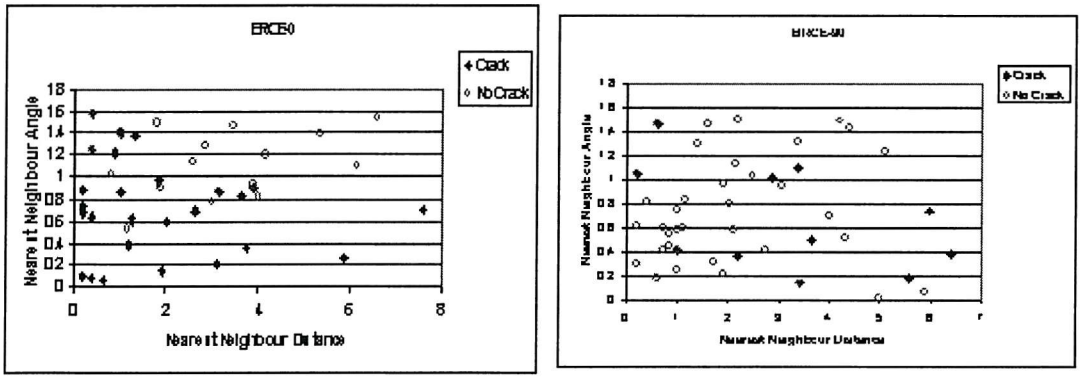
Bivariate plots of L.A.F Vs d_{Mean}



a.) BRCC - O.A are FIXED b.) BRVC - O.A are VARIED

Figure 7.6: The bivariate plots of d_{Mean} and L.A.F. Their relationship can be seen as an inverse exponential trend. When the O.A is fixed (a), this relationship is not obvious as compared to the case when O.A is varied (b.)

Bivariate plots of N.N.Ang Vs d_{Min}



a.) BRCE-0 - O.Ang parallel to the loading axis b.) BRCE-90 - O.Ang perpendicular to the loadi

Figure 7.7: The bivariate plots of N.N.Ang and d_{Min}. a.) when the O.Ang is parallel (or large O.Ang) to loading axis, the "crack" class tends to lie on small d_{Min} (range between 0-2.5). b.) when the O.Ang is perpendicular (or small O.Ang) to loading axis, the "crack" class tends to be well distributed along d_{Min}

7.4 Relationship between the results from SUPANOVA model and simulated data

Chapter 6.2.1 and 6.3.2 provides a detailed description of the results obtained from the SUPANOVA model. The previous section in this chapter described the results from our simulated particle distributions. Now, let us compare the two sets of results together in each application (i.e. ADI and Al-Si-Sn).

ADI

Let us begin with ADI. The effect of the univariate components selected from the SUPANOVA is fairly clear cut. It shows that as C.A, L.A.F, N.N.N gets larger, cracks will initiate. From the simulated data, we can see that the effect of C.A is not independent of O.A and L.A.F sizes as when O.A is held constant, the large C.A no longer predicts crack initiations (see App A2.1, Fig. 1). Although inspection of means indicates the crack class tends to have a large C.A, the standard deviation is very high, thus large C.A alone does not uniquely specify crack initiation as shown in our simulation. For the case of the L.A.F, the SUPANOVA model results tallies with our simulated model (see App A2.1, Fig. 5), as we can clearly see that large L.A.F tends to initiate cracks. The final univariate is the N.N.N. Again, the SUPANOVA model results tally with our simulated model as we can see that clustering (see App A1, Fig. 2) initiates more cracks than randomly distributed objects (see App A1, Fig. 1). Furthermore, we observed from the simulated data set that when the O.A is fixed, the significance of the clustering effect can be visualised easily.

The bivariate component from the SUPANOVA model indicates that as O.Ang and d_{Min} gets larger, cracks are less likely to initiate. The simulated data tallies with this finding (i.e. when the O.Ang is perpendicular to the loading axis (APP A1, Fig. 5), cracks are unlikely to initiate compared to those that are parallel (App A1, Fig. 6)). Finally, the bivariate plot of d_{Mean} and N.N.Ang from the SUPANOVA model indicates that as both N.N.Ang and d_{Mean} are large, cracks are likely to initiate. The d_{Mean} value depends on many factors, including N.N.N and their spacing, where it can be considered to reflect clustering (although not unambiguously) and a high d_{Mean} may reflect a relatively unclustered situation,

which may allow for the positioning of the N.N to be more influential in affecting the central graphite nodule. Our simulated data indicates that when the object is randomly distributed (App A2.3, Fig. 13) the "crack" class have a large N.N.Ang.

Al-Si-Sn

The above analysis for the univariate components (i.e. the C.A and L.A.F) in the ADI is also applicable to the case of Al-Si-Sn. Next, we will examine the bivariate and trivariate components for the Al-Si-Sn. The bivariate plots of O.Ang and C.Ang from the SUPANOVA model shows a complex non-linear trend which indicates a large O.Ang and C.Ang are likely to initiate cracks (Fig. 6.5c). Our simulated data tallies with this finding, as we observed that as our O.Ang is set to be parallel to the loading axis (i.e. large), cracks are more likely to initiate (App B1, Fig. 5 and 7) compared to those that are perpendicular (App B1, Fig. 6 and 8). Also, we observed that when the O.Ang is set to be perpendicular to the loading axis, the "crack" class lies on either low and high values of C.Ang. These trends can be observed in the SUPANOVA figure 6.5.

For the next bivariate plot of L.A.F and d_{Mean} from the SUPANOVA model, a hyperplane of a concave shape is seen along the diagonal of both increasing axes (i.e. as both features increase) indicating a threshold (see Fig. 6.5d) for these two features. This threshold effect can be seen in our simulated data (App B2.3, Fig. 12), indicates that the "crack" class has large d_{Mean}. The trivariate components selected by SUPANOVA indicate that as all inputs increase (i.e. O.Ang, d_{Min} and N.N.Ang) it is unlikely to initiate cracks (Fig. 6.5e). This trend can be observed from our simulated data (App B2.4, Fig. 13).

Conclusion

The relationship between the results obtained from the SUPANOVA models mostly tallies with that of the simulated results accept for the case of the C.A. The SUPANOVA model indicates that large C.A tends to initiate cracks. On the other hand, we observed from our simulated model when the O.A is fixed large C.A does not necessarily initiate cracks. A better indication of crack initiation would be the L.A.F. Clustering in particle distributions has generally been shown to promote

crack initiation. Another example which is interesting is the O.Ang. When the O.Ang is parallel to the loading axis, we see that for both cases (ADI and Al-Si-Sn), crack initiation is more likely to occur compared to when the O.Ang is perpendicular to the loading axis. This is somewhat counterintuitive, as if crack initiation occurred by particle cracking, we might expect a particle aligned perpendicular to the tensile axis to crack more easily. However, in these 2 cases, the initiation mechanism is via decohesion, which appears promoted for interfaces aligned parallel to the loading axis. Decohesion may occur by a combination of peak tensile stress, peak hydrostatic stress and strain accumulation effects which may be affected differently by particles shape.

7.5 Summary

The FBT components selected by the SUPANOVA decomposition identifying crack initiation sites are inter-related and it is difficult to simulate variations in them systematically. However, we can use particle simulations which provide self-consistent distributions to assess those components which give rise to increased fatigue initiation. Four parameters in the particle simulations were varied systematically, namely, the object shape, the O.A, the object distribution and the O.Ang. The object shape allows the effect of the O.Ang to be eliminated (i.e. when it is circular in shape, there is no O.Ang). The O.A allows the effect of object size to be assessed and helps to identify which of the linked parameters (C.A, L.A.F) are determining initiation (i.e. L.A.F and not C.A). The object distribution allows the effect of clustering to be assessed. Finally, the O.Ang has been assessed for the two extreme situations, perpendicular to the loading axis or parallel to the loading axis. Varying this parameter systematically provides a better interpretability to our model produced by the SUPANOVA. For example, a fixed O.A shows the effect of clustering (i.e. clustering tends to lead to more cracks). When the O.A is varied, the significance of the clustering effect is shielded or outweighed by the effect of large objects. This is seen in both our data sets, given that they are quite different mechanical situations (soft particles in a hard matrix (ADI) and vice versa

(Al-Si-Sn)) this is intriguing, although it should be noted that a decohesion mechanism of fatigue initiation has been proposed for both cases. These are specific examples of how we have used systematic variations in simulated particle distributions to further assess the SUPANOVA classification model. Our analysis goes on further to make comparisons between the results obtained from the application of the SUPANOVA classification model to the simulated particle distributions and the input terms selected by the SUPANOVA model. These trends observed in the SUPANOVA model tally with most of the simulated data set except for the C.A. The simulated data shows that only large C.A with large L.A.F are the true fatigue crack initiation factors. O.Ang shows the same effect in the two applications considered, where particles with their major axis oriented parallel to the loading axis are more likely to initiate cracks. Generally speaking, similar relationships have been identified for the two applications - e.g. C.A, L.A.F. In both cases, more clustered particle distributions are predicted to initiate more fatigue cracks, as indicated by L.A.F, N.N.N dependencies. There are clearly more parameters within the particle simulation packages to be investigated in future work.

Chapter 8

Conclusions and Future Work

Most machine learning requires modifications to the cost function to incorporate misclassification costs and sampling bias costs in order to be used appropriately for imbalanced data. Their performance criteria (e.g. Amean or Gmean) may also be altered to be less sensitive to the skewness distribution of the classification rate (Gmean being more appropriate than Amean). A classification model with good prediction tends to be complex and therefore difficult to interpret. Interpretation is valuable in identifying which features are important in classifying behaviour and hence may help in identifying optimisation criteria. A parsimonious model can be made by model structure decomposition and sparse selection, hence providing an interpretable model. The SUPANOVA approach uses a spline kernel and ANOVA decomposition followed by a sparse selection of ANOVA terms to provide model interpretability. (Kandola et al. *1999, Christensen* et al. *2001, Gunn 1999) have successfully applied SUPANOVA for regression tasks in various materials science applications. We extend this work to the case of classification with imbalanced data to materials science fatigue failure problems.*

8.1 Summary of Work

This thesis focuses on (1) the classification and hence prediction of crack initiation sites in two automotive materials systems (2) producing an interpretive model, hence developing a new understanding of the relationship between the input features obtained from Finite Body Tessellation (FBT) and fatigue initiation. Two sets of fatigue initiation data which were obtained from automotive materials, (camshaft

(ADI) and plain journal bearing lining (Al-Si-Sn)), and using SUPANOVA developed for classification of imbalanced data. As in many real world problems, the crack initiation data was smaller than the "no crack" majority class. The results show that the extended approach for SVMs requires a sampling bias and if necessary, a higher misclassification cost for the minority class for a set of imbalanced data. The ratio between the imbalanced modification factor (i.e. L's) is an important parameter. The results from the Non-Standard Situation (NSS) SVM provide a good guide to the necessary ratio of the L's. However, fine tuning is required to obtain better results. In both data sets, a successful classification rate of both classes of at least 0.70 was obtained. The structure of this classification model is then decomposed to provide a parsimonious model to aid model interpretability. This is done using the SUPANOVA for classification for imbalanced data. The parsimonious model comprises a sum of subset components (6 components for both applications) which were selected out of a possible 512 and 1024 combinations for the ADI and the Al-Si-Sn respectively. The trends of the 6 input components selected by the model have been assessed in terms of the mechanistic understanding they provide for the fatigue initiation phenomena. Here, it has been possible to consider significant bivariate and trivariate interactions in addition to univariate effects which could also be picked out by simple approaches such as observing their means, standard deviations and simple visualisation plots. A simulated data set was then used to further visualise the effects and interrelationships of these 6 components selected for each case. For example, in both cases, the univariate function selected by our model, which shows that a large cell area (C.A) promotes crack initiation, has been further examined. It has been shown that it is not a good indicator of crack initiation by itself. It is necessary for both a large C.A and a large L.A.F to be present to initiate cracks. As such, this work has successfully picked up some of the significant features of the particle *distributions* that lead to fatigue initiation in these materials (e.g. clustering) allowing further optimisation of these microstructures by considering the model predictions on simulated particle distributions. There is a diversity of real world applications with similar problems,

(i.e. with imbalanced data where parsimonious models are desirable, as opposed to complex models). The work, therefore has a broad-based application potential.

8.2 Future Work

There are several areas of future work that can be extended from this research from both the modelling and materials point of view. They are listed as follows :

From the modelling point of view :

- The current work uses misclassification cost and sampling bias to tackle the problem of imbalanced data. Other approaches, such as clustering (e.g. Learning Vector Quantisation (LVQ) as described in chapter 5.1) can be used systematically to reduce the number of majority class samples. The reason for not using more data from the majority class in this work has been to enable computation efficiency.
- Rather than reducing the number of data which may lead to less true representation of our data distribution, a faster algorithm could be used. Work has already been carried out in the image processing field to increase computational efficiency in SVM such as chunking algorithms which essentially breaks the large data set into smaller subsets which are then combined (Osuna *et al.* 1996). The SVM is trained with an algorithm that starts with an arbitrary subset/'chunk' of training data, those support vectors are used to construct the hypothesis on the remaining training data and the points that violate the KKT conditions are added to the previous support vectors of the previous system to form a new chunk. A stopping criteria is then used to stop this procedure.
- The current work on SUPANOVA for imbalanced data requires four stages, described in chapter 5.5 and the best classification rate is obtained in a heuristic way. The classification rate used is based on the Geometric Mean (GMean). It would be interesting to incorporate this model selection into the loss function and then optimising the model can be done automatically.

- Setting the output of the SVM for classification to be probabilistic is advantageous as it allows for confidence in the determination of class membership. A brief description of work done based on probabilistic SVM follows. (Vapnik 1995) (decompose the feature space), (Wahba *et al.* 1999) (logistic link function), (Platt 2000) (logistic link function with sparse representation in place) and several authors (Sollich 2000, Kwok 1999, Seeger 1999) use the Bayesian framework. With a Bayesian framework, the training of SVM can be viewed as maximising posterior (MAP) solution to an inference problem (Kwok 1999). The problem associated with a Bayesian approach for probabilistic SVM lies in the difficulties involved in trying to normalise the prior (note: SVM prior is simply a Gaussian Process over latent function) (Sollich 2000). Recent work by (Tipping 2000) uses a unique prior defined by its data size and location of the training input. This is known as the relevant support vector machine. (Herbrich *et al.* 2001) eliminate the SVM prior to normalising using the Bayes point SVM. In this work, the prior is replaced by a spherical one (i.e. $||\mathbf{w}|| = 1$ uses only the spatial direction of the weight vectors which is important for classification).
- Transforming the SVM output to a probabilistic term and then using the SUPANOVA for decomposition would provide more meaningful interpretability of the value of the output (e.g. in the plot of the components selected, the y axis is an indication function where the absolute value has no significance but we use the "sign" to provide interpretability to our model (i.e. a "-" implies crack initiation and a"+" implies no crack initiation)).

From a metallurgist's view point :

- The discussion of the work here assumes that particles spacing has no direct effect on the matrix properties, however in the case of ADI the graphite nodule spacing may affect local concentration profiles and hence matrix properties which should be considered in further analysis.
- The difficulties of simulating the local clustering distribution of the particles (i.e. the near neighbour and its angle) is a current pit fall of the particle

simulation process. It would be interesting to incorporate simple algorithms so that specific location (e.g. N.N.Ang can also be specified) of the particles can be simulated.

- The Finite Body Tessellation (FBT) captures information on the distribution of the secondary phase and the morphology of the particles. This gives us our prior knowledge of the features that initiate fatigue cracks in our two sets of automotive material. Other techniques such as Finite Element Analysis (FEA) can also be used to investigate fatigue crack initiation. FEA requires prior knowledge of the material properties (e.g. Young's modulus of the individual/surrounding particles and their elastic-plastic material properties) and their testing condition (i.e. loading condition) to be specified correctly in order to obtain an accurate analysis. The analysis obtained from the FEA can provide insight into the local stress-strain fields indicating the region where, for example, when the strain is high and crack initiation is likely to occur. The FEA can then be compared with the SUPANOVA model predictions to confirm whether the particles selected experience the stress-strain conditions that will initiate fatigue cracks. This will provide independent correlation of the SUPANOVA model predictions.
- Our work on Al-Si-Sn assumes that the background and bordering secondary particles of the Si are the same. As such, only two classes are required to be classified as "crack" or "no crack" class. It might be interesting to investigate the effect of the bordering secondary particles, hence making a three class classification problem. A brief description about multi-class SVM can be obtained from chapter 3.8.1.

These are some of the areas that are worth investigating further, based on the results from this research work, although these ideas are not exhaustive.

Appendix A

ADI

A.1 Simulated Particle Distribution and their associated tessellation cells

ARCC

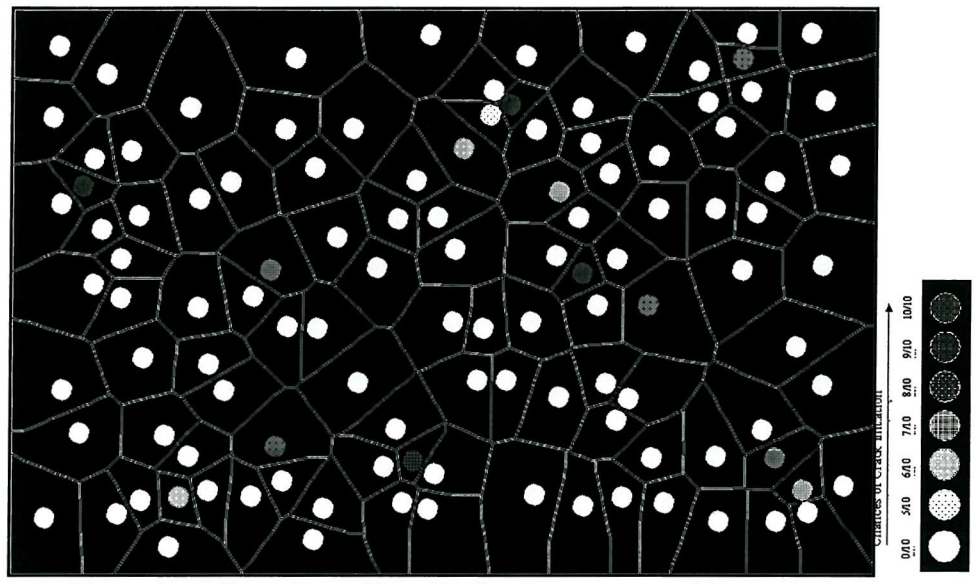


Figure 1: ADI, random object distribution, constant object area, circular shapes.

ACCC

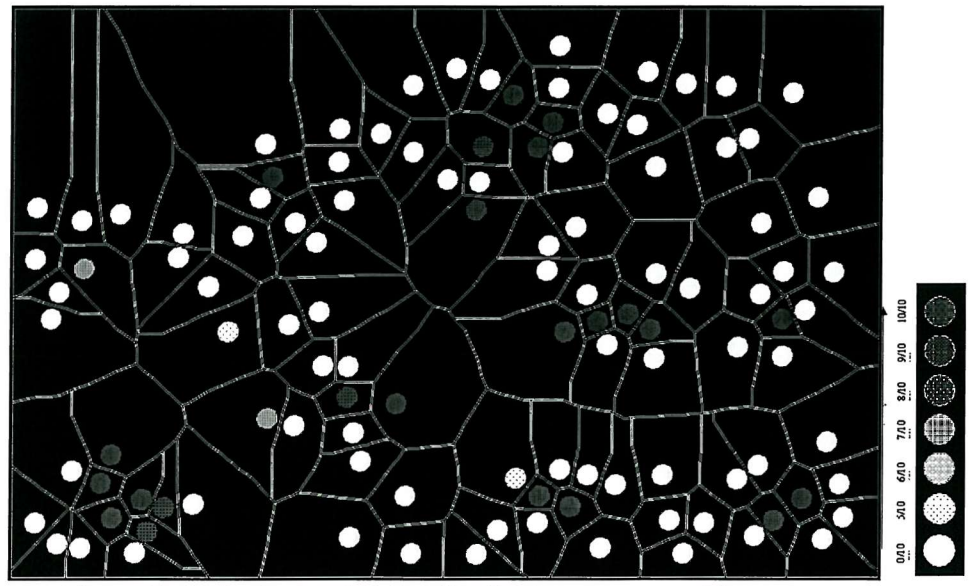


Figure 2: ADI, clustered object distribution, constant object area, circular shapes.

ARVC

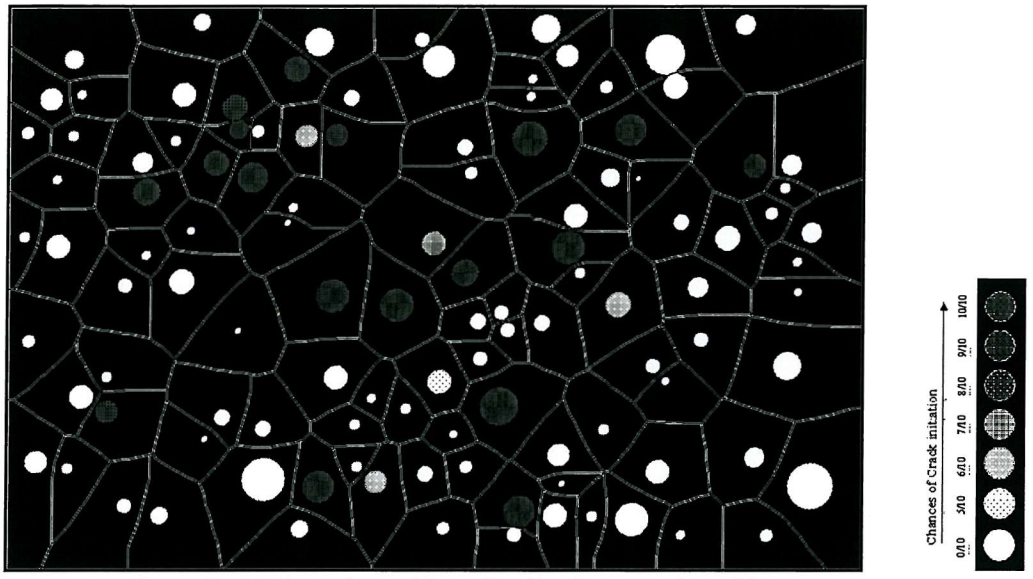


Figure 3: ADI, random object distribution, varying object area, circular shapes.

ACVC

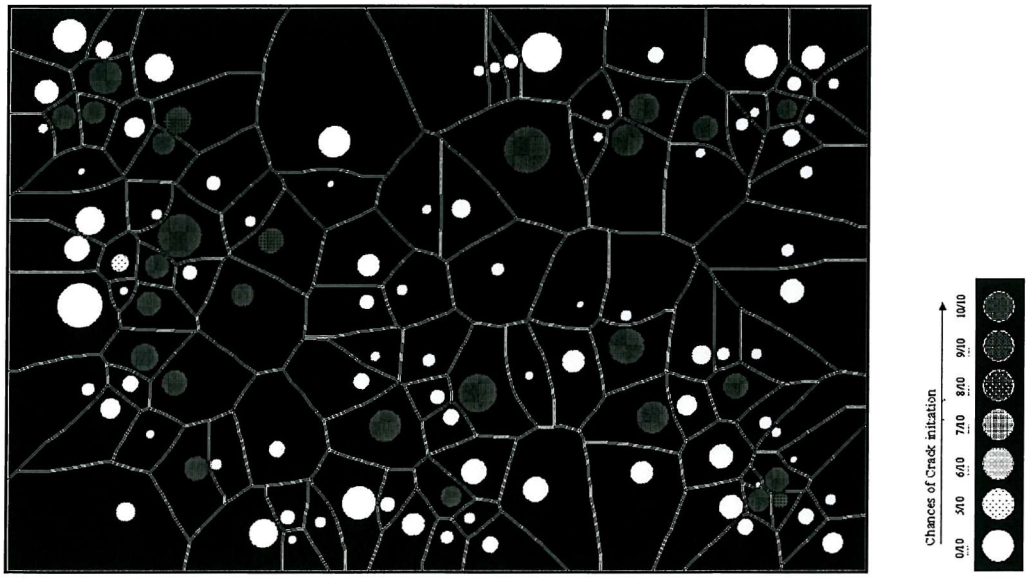


Figure 4: ADI, clustered object distribution, varying object area, circular shapes.

ARCE-90

Figure 5: ADI, fixed object area, random object distribution, constant object area, ellipse shapes at angle 90° to the loading axis.

ARCE-0

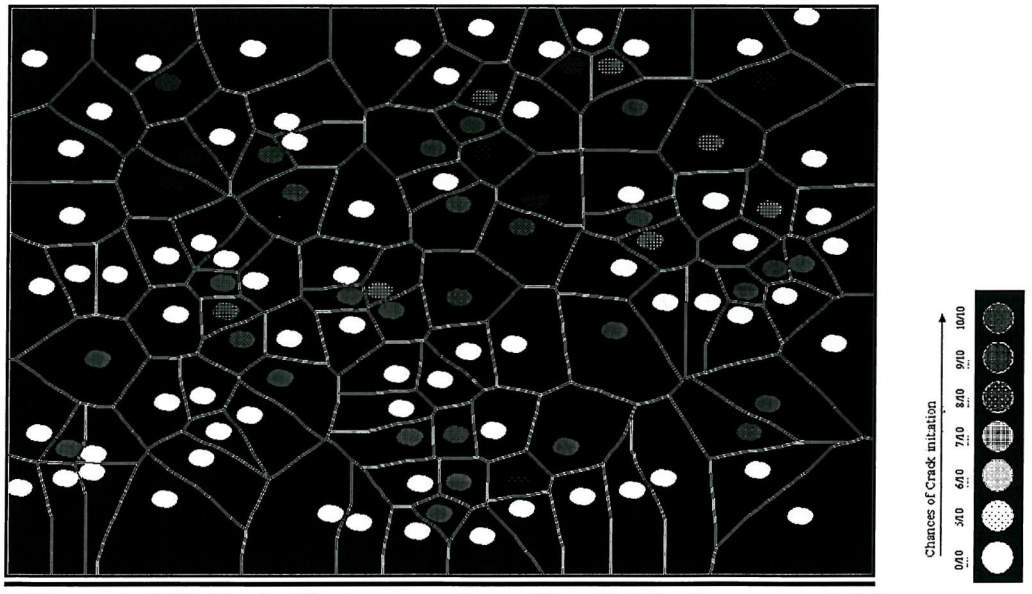


Figure 6: ADI, fixed object area, random object distribution, constant object area, ellipse shapes parallel to the loading axis.

ACCE-90

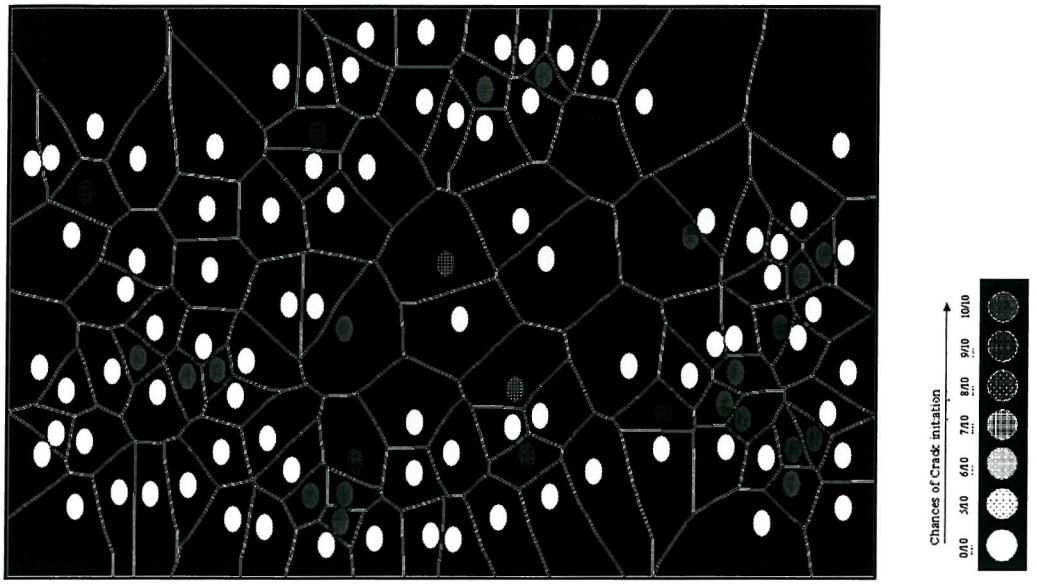


Figure 7: ADI, fixed object area, clustered object distribution, constant object area, ellipse shapes at angle 90° to the loading axis.

ACCE-0

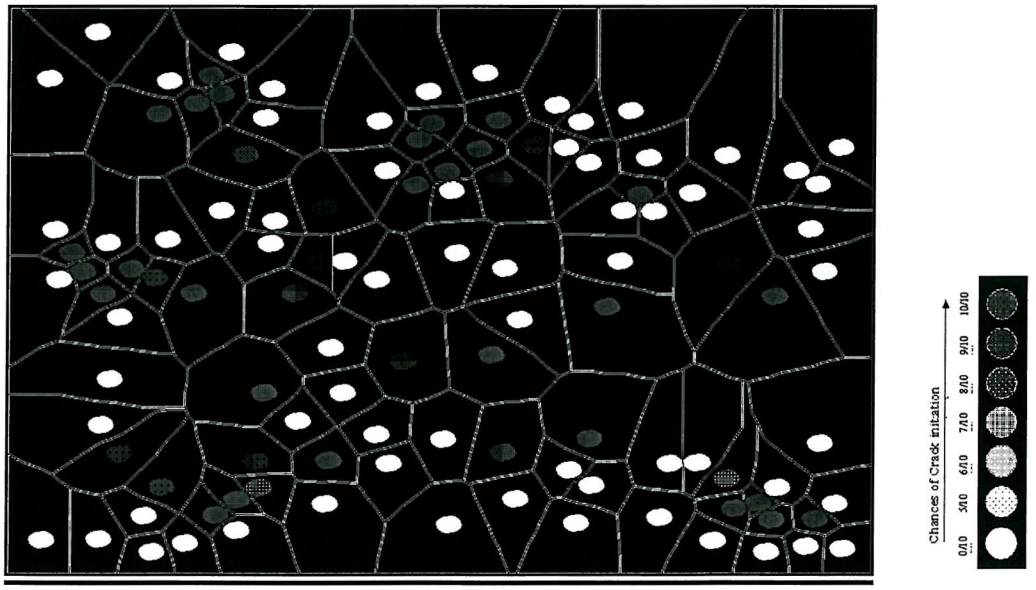


Figure 8: ADI, fixed object area, clustered object distribution, constant object area, ellipse shapes parallel to the loading axis.

ARVE-θ

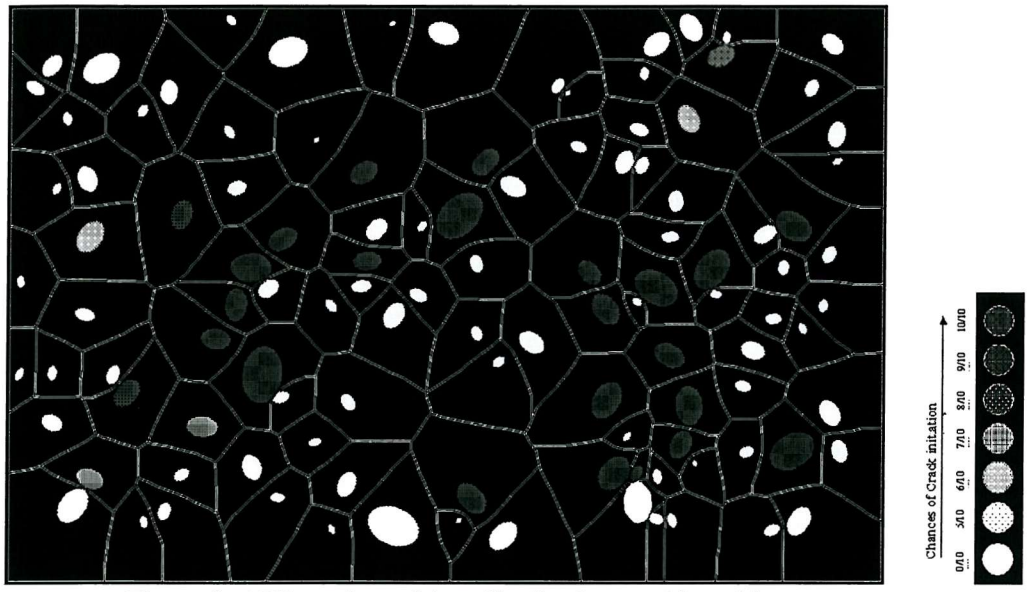


Figure 9: ADI, random object distribution, varying object area, ellipse shapes at angle θ to the loading axis.

ACVE-θ

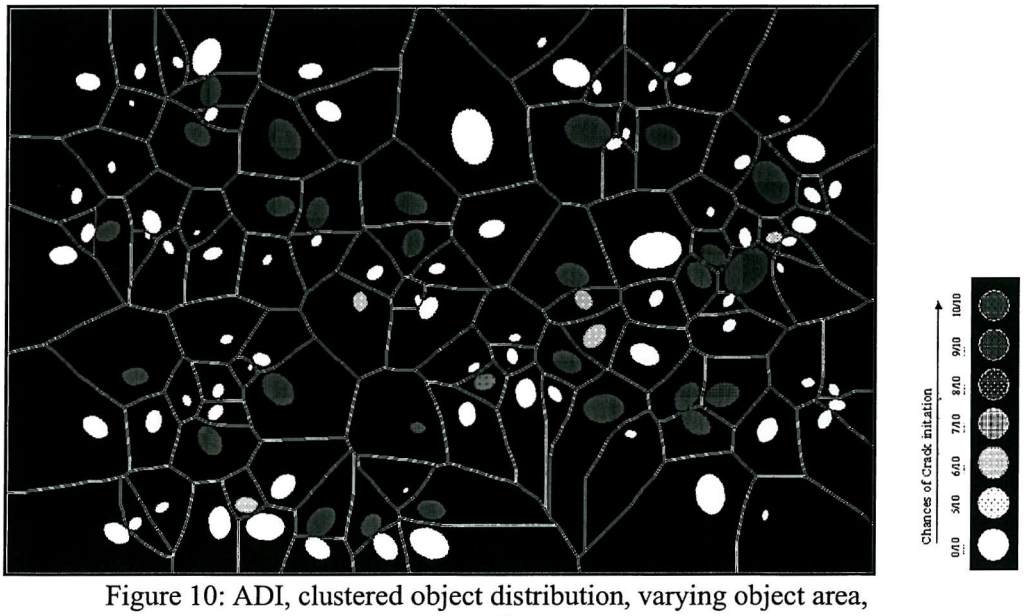


Figure 10: ADI, clustered object distribution, varying object area, ellipse shapes at angle θ to the loading axis.

A.2 Analysis of the Simulated Particle Distribution

A.2.1 Analysis of the Cell Area (C.A), Local Area Fraction (L.A.F), Number of Near Neighbour (N.N.N)

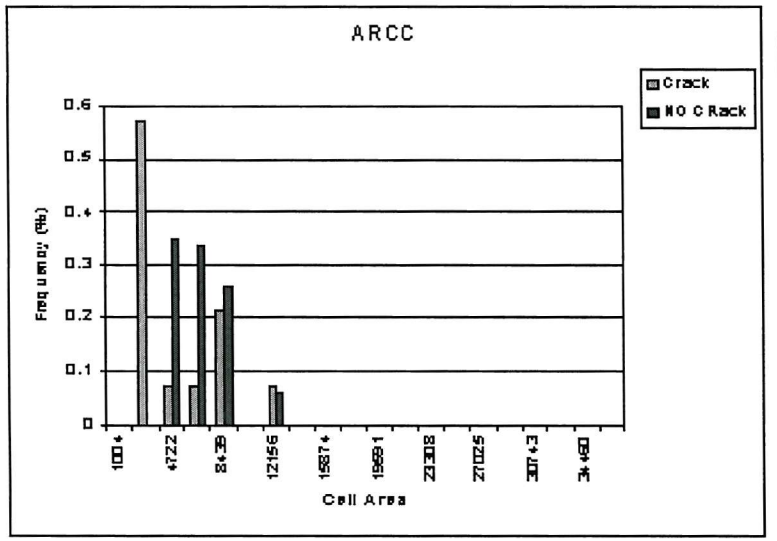


Fig 1 Histogram plots of ARCC - Cell Area

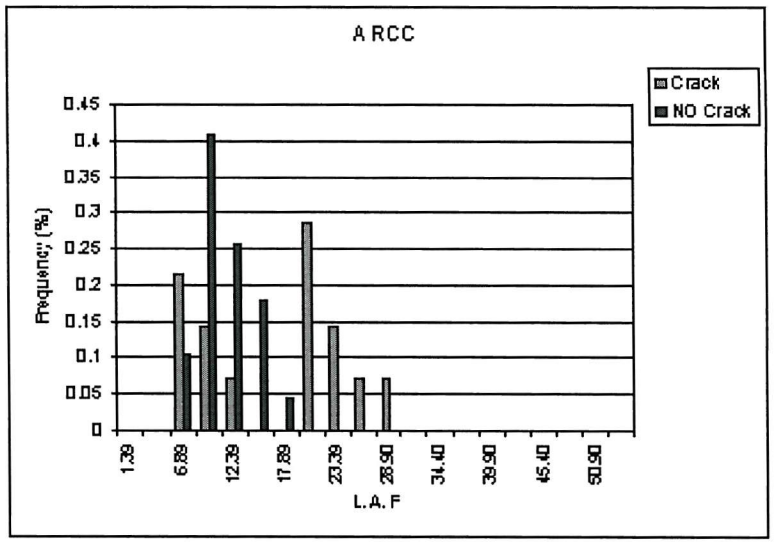


Fig 2 Histogram plots of ARCC - Local Area Fraction

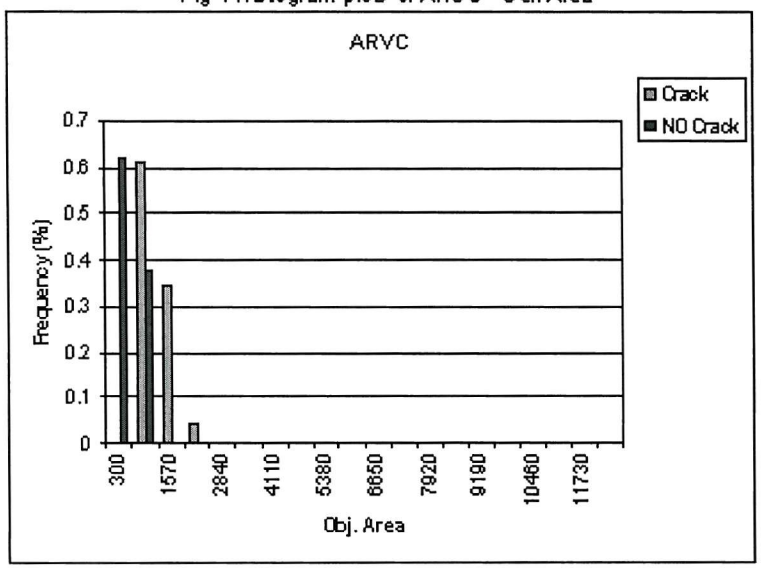


Fig 3 Histogram plots of ARVC - Object Area

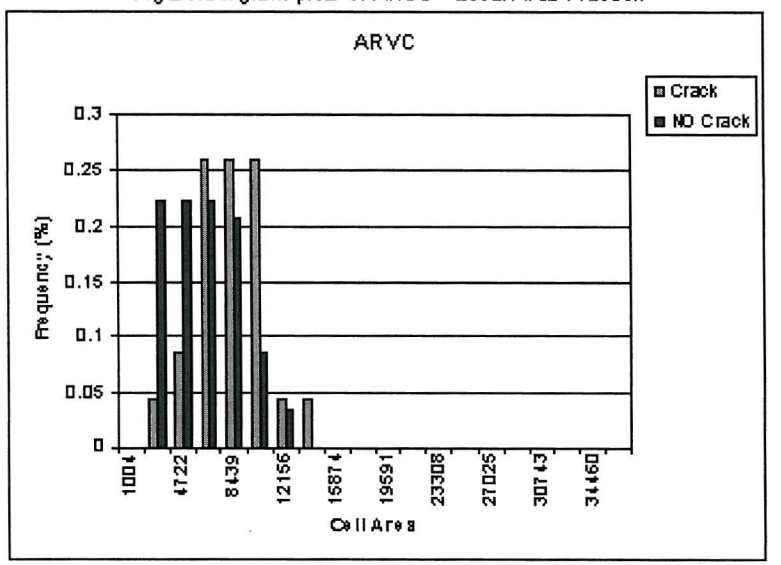


Fig 4 Histogram plots of ARVC - Cell Area

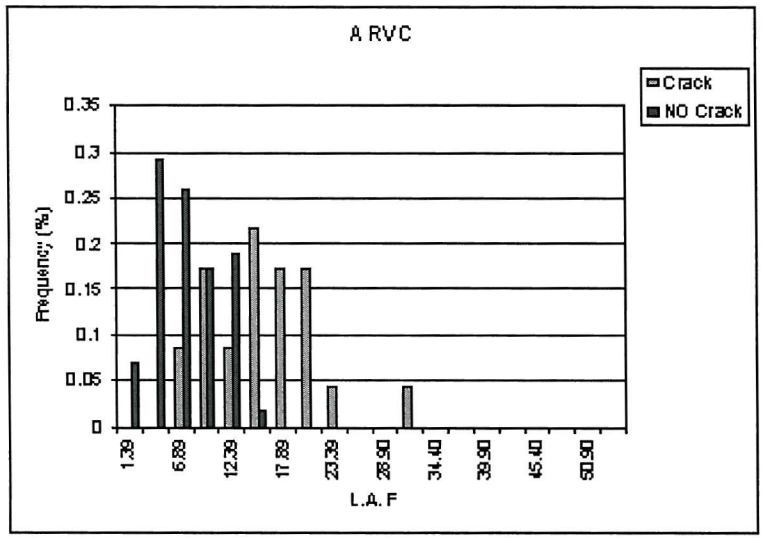


Fig 5 Histogram plots of ARVC - Local Area Fraction

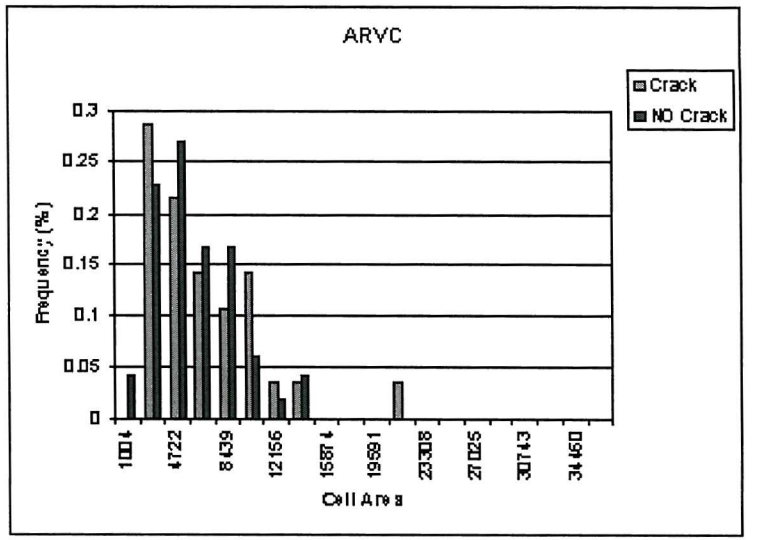


Fig 6 Histogram plots of ARVC - Cell Area

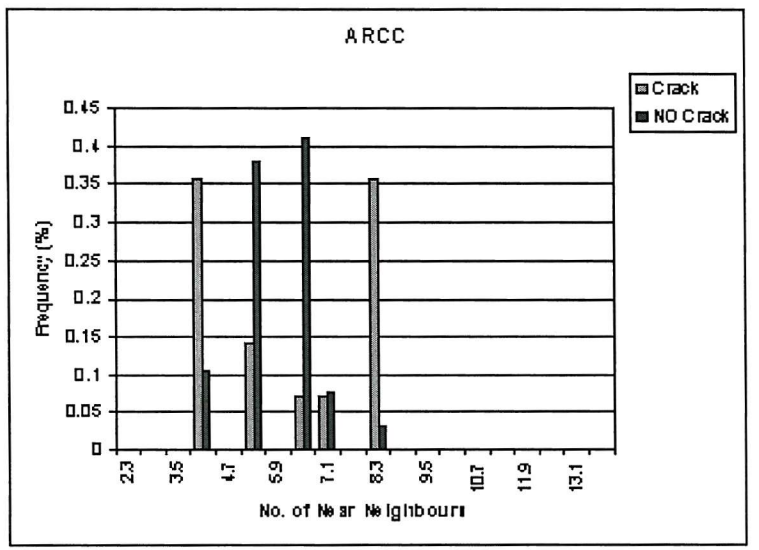


Fig 7 Histogram plots of ARCC - Number of Near Neighbours

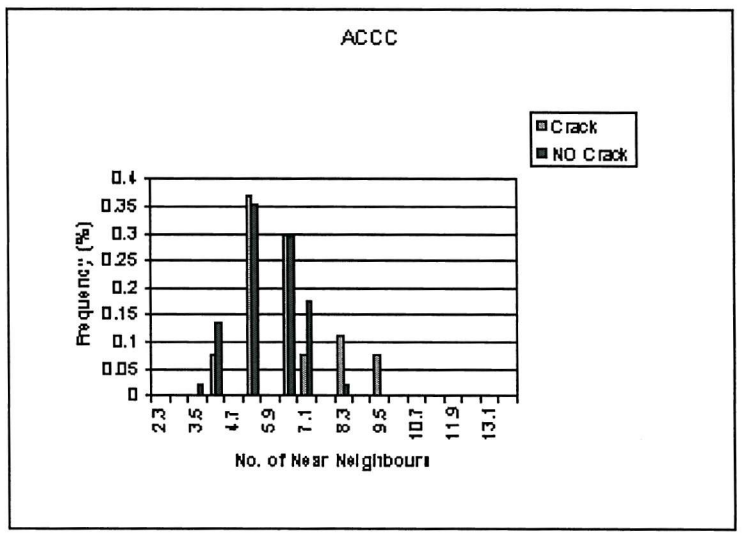


Fig 8 Histogram plots of ACCC - Number of Near Neighbours

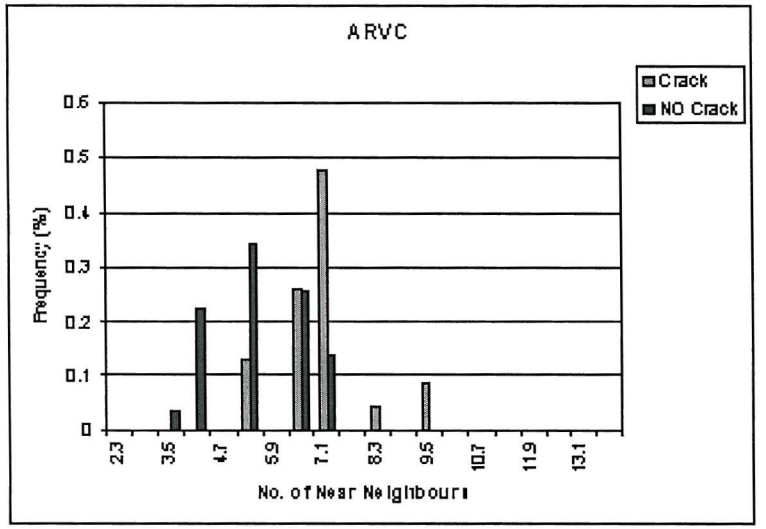


Fig 9 Histogram plots of ARVC - Number of Near Neighbours

A.2.2 Analysis of the Object Angle (O.Ang) Vs Nearest Neighbour Distance (d_{Min})

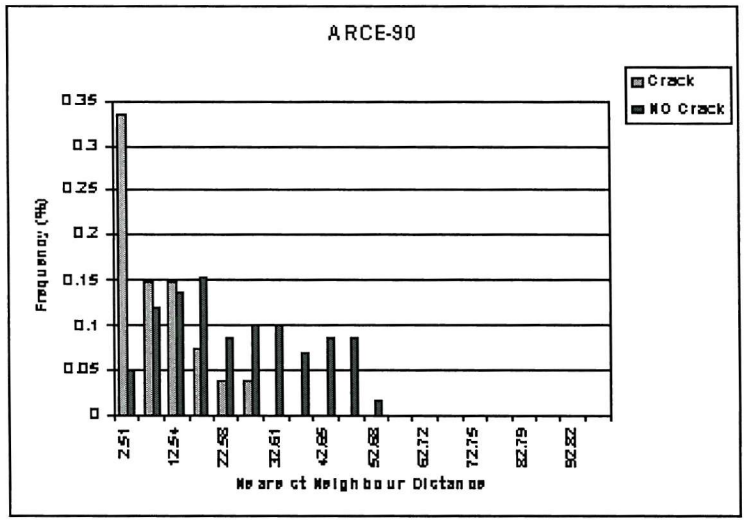


Fig 10 Histogram plots of ARCE-90 - Nearest Neighbours Distance

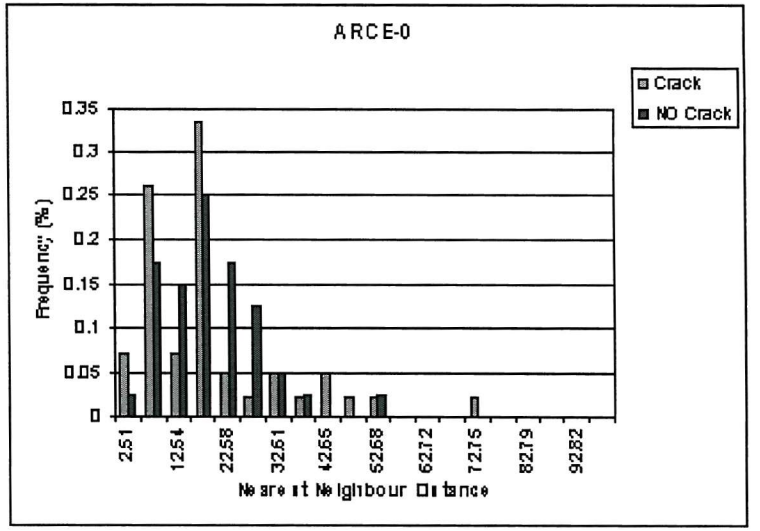


Fig 11 Histogram plots of ARCE-0 - Nearest Neighbour Distance

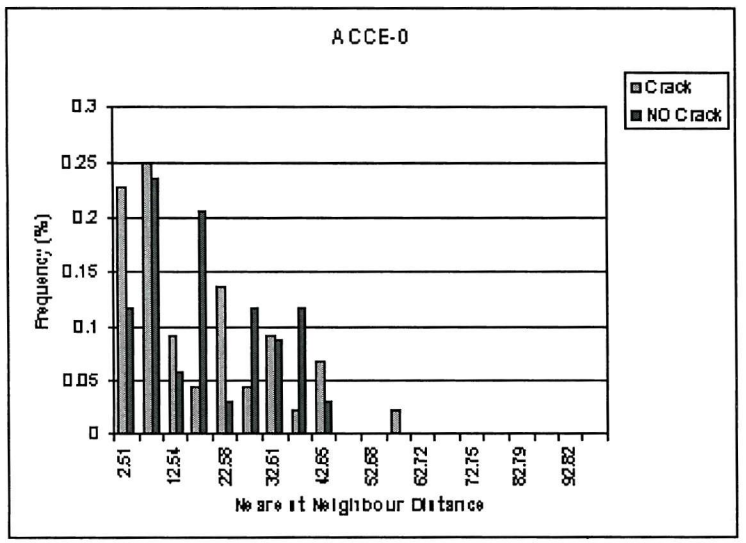


Fig 12 Histogram plots of ACCE-0 - Nearest Neighbour Distance

A.2.3 Analysis of the Mean Near Neighbour Distance (d_{Mean}) Vs Nearest Neighbour Angle (N.N.Ang)

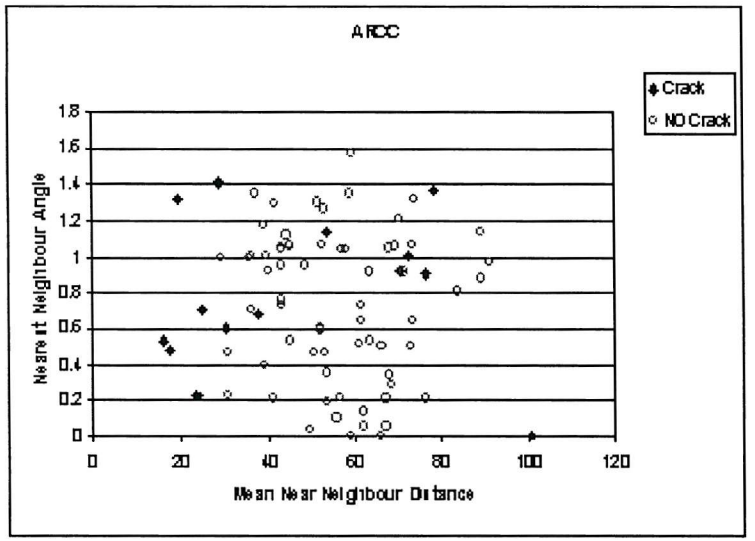


Fig 13 Bivariates plots of ARCC - N.N.Ang Vs d_{Mean}

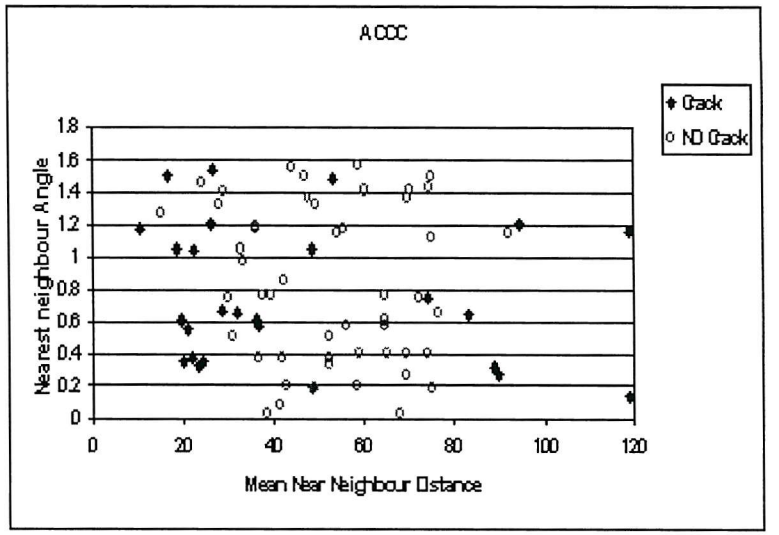


Fig 14 Bivariates plots of ACCC - N.N.Ang Vs d_{Mean}

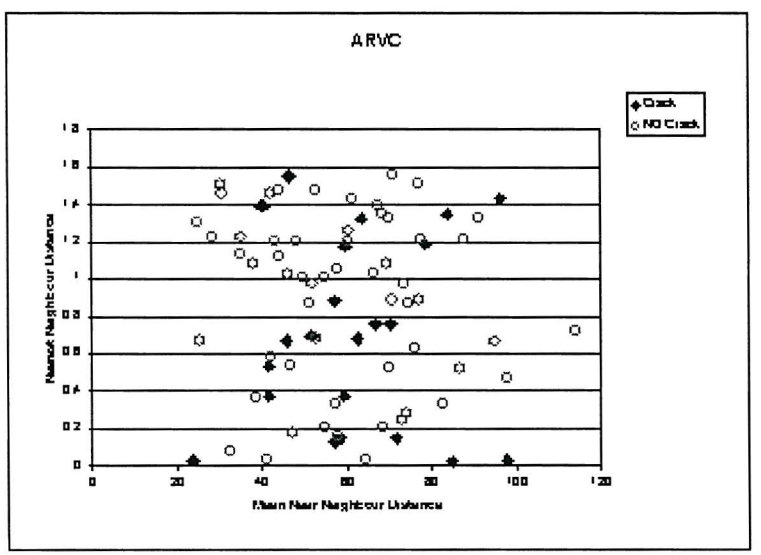


Fig 15 Bivariates plots of ARCC - N.N.Ang Vs d_{Mean}

A.2.4 Analysis of the Overview of the Simulated Data set

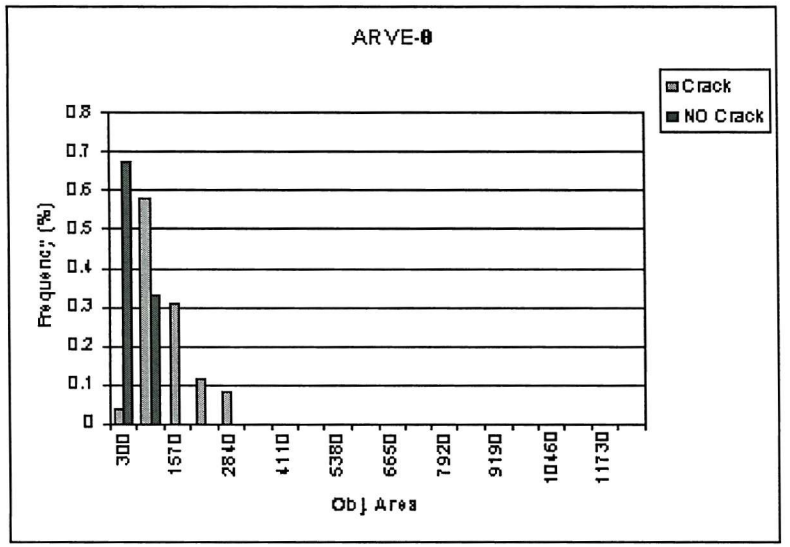


Fig 16 Histogram plots of ARVE-θ- object Area

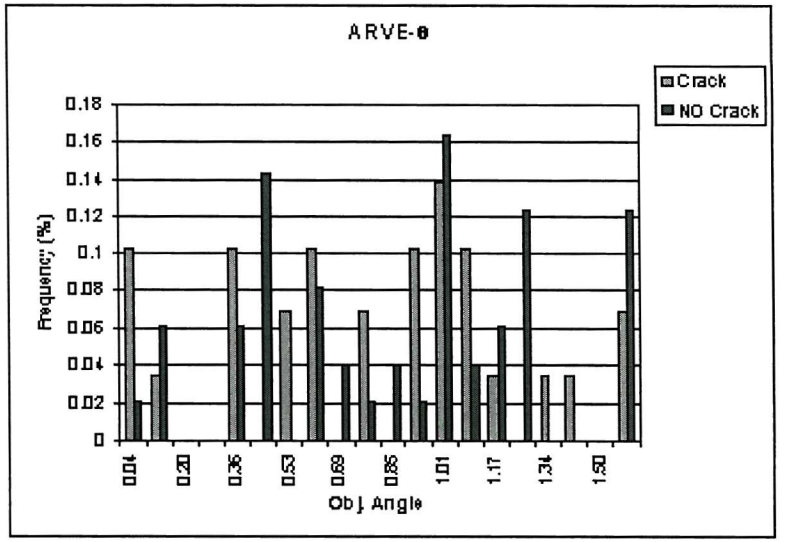


Fig 17 Histogram plots of ARVE-θ- object Angle

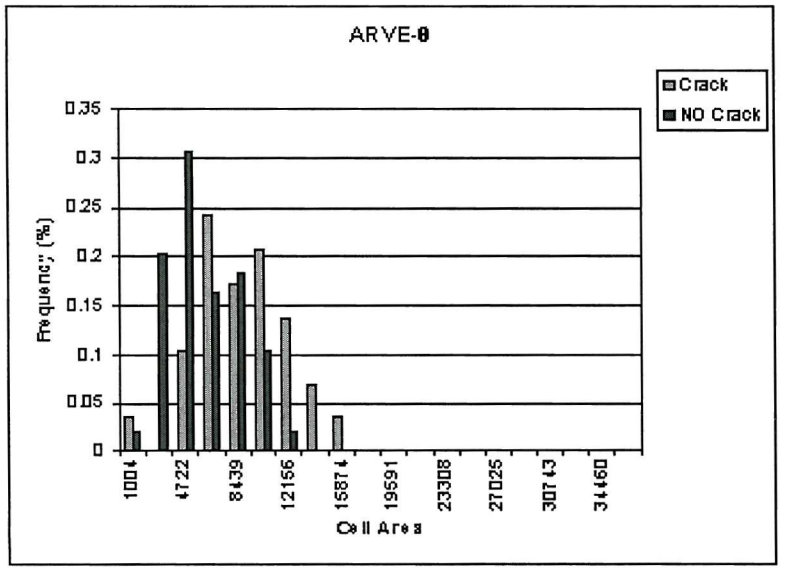


Fig 18 Histogram plots of ARVE-θ- Cell Area

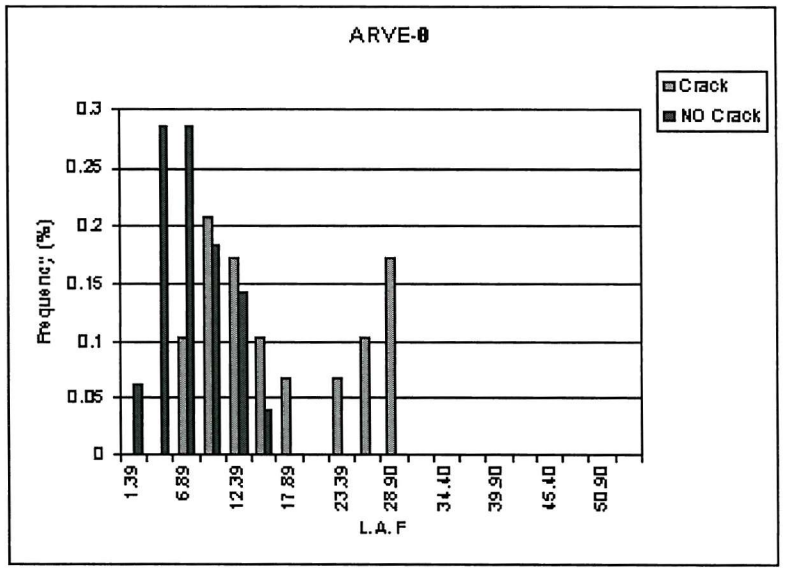


Fig 19 Histogram plots of ARVE-θ- L.A.F

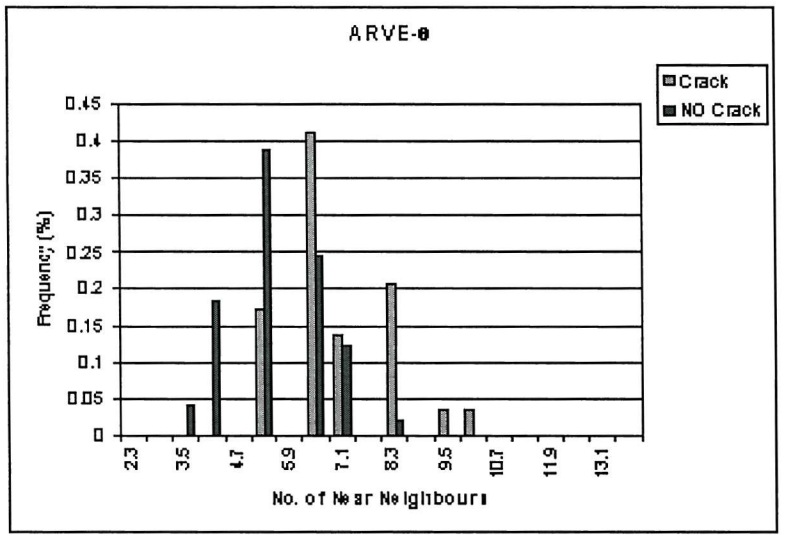


Fig 20 Histogram plots of ARVE-θ- Number of Near Neighbours

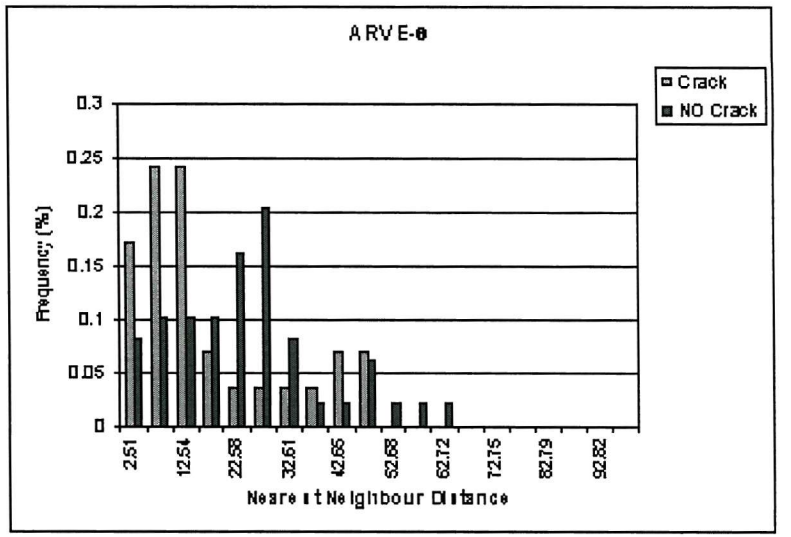


Fig 21 Histogram plots of ARVE-θ- Nearest Neighbour Distance

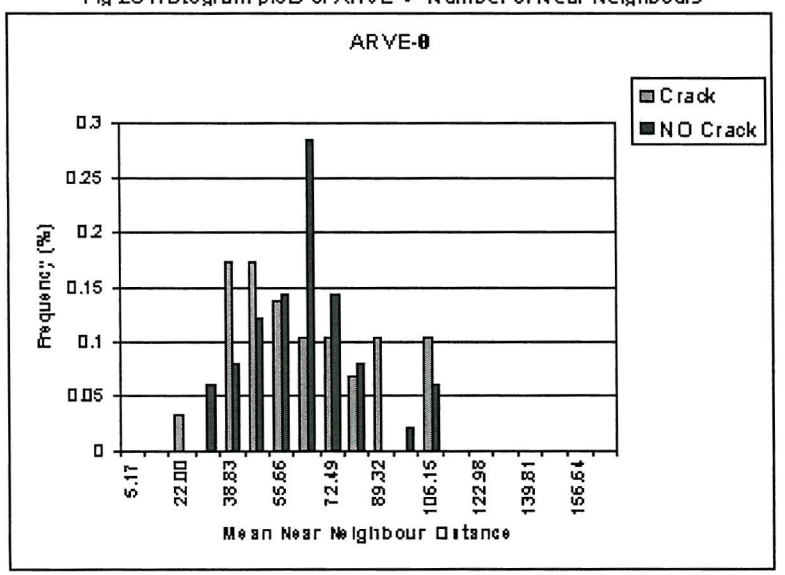


Fig 22 Histogram plots of ARVE-θ- Mean Near Neighbour Distance

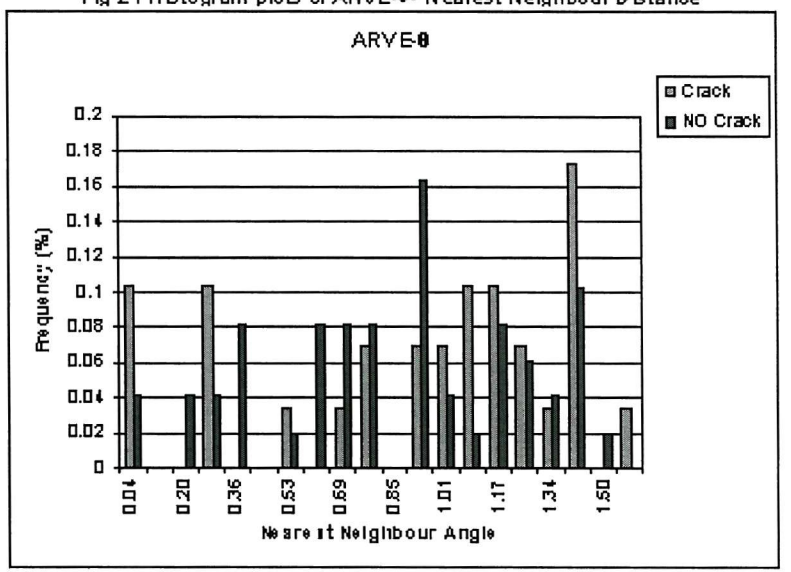


Fig 23 Histogram plots of ARVE-θ- Nearest Neighbour Angle

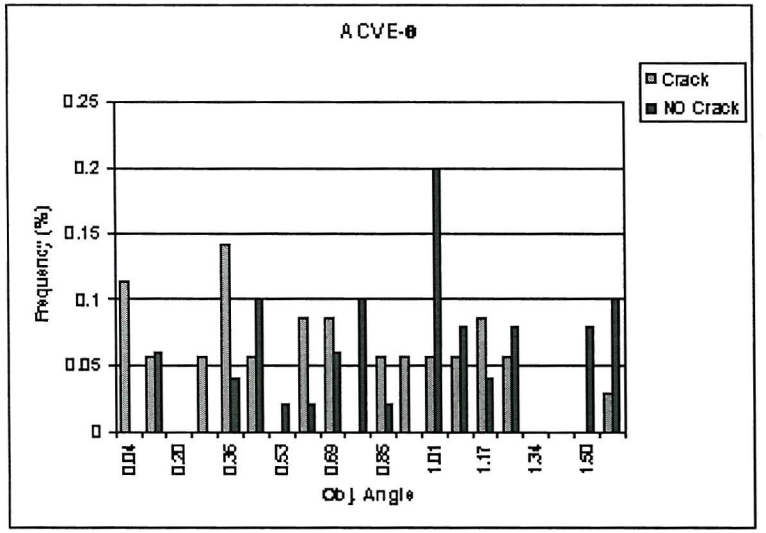


Fig 24 Histogram plots of ACVE-θ- Object Angle

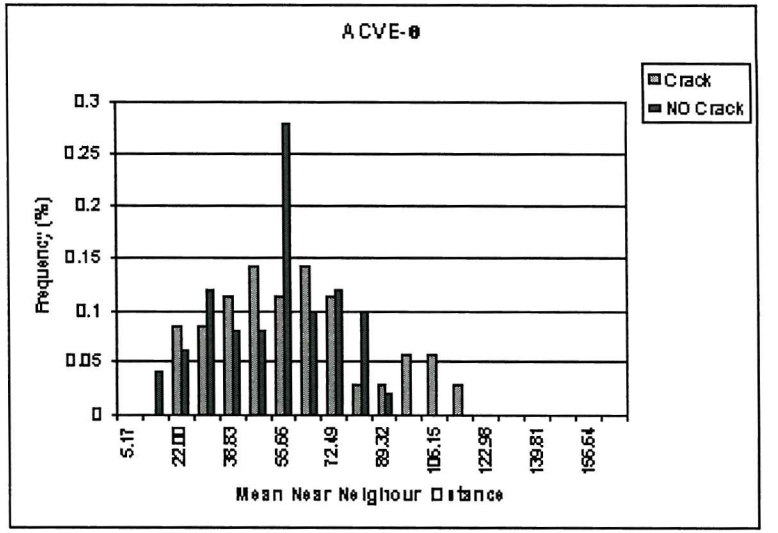


Fig 25 Histogram plots of ACVE-θ- Mean Near Neighbour Distance

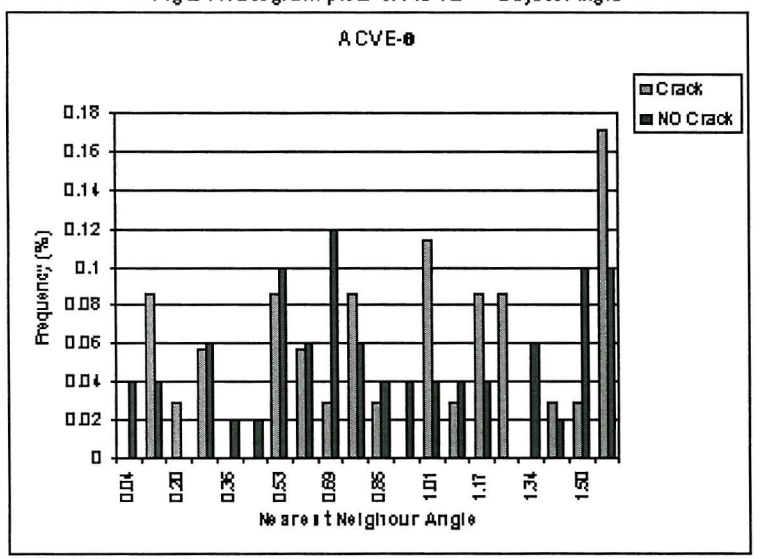


Fig 26 Histogram plots of ACVE-θ- Nearest Neighbour Angle

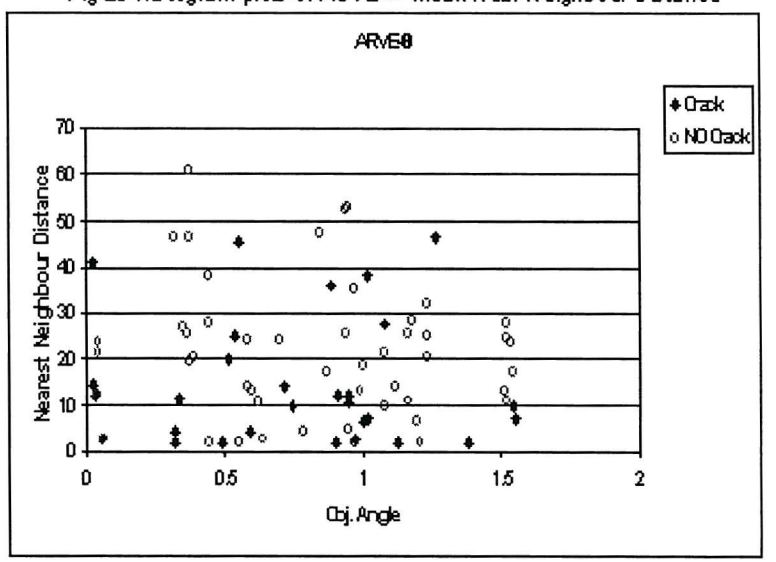


Fig 27 Bivariates plots of ARVE-θ- d_{MIN} Vs Object Angle

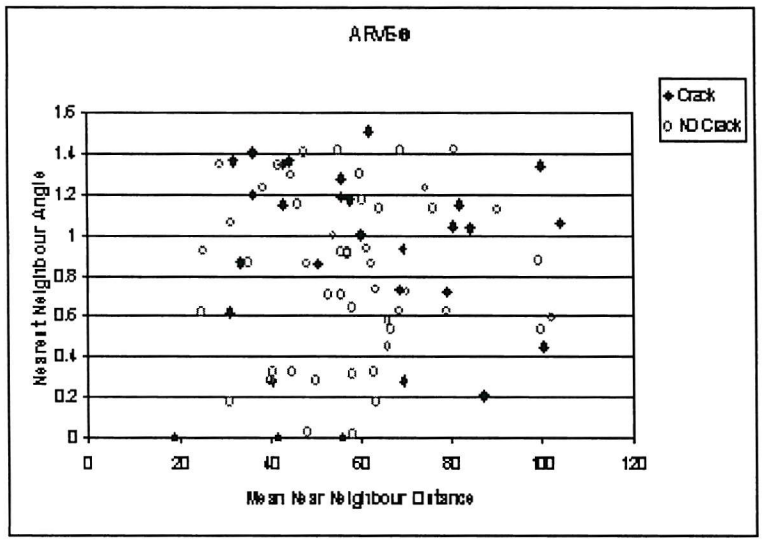


Fig 28 Bivariates plots of ARVE-θ- N.N.Ang Vs d_{Mean}

Appendix B

Al-Si-Sn

B.1 Simulated Particle Distribution and their associated tessellation cells

BRCC

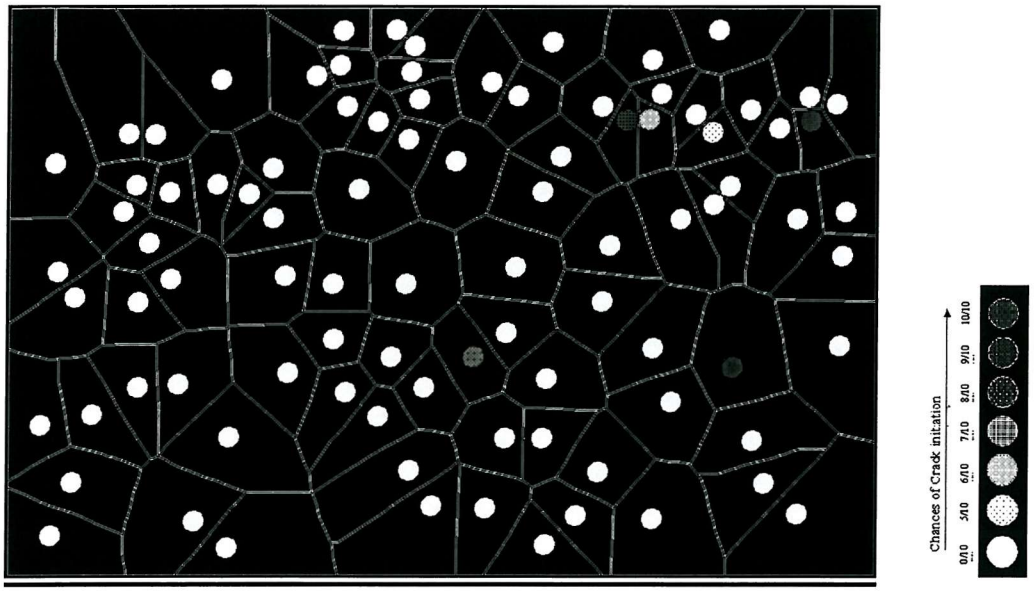


Figure 1: Al-Si-Sn, random object distribution, constant object area, circular shapes.

BCCC

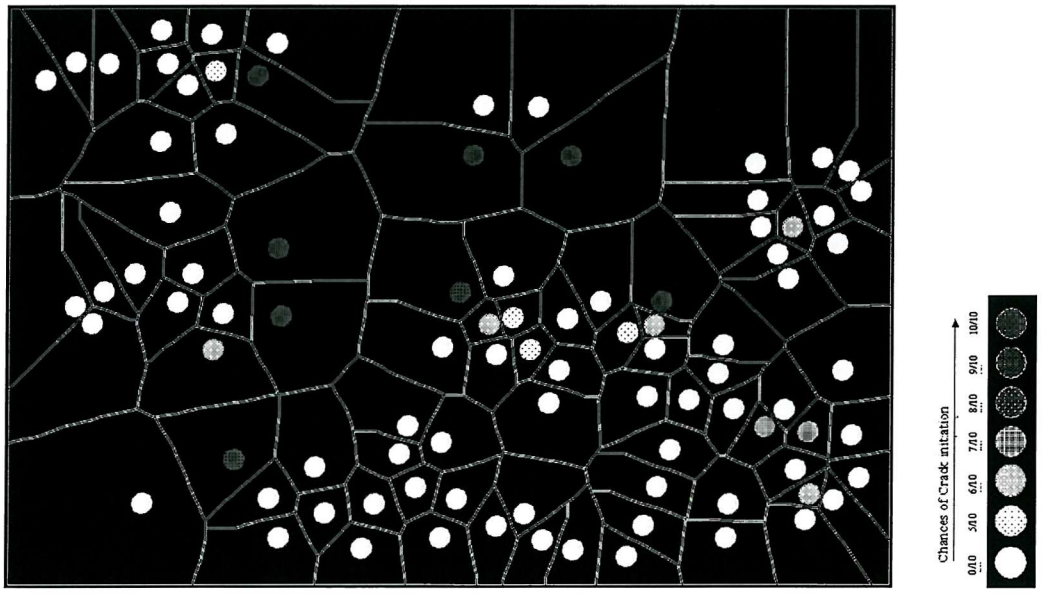


Figure 2: Al-Si-Sn, clustered object distribution, constant object area, circular shapes.

BRVC

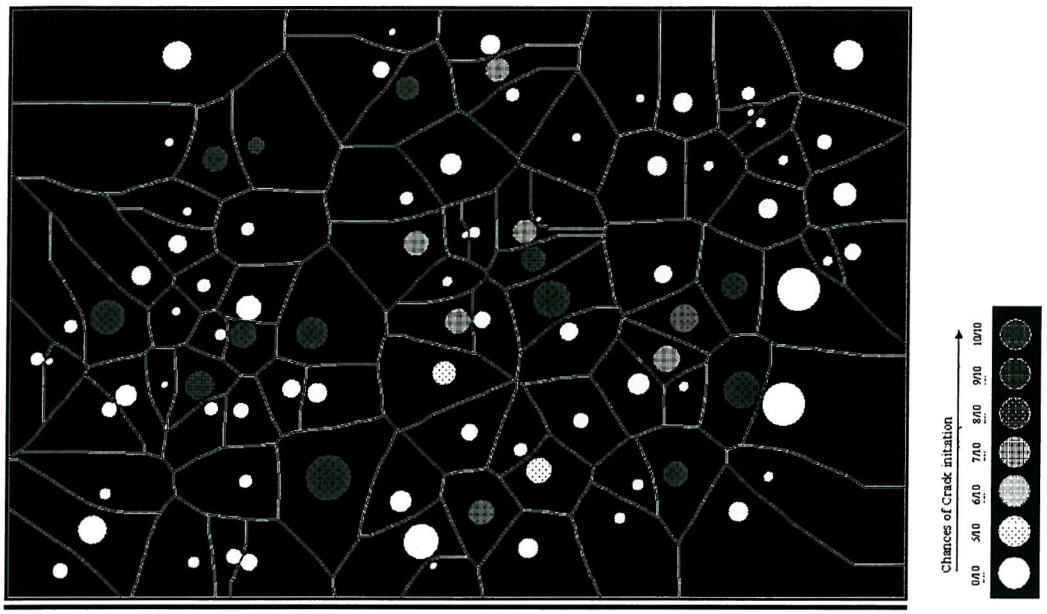


Figure 3: Al-Si-Sn, random object distribution, varying object area, circular shapes.

BCVC

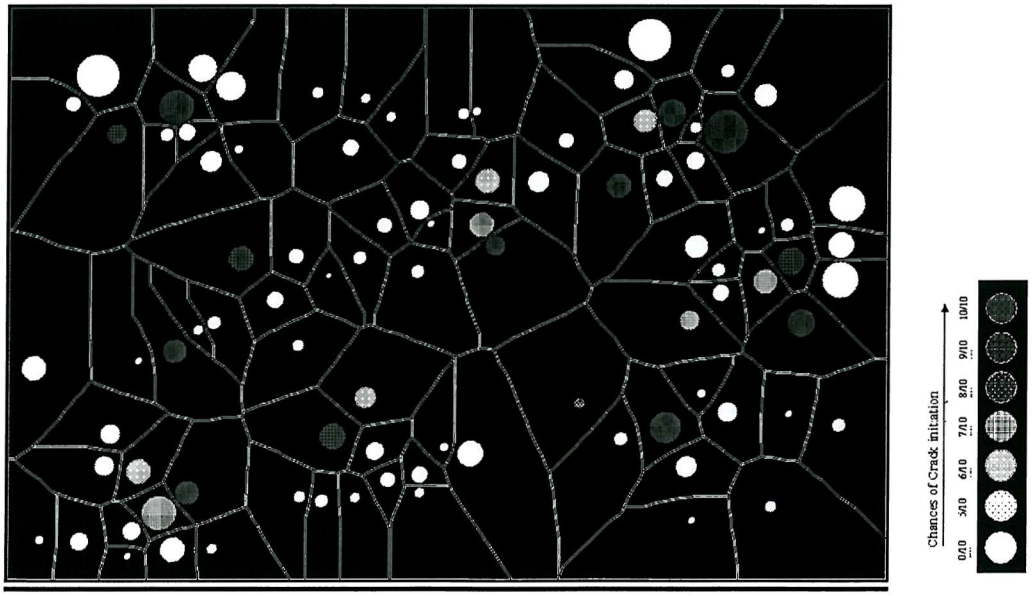


Figure 4: Al-Si-Sn, clustered object distribution, varying object area, circular shapes.

BRCE-0

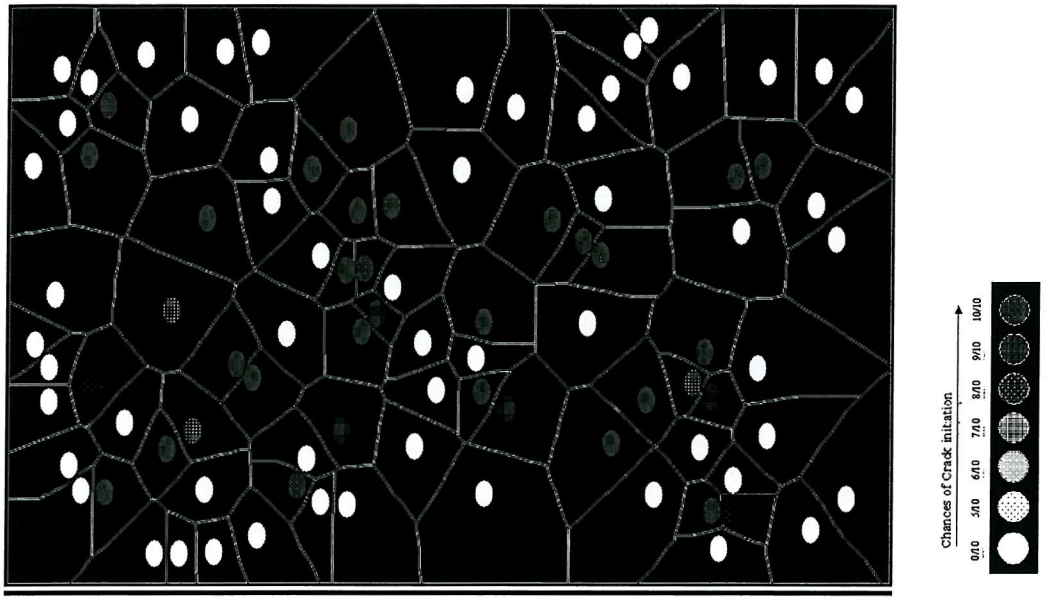


Figure 5: Al-Si-Sn, fixed object area, random object distribution, constant object area, ellipse shapes parallel to the loading axis.

BRCE-90

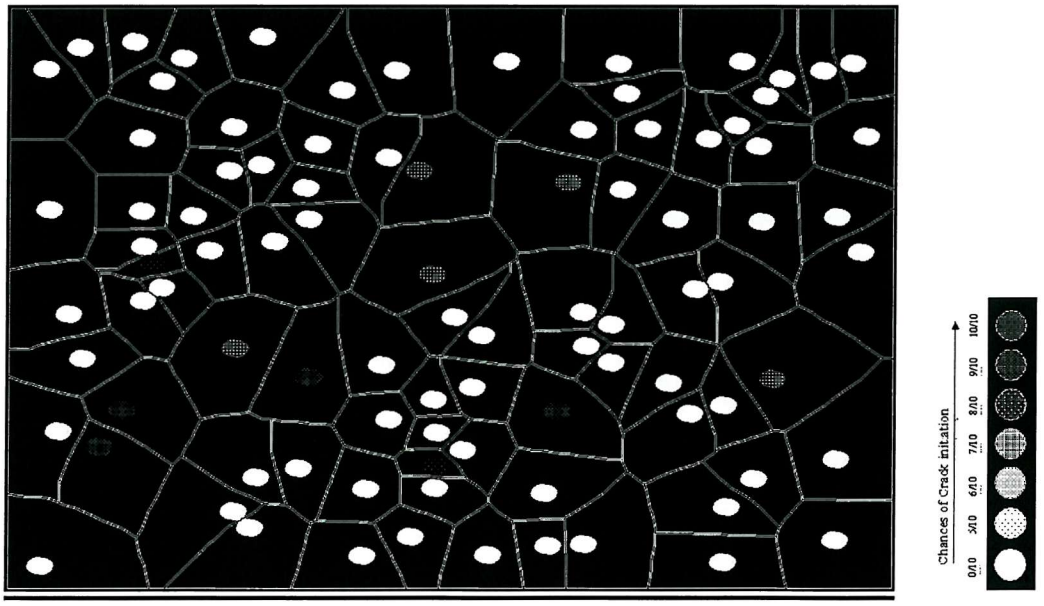


Figure 6: Al-Si-Sn, fixed object area, random object distribution, constant object area, ellipse shapes at 90° to the loading axis.

BCCE-0

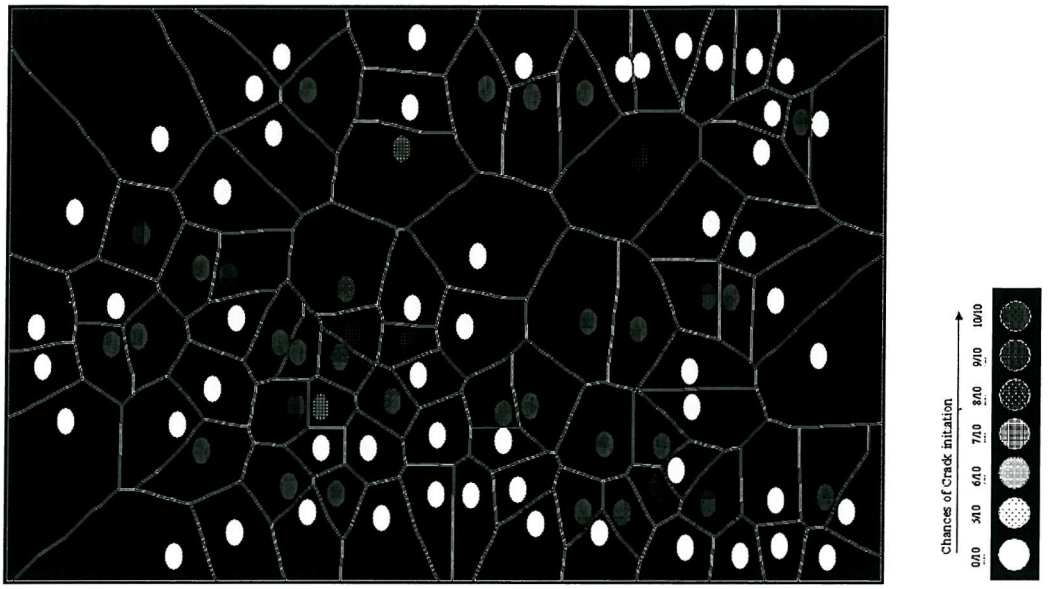


Figure 7: Al-Si-Sn, fixed object area, clustered object distribution, constant object area, ellipse shapes parallel to the loading axis.

BCCE-90

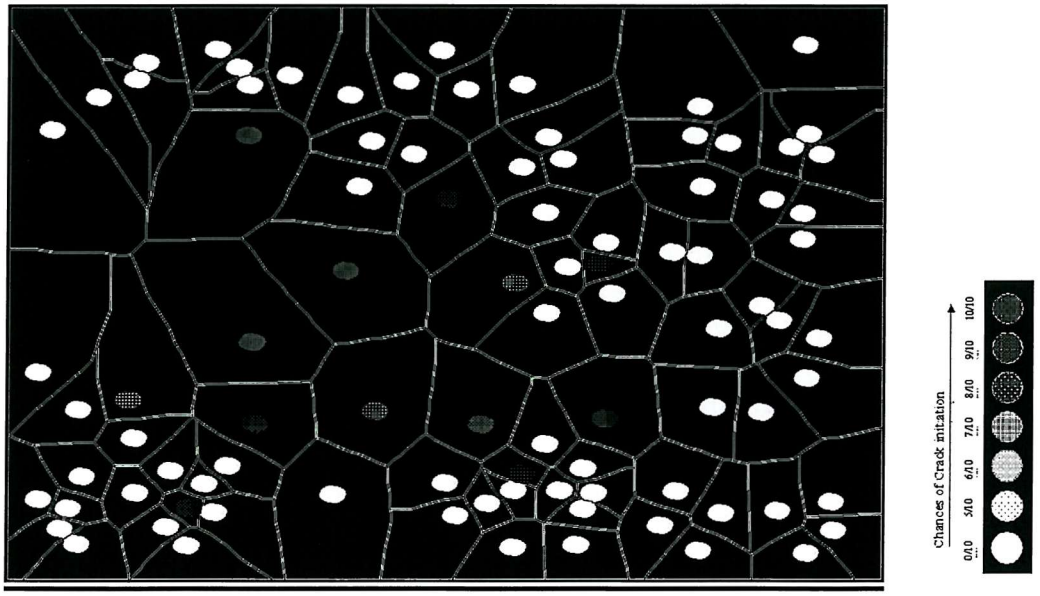


Figure 8: Al-Si-Sn, fixed object area, clustered object distribution, constant object area, ellipse shapes at 90° to the loading axis.

BRVE-θ

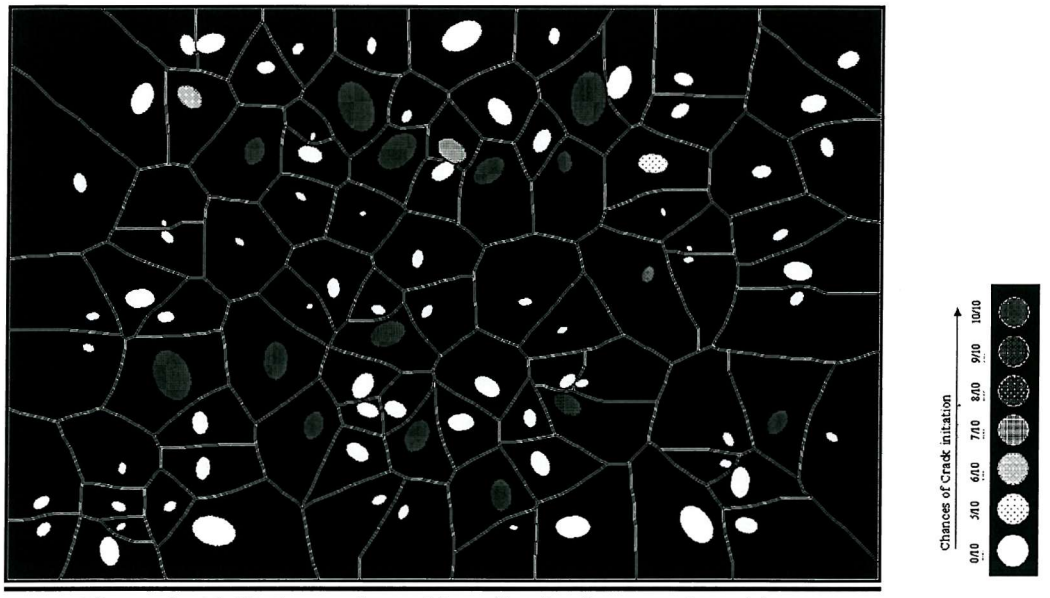


Figure 9: Al-Si-Sn, random object distribution, varying object area, ellipse shapes at angle θ to the loading axis.

BCVE-θ

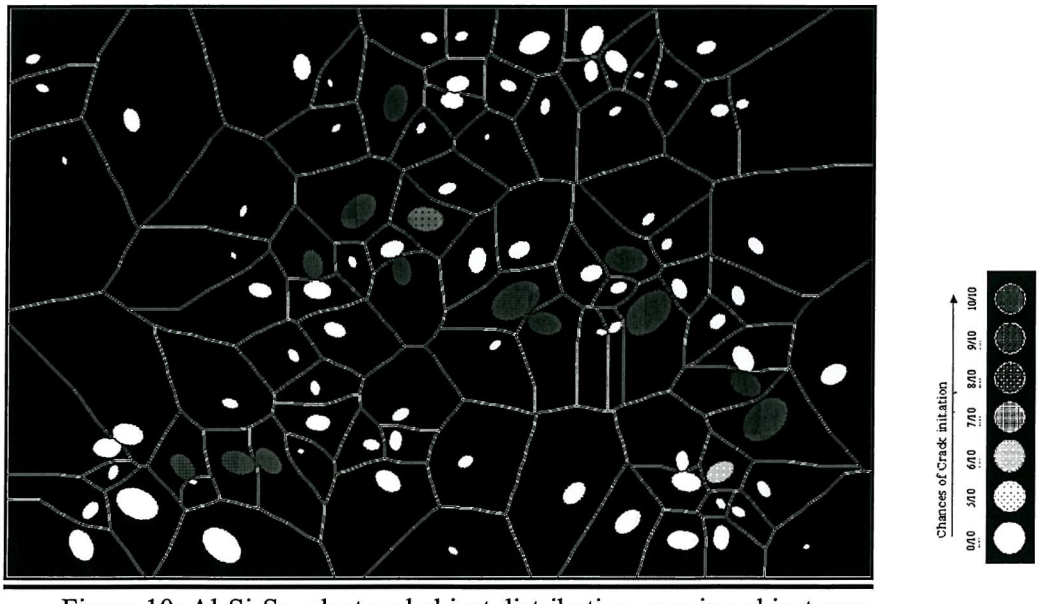


Figure 10: Al-Si-Sn, clustered object distribution, varying object area, ellipse shapes at angle θ to the loading axis.

B.2 Analysis of the Simulated Particle Distribution

B.2.1 Analysis of the Cell Area (C.A) and Local Area Fraction (L.A.F)

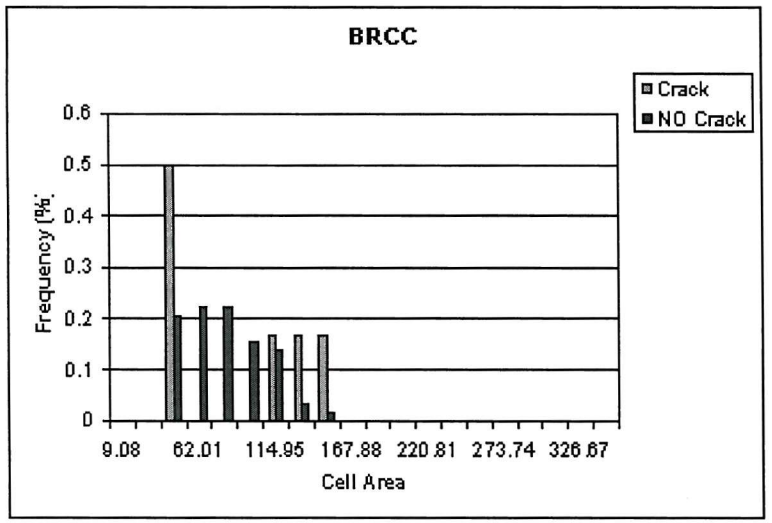


Fig 1 Histogram plots of BCCC - Cell Area

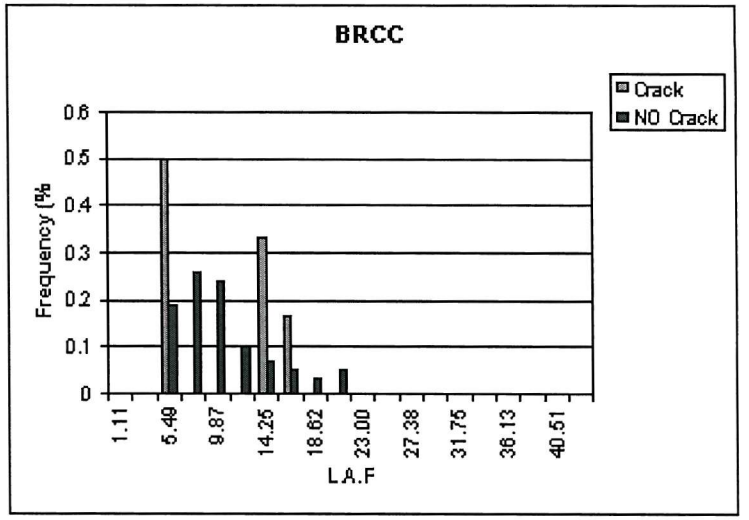


Fig 2 Histogram plots of BRCC - Local Area Fraction

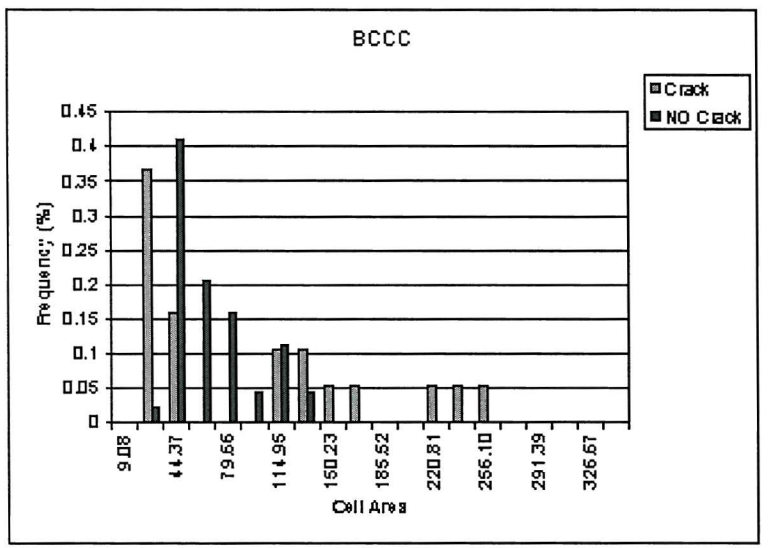


Fig 3 Histogram plots of BCCC - Cell Area

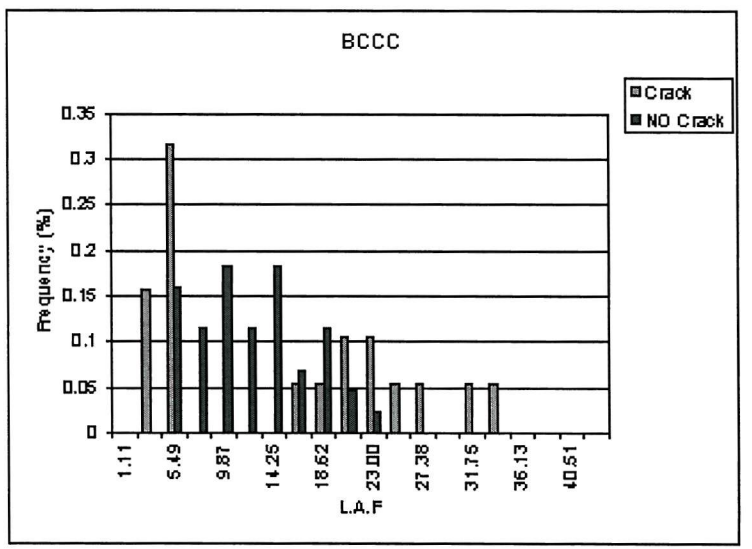


Fig 4 Histogram plots of BCCC - L.A.F

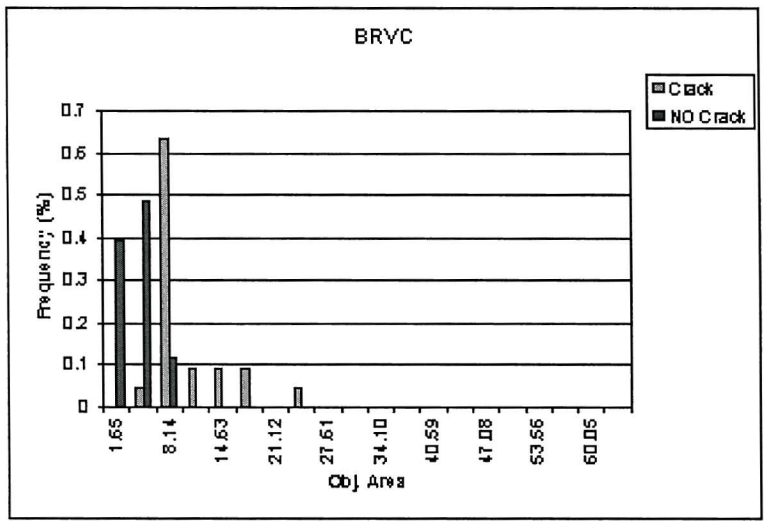


Fig 5 Histogram plots of BRVC - Object Area

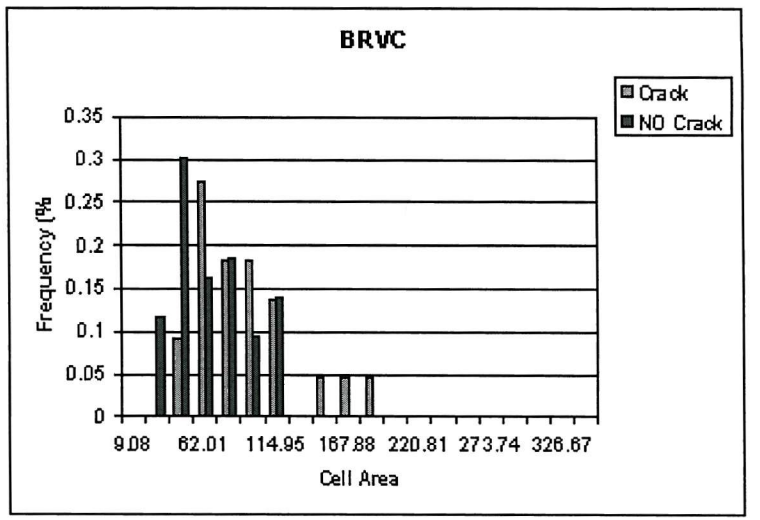


Fig 6 Histogram plots of BRVC - Cell Area

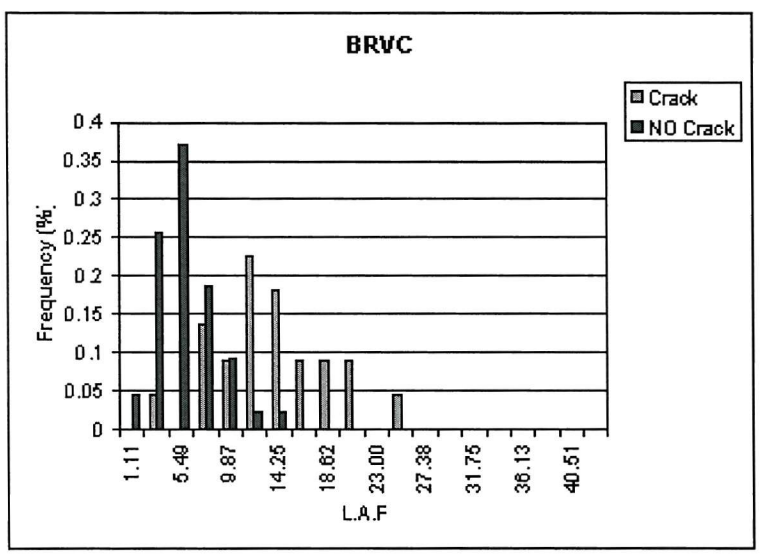


Fig 7 Histogram plots of BRVC - L.A.F

B.2.2 Analysis of the Object Angle (O.Ang) Vs Cell Angle (C.Ang)

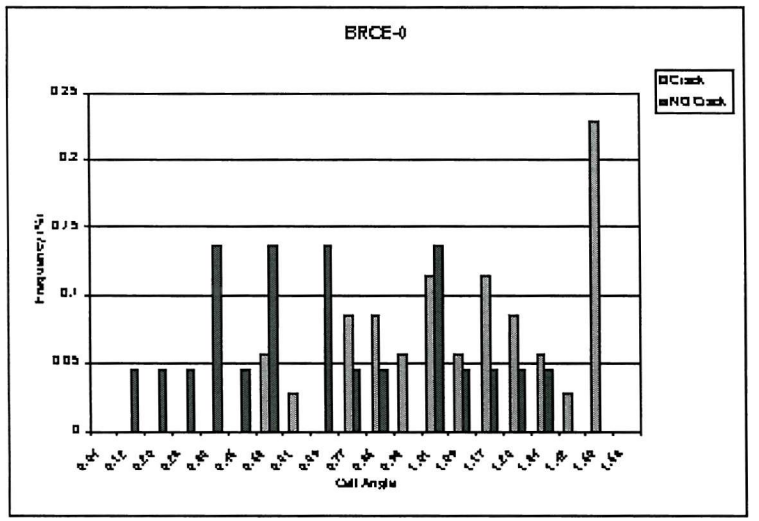


Fig 8 Histogram plots of BRCE-0 - Cell Angle

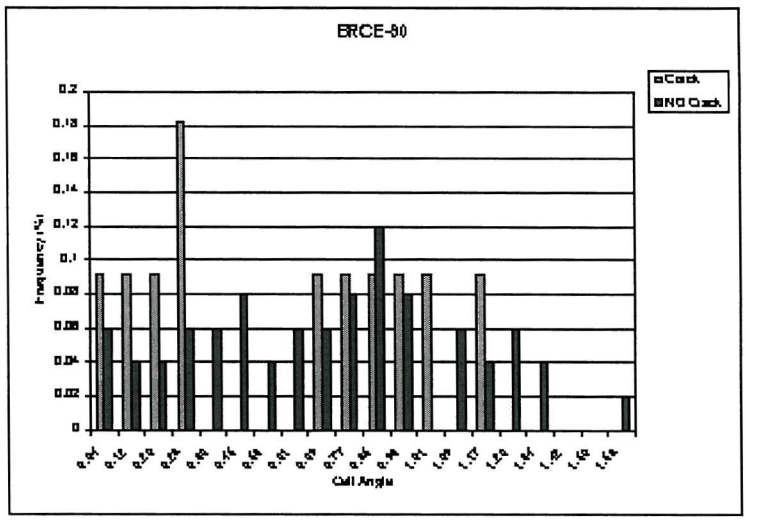


Fig 9 Histogram plots of BRCE-90 Cell Angle

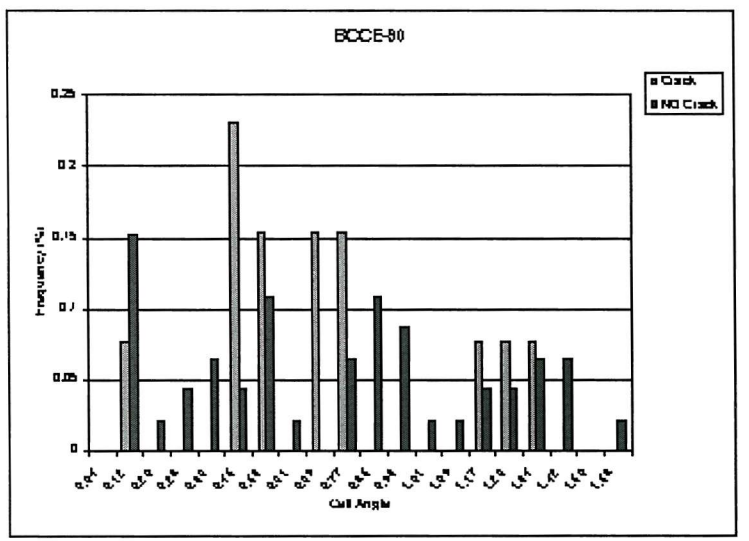


Fig 10 Histogram plots of BCCE-90 Cell Angle

B.2.3 Analysis of the Local Area Fraction (L.A.F) Vs Mean near Neighbour Distance (d_{Mean})

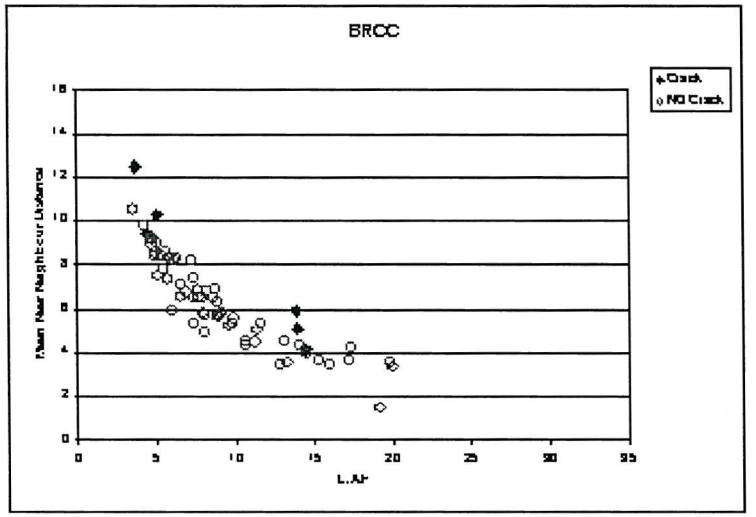


Fig 11 Bivariates plots of BRCC - d_{Mean} Vs L.A.F

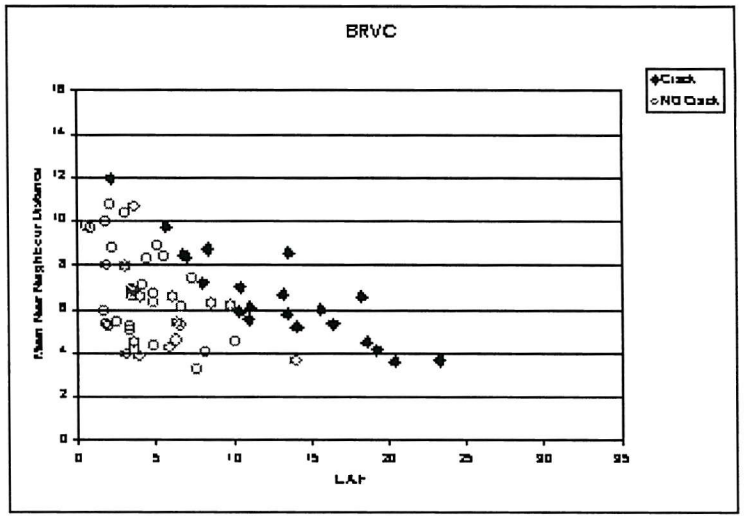


Fig 12 Bivariates plots of BRVC - d_{Mean} Vs L.A.F

B.2.4 Analysis of the Object Angle (O.Ang) Vs Nearest Neighbour Distance (d_{Min}) Vs Nearest Neighbour Angle (N.N.Ang)

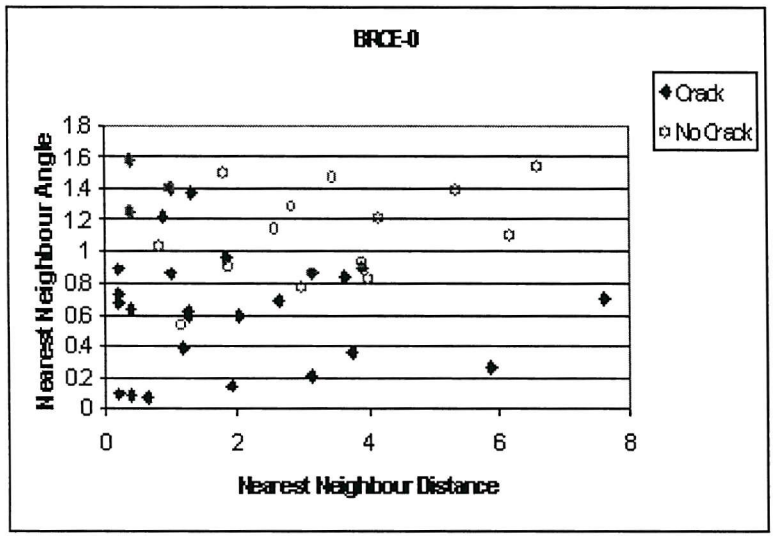


Fig 13 Bivariates plots of BRCE-0 - N.N.Ang Vs d_{min}

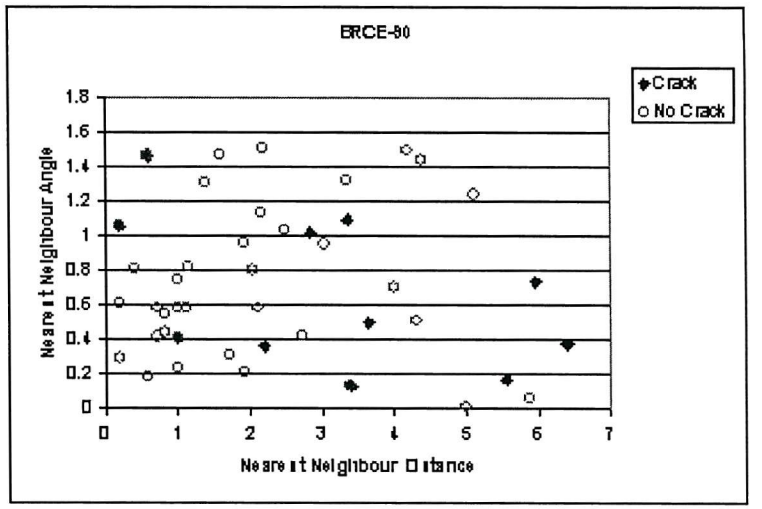


Fig 14 Bivariates plots of BRCE-90 - N.N.Ang Vs d_{min}

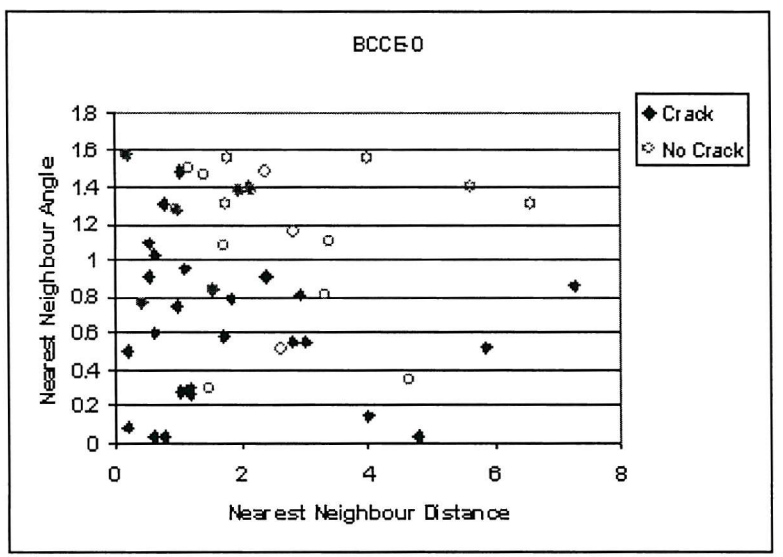


Fig 15 Bivariates plots of BCCE-0 - N.N.Ang Vs d_{min}

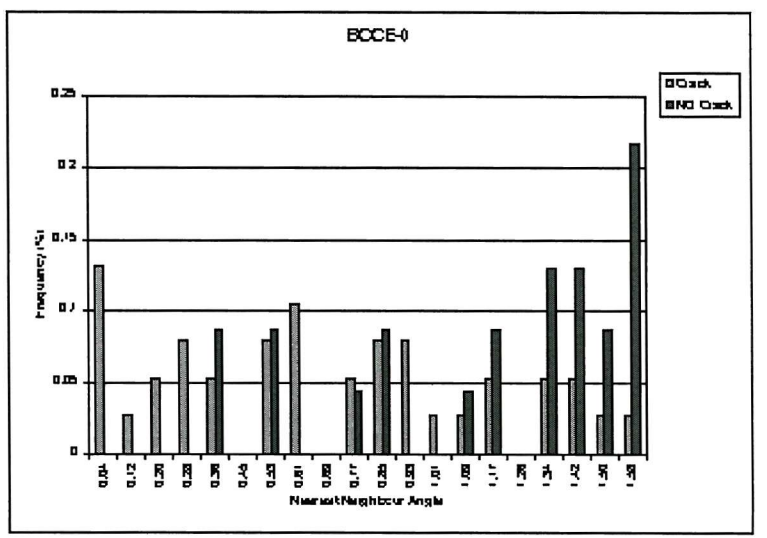


Fig 16 Histogram plots of BCCE-0 Nearest Neighbour Angle

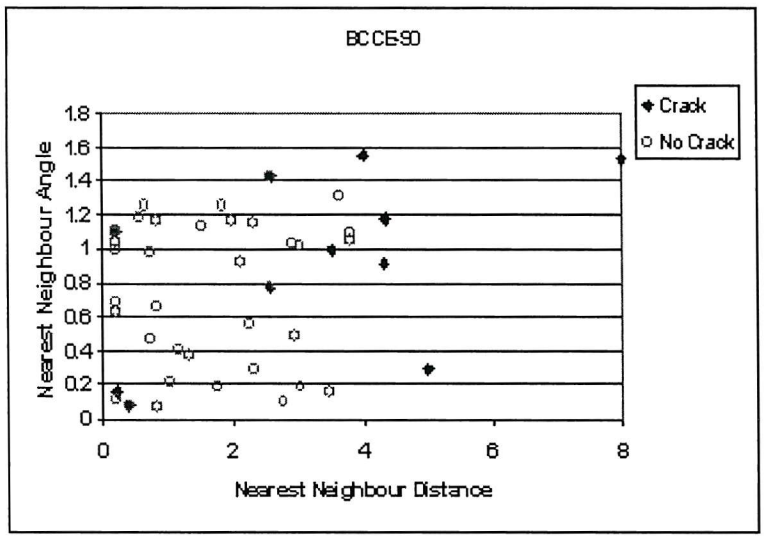


Fig 17 Bivariates plots of BCCE-90 N.N.Ang Vs d_{nn}

B.2.5 Analysis of the Overview of the Simulated Data set

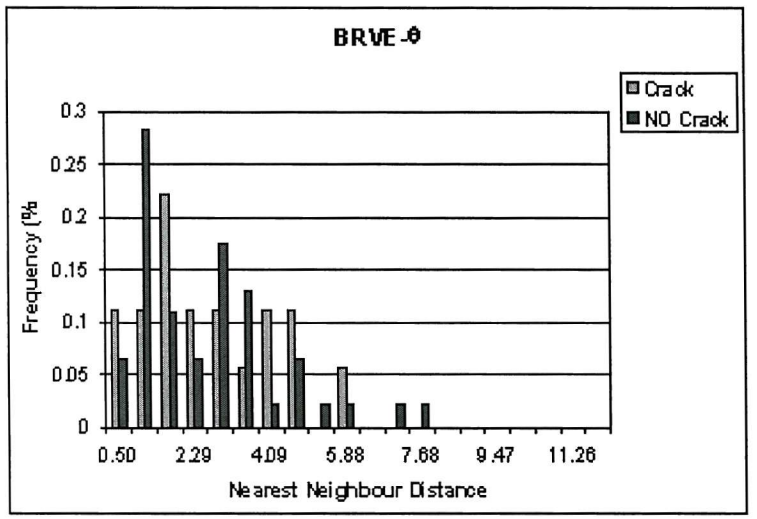


Fig 18 Histogram plots of BRVE-θ- Nearest Neighbour Distance

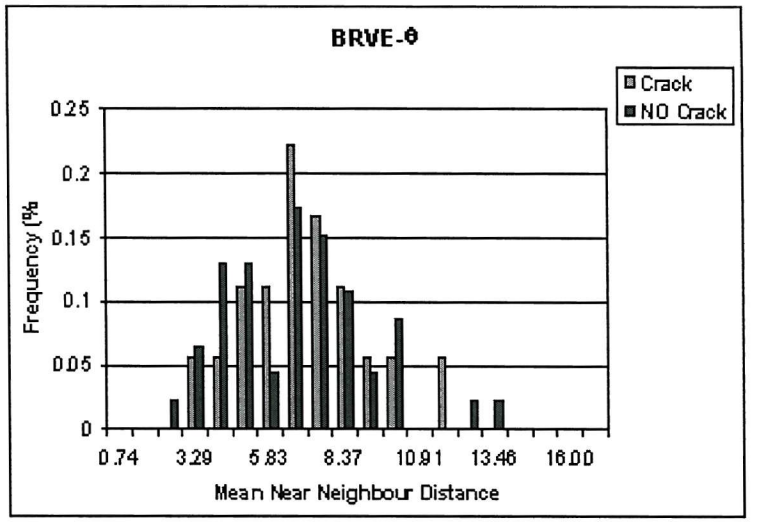


Fig 19 Histogram plots of BRVE-θ- Mean Near Neighbour Distance

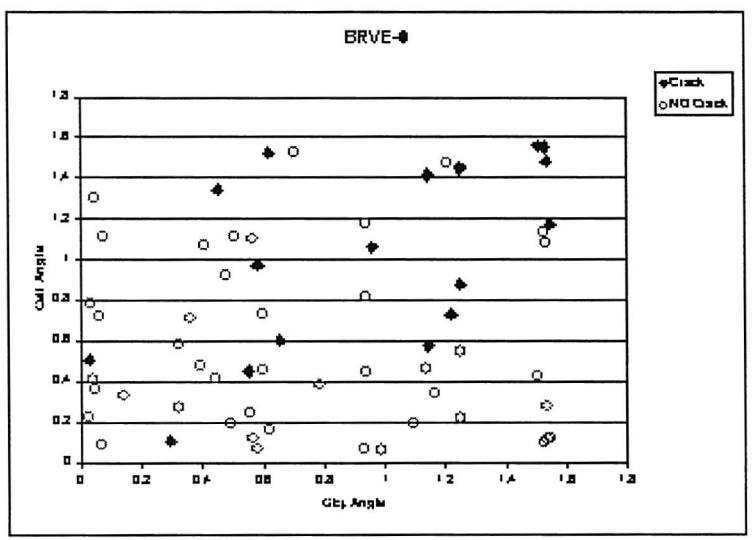


Fig 20 Bivariates plots of BRVE-θ- cell angle Vs Object Angle

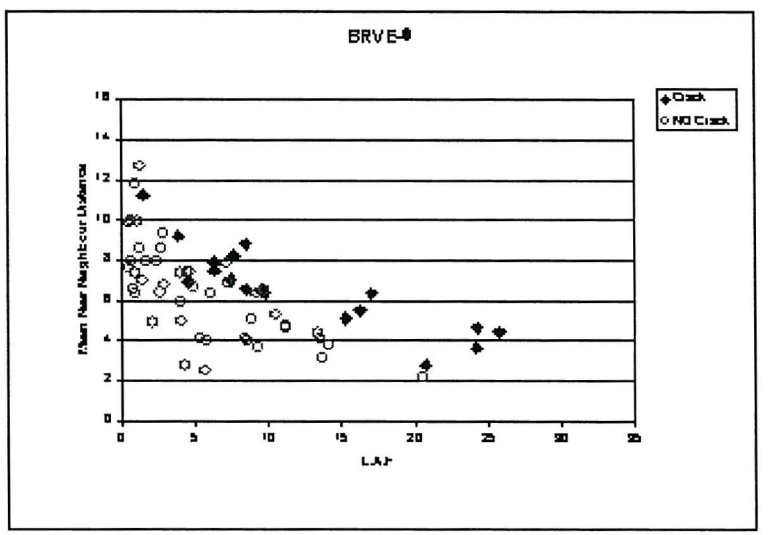


Fig 21 Bivariates plots of BRVE-θ- d_{Mean} Vs L.A.F

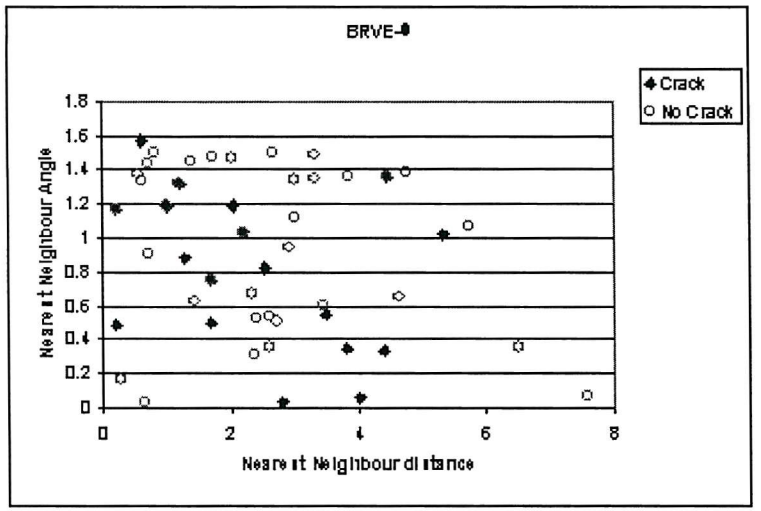


Fig 22 Bivariates plots of BRVE-θ N.N.Ang Vs dun

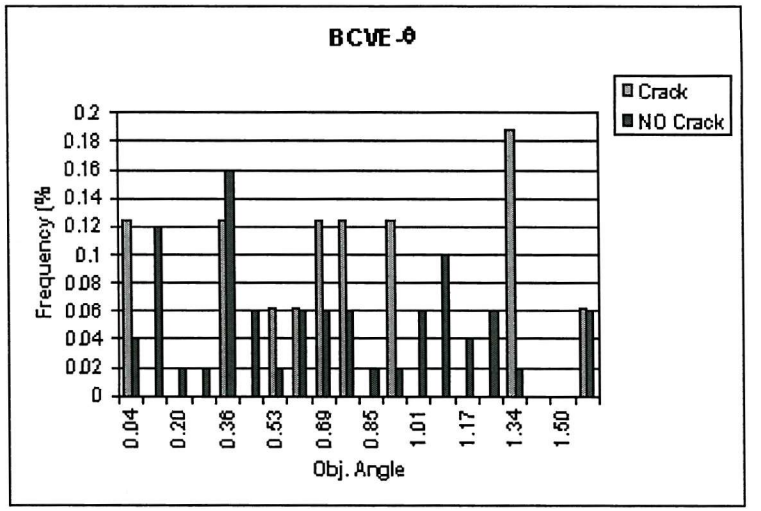


Fig 23 Histogram plots of BCVE-θ- Object Angle

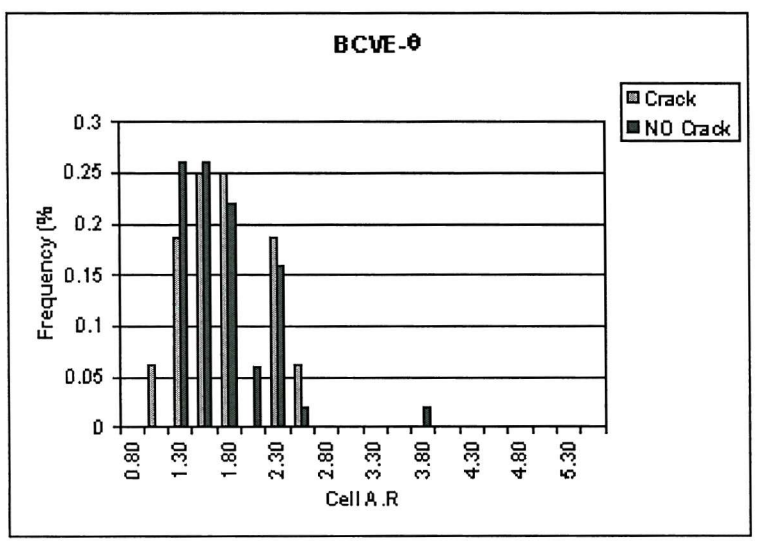


Fig 24 Histogram plots of BCVE-θ- Cell A.R

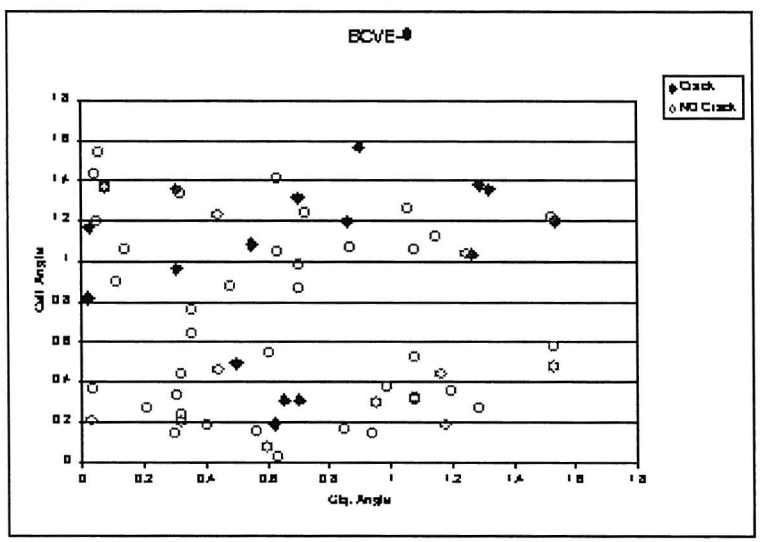


Fig 25 Bivariates plots of BCVE-θ- cell angle Vs Object Angle

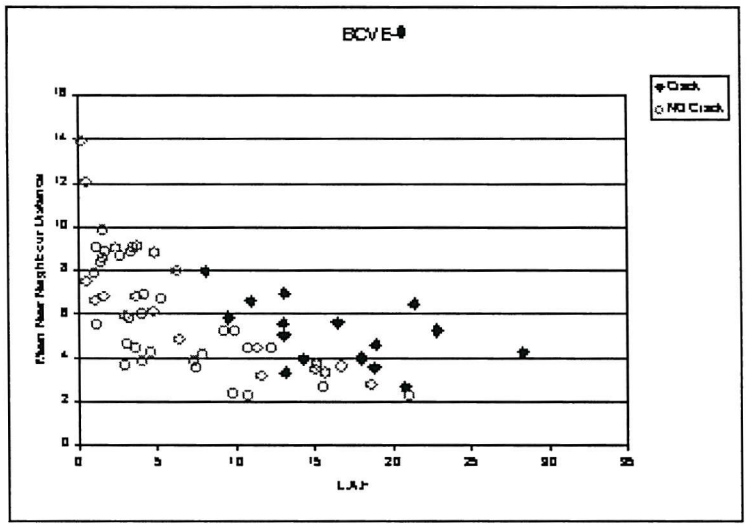


Fig 26 Bivariates plots of BCVE-θ- d_{Mean} Vs L.A.F

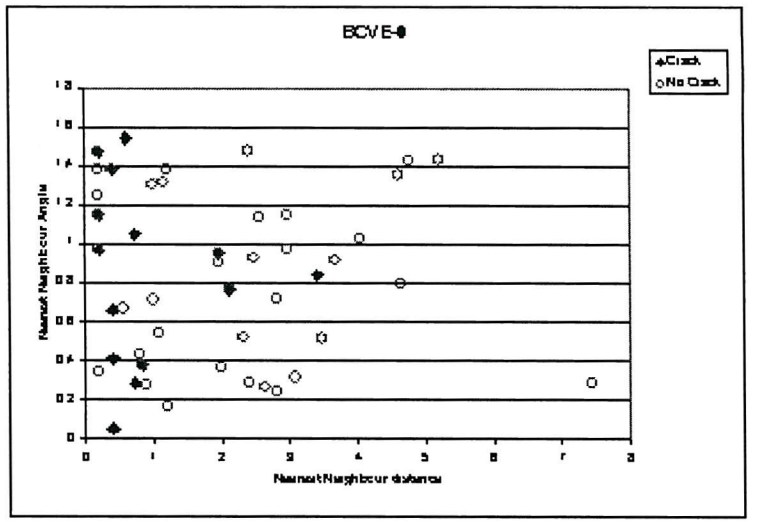


Fig 27 Bivariates plots of BCVE-θ N.N.Ang Vs d_{NN}

Bibliography

Bailer-Jones C, Bhadeshia H & Mackay D (1999). Gaussian process modeling of austenite formation in steel, *Materials Science and technology* **15(3)**, 287–292.

Bailer-Jones C, Sabin T, Mackay D & Withers P (1997). Prediction of deformed and annealed microstructures using Bayesian neural networks and Gaussian Processes, *Proceedings of the Australasia Pacific Forum on Intelligent Processing and Manufacturing of Materials* .

Bellman R (1961). *Adaptive control Processes*, Princeton University Press.

Bhadeshia H (1999). Neural Networks in Material Science, *ISIJ International* **39**, 966–979.

Bhadeshia H, Mackay D & Svensson L (1995). Impact toughness of C-Mn steel arc welds - Bayesian neural network analysis, *Materials Science and technology* **11**, 1046–1050.

Bishop C (1995). *Neural Networks for Pattern Recognition*, Oxford : Clarendon Press.

Blanz V, Schökopf B, Bülthoff B, Burges C, Vapnik V & Vetter T (1996). Comparison of View-Based Object Recognition Algorithms Using Realistic 3D Models, *in* C von der Malsburg, W von Seelen, J C Vorbrüggen & B Sendhoff, eds, *Proceedings of ICANN'96, International Conference on Artificial Neural Networks*, Springer Lecture Notes in Coputer Science, Berlin, pp. 251–256.

Borgefors G (1986). Distance Transforms in Digital Images, *Computer Vision, Graphics and Image Processing* **34**, 334–371.

Boselli J, Pitcher P, Gregson P & Sinclair I (1999). Secondary Phase Distribution Analysis via Finite Body Tessellation, *Journal of Microscopy* **TM 195**, 104–112.

Boser B, Guyon I & Vapnik V (1992). A Training Algorithm for Optimal Margin Classifiers, *in* D Haussler, ed., *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, ACM Press, Pittsburgh, PA, pp. 144–152.

Breiman L, Friedman J, Olshen R & Stone C (1984). *Classification and Regression Trees*, Wadsworth Inc.

Brown M, Grundy W, Lin D, Cristianini N, Sugnet C, Jr. M A & Haussler D (1999). Support Vector Machine Classification of Microarray Gene Expression Data, Techincal Report UCSC-CRL-99-09, University of California.

Brown M & Harris C (1994). *Neurofuzzy Adaptive Modeling and Control*, Prentice Hall Publishers.

Burges C (1998). A Tutorial on Support Vector Machines for Pattern Recognition, *Data Minind and Knowledge Discovery* **2**(2), 121–167.

Callister W J (1997). *Materials Science and Engineering - An introduction*, 4 edn, John Wiley & Sons, Inc.

Cawley G & Talbot N (2001). Manipulation of Prior Probabilities in Support Vector Classification, *International joint Conference on Neural Networks (IJCNN)* .

Chapelle O & Vapnik V (1999). Model Selection for Support Vector Machines, *In advances in Neural Information Processing Systems NIPS 12* **12**.

Chen S, Donoho D & Saunders M (1999). Atomic Decomposition by Basis Pursuit, *SIAM Journal of Scientific Computing* **41**(2), 33–61.

Cherkassky V & Mulier F (1998). *Learning from Data: Concepts, Theory and Methods*, John Wiley & Sons.

Christensen S, Reed P, Gunn S & Sinclair I (2001). A Comparison of modelling techniques in the analysis of commercial materials data, *Proceedings of IPMM 2001* .

Cortes C & Vapnik V (1995). Support Vector Networks, *Machine Learning* **20**, 273–297.

Cover T (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition, *IEEE Transactions on Electronic Computers* **14**, 326–334.

Daubechies I (1992). *Ten Lectures on Wavelets*, SIAM.

Duan, K an d Keerthi S & Poo A (2001). Evaluation of simple performance measures for tuning SVM hyperparameters, Techincal Report CD-01-11, National University of Singapore, Department of Mechanical Engineering (Control Division).

Duda R, Stork D & Hart P (2000). *Pattern Classification and Scene Analysis Part I: Pattern Classification*, John Wiley and Sons, New York.

Egan J (1975). *Signal Detection Theory and ROC Analysis*, Cognition and Perception, Academic Press.

Evgeniou T, Pontil M & Poggio T (1999). A Unified Framework for Regularisation Networks and Support Vector Machines, Technical Report 1654, MIT.

Friedman J (1997). On Bias, variance, 0/1 - Loss and curse of dimensionality, *Data Mining and Knowledge Discovery* **1**, 55–77.

Fuji H, Mackay D & Bhadeshia H (1996). Bayesian neural network analysis of fatigue crack growth rate in Nickel base Supperalloys, *ISIJ* **11**, 1371–1382.

Fukunaga K (1990). *Introduction to Statistical Pattern Recognition*, second edn, Academic Press.

Gavard L, Bhadeshia H, Mackay D & Suzuki S (1996). Bayesian neural network model for austenite formation in steels, *Materials Science and technology* **12**, 453–462.

Gunn S (1998). Support Vector Machine for Classification and Regression, Technical report, University of Southampton.

Gunn S (1999). SUPANOVA - A Sparse, Transparent modelling Approach, *In Proc. IEEE Int. Workshop on Neural Networks for Signal Processing* .

Hastie T & Tibshirani R (1990). *Generalized Additive Models*, Chapman and Hall.

Herbrich R (2001), Learning Linear Classifiers - Theory and Algorithm, Thesis, Technical University of Berlin.

Herbrich R, Graepel T & Campbell C (2001). Bayes Point Machines, *Journal of Machine Learning Research* **1**, 245–279.

Hockley R, Thakar D, Boselli J, Sinclair I & Reed P (1999). Effect of Graphite Nodule Distribution on 'Crack' Initiation and Early Growth in Austempered Ductile Iron, *in* R K et al., ed., *Small Fatigue Cracks Mechanics, Mechanisms and Applications*, Elsevier Science, Hawaii, USA, pp. 49–56.

Joachims T (2000). Estimating the Generalisation Performance of SVM Efficiently, *in In Proceedings of the International Conference of Machine Learning*, Morgan Kaufman, San Francisco, pp. 431–438.

Joyce M (2001), Fatigue of Aluminium Linings in Plain automotive Bearings, PhD thesis, University of Southampton.

Kandola J, Gunn S, Sinclair I & Reed P (1999). A Data Driven knowledge extraction of materials properties, *Proceedings of IPMM 1999* .

Kohonen T (1990). The Self-Organising Map, *Proceedings of IEEE* **78**, 1464–1480.

Kubat M, Holte R & Matwin S (1998). Learning When Negative Examples Abound, *Machine Learning* **30**, 195–215.

Kwok J (1999). Moderating the Outputs of Support Vector Machine Classifiers, *IEEE Trans. Neural Networks* **10**(5), 1018–1031.

Lee K, Harris C, Gunn S & Reed P (2001*a*). Approaches to Imbalanced Data for Classification : A Case Study, *International ICSC Congress on Computational Intelligence Methods and Applications (CIMA) - Advances in Intelligent Data Analysis* .

Lee K, Harris C, Gunn S & Reed P (2001*b*). A case study of SVM extension techniques on classification of imbalanced data, *Proceedings in 2001 WSES International Conference on: Neural Networks and Applications (NNA '01)* .

Lee K, Harris C, Gunn S & Reed P (2001*c*). Classification of Imbalanced Data with Transparent Kernels, *International joint Conference on Neural Networks (IJCNN)* .

Lee K, Harris C, Gunn S & Reed P (2001*d*). Control Sensitivity SVM for Imabalanced data - A Case Study on Automotive Material, *Proceedings in 5th International conference on Artifical Neural Networks and Genetic Algorithms (ICANNGA 01)* .

Lee K, Harris C, Gunn S & Reed P (2001*e*). Regression Models for Classification to Enhance Interpretability, *Intelligent Processing and Manufacturing of Materials* .

Lin Y (1999). Support Vector Machines And The Bayes Rule in Classification, Technical Report No. 1014, University of Wisconsin, Department of Statistics.

Lin Y, Lee Y & Wahba G (2000). Support Vector Machines for Classification in Nonstandard Situations, Technical Report 1016, University of Wisconsin.

Linkens D & Yang Y (2001). Black Box modelling and Its application in Stell Property Prediction - A Brief Review, *Proceedings of IPMM 2001* .

MacKay D (1991), Bayesian Modelling and Neural Networks, PhD thesis, California Institute of Technology.

Mallat S & Zhang Z (1993). Matching Pursuit in a Time Frequency Dictionary, *IEEE Transactions on Signal Processing* **41**, 3397–3415.

Mercer J (1909). Functions of postive and negative type, and their connection with theory of integral equations, *Transactions of london Philosophical Scoiety (A)* **209**, 415–446.

Mray P, Richmond O & H.L M (1983). Use of the Dirichlet tessellation for characterising and modelling nonregular dispersions of secondary-phase particles, *Metallography* **16**, 39–58.

Murphy K (2001). An Introduction to Graphical Models, Techincal report, Intel Research.

Neal R (1996). *Bayesian Learning for Neural Networks*, No. 118 in Lecture Notes in Statistics, Springer, New York.

Osuna E, Freund R & Girosi F (1996). Support Vector Machines: Training and Applications, Technical Report AIM-1602, MIT A.I. Lab.

Pearl J (1998). *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann Publishers.

Plate T (1999). Accuracy Versus Interpretability in Flexible Modelling: Implementing a Tradeoff Using Gaussian Process Models, *Behaviourmetrika Special Issue on Interpreting Neural Network Models* (26), 29–50.

Platt J (2000). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, *in* A Smola, P Bartlett, B Schlkopf & D Schuurmans, eds, *Advances in Large Margin Classifiers*, MIT Press, pp. 61–73.

Platt J, Cristianini N & Shawe-Taylor J (2000). Large Margin DAGs for Multiclass Classification, *in Advances in Neural Inforamtion Processing Systems NIPS 12*, MIT Press, San Mateo, CA, pp. 547–553.

Provost F (2000). Learning with Imbalanced Data Sets 101, *Invited paper for the AAAI'2000 Workshop on Imbalanced Data Sets* .

Provost F & Fawcett T (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions, *In Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining, AAAI Press* pp. 43–48.

Rasmussen C (1996), Evaluation of Gaussian Processes and Other Methods for Non-linear Regression, PhD thesis, Univeristy of Toronto.

Rencher A (1998). *Multivariate Statistical Inference and Application*, Wiley Series.

Schooling J, Brown M & Reed P (1999). An Example of the use of Neural Computing Techniques in Materials Science - The Modeling of fatique Thresholds in Ni-Base Supperalloys, *Material Science and Engineering* **A260**, 222–239.

Schwarz H & Exner H (1983). The characterisation of the arrangement of feature centroids in planes and volumes, *Journal of Miscroscopy* **129**, 155–169.

Seeger M (1999). Bayesian Methods for Support Vector Machines and Gaussian Processes, Technical report, University of Edinburgh.

Smola A (1998), Learning with Kernels, PhD thesis, Technical University of Berlin.

Sollich P (2000). Probabilistic Methods for Support Vector Machines, *in* S Solla, T Leen & K R Mller, eds, *Advances in Neural Information Processing Systems NIPS 12*, MIT Press, pp. 349–355.

Spitzig W, Kelly J & Richmond O (1985). Quantitative Characterisation of Secondary Phase Populations, *Metallograhy* **18**, 235–261.

Stitson M & Weston J (1996). Implementational Issues of Support Vector Machines, Technical Report CSD-TR 96-18, Computational Intelligence Group, Royal Hollway, University of London.

Sumpter B & Noid D (1996). On the Design,analysis, and characterisation of materials using computational neural network, *Annual Review Material Science* **26**, 223–277.

Suresh S (1998). *Fatigue of Materials*, 2 edn, Cambridge University Press.

Taylor S (1998). Classification Accuracy Based on Observed Margin, *Algorithmica* **22**, 157–172.

Tipping M (2000). The Relevance Vector Machine, *in* S Solla, T Leen & K R Mller, eds, *Advances in Neural Information Processing Systems NIPS 12*, MIT Press, pp. 652–658.

Vander Voort G (1990). Evaluating clustering of secondary-phase particles, *Micron. Advances in Video Technology for Microstructural Control.* (90), 242–267.

Vapnik V (1995). *The Nature of Statistical Learning Theory*, Springer Verlag, New York.

Veropoulos K, Campbell C & Cristianini N (1999). Controlling the Sensitivity of Support Machines, *Proceedings of the Int. Joint Conf. on Artifical Intelligence (IJCAI99), Sweden* pp. 55–60.

Vincent L & Soille P (1991). Watersheds in Digital spaces : An Efficient Algorithm Based on Immersion Simulations, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(6), 583–598.

Wahba G (1990). *Spline Models for Observational Data*, Vol. 59 of *Series in applied Mathematics*, SIAM, Philadelphia.

Wahba G, Lin Y. & Zhang H (1999). GACV for Support Vector Machines, or Another way to Look at Margin-Like Quantities, Technical Report 1006, University of Wisconsin.

Weston J (1999). Leave One Out Support Vector Machines, *Proccedings of the International Joint Conference on Artifical Intelligence (IJCAI99), Sweden* pp. 727–733.

Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T & Vapnik V (2000). Feature Selection in SVMs, *in Advances in Neural Information Processings Systems NIPS 13*, MIT Press, pp. 668–674.

Weston T & Watkins C (1998). Multi-class Pattern Support Vector Machines, Technical Report CSD-TR-98-04, Royal Holloway.

Yang N, Boselli J, Gregson P J & Sinclair I (2000). Simulation and quantitative assessment of finite-size particle distributions in metal matrix composites, *Material Science and Technology* **16**, 797–805.

Yang N, Boselli J & Sinclair I (2001). Simulation and quantitative assessment of homogeneous and inhomogeneous particle distributions in particulate metal matrix composites, *Journal of Microscopy* **201-2**, 189–200.