

Defeaturing CAD models using Machine Learning

Submitted in partial fulfillment of the requirements of Dual Degree
(Bachelor and Master of Technology) in Mechanical Engineering

by

Krutika Patil

Roll No. 12D100005

in the supervision of

Prof. S. S. Pande



Department of Mechanical Engineering
Indian Institute of Technology Bombay
India

July 13, 2017

Defeaturing CAD models using Machine Learning

Krutika Patil

Abstract

CAD software is extensively used in applications from modeling parts, CAE analysis to product life cycle management. Complex CAD models have a large number of geometric features and a complex topology with numerous faces, edges, vertices. A representation of the model with such a high level of detail requires a large amount of computer memory.

Such a detailed description is not often necessary for other downstream engineering applications. When the complexity of the CAD model increases, the FEA meshing procedure takes significant time, algorithms generate poor quality meshes near small features like chamfers, holes, fillets etc. Poor mesh quality can lead to erroneous FEA results. Thus CAD models are simplified by suppressing detailed geometric features that do not affect the finite element analysis results and significantly decrease the mesh size and improve mesh quality. A lot of this simplification is done manually. A need thus exists to evolve procedures for automatic defeaturing of CAD models for such tasks.

Most of the existing literature to automate the defeaturing process focuses on using purely geometric measures like the dimensions of the geometric feature with respect to the dimensions of the part, the distance of the geometric feature from the displacement constraints and point of application of load to decide whether a geometric feature is suppressible.

The objective of the present project was to use machine learning techniques and build a prototype system to identify suppressible holes / pockets in brackets subjected to FEA simulations for static structural analysis. Unlike all the other approaches, we also take into account the effect of direction of loading in determining suppressible features.

A comprehensive dataset was generated through extensive FEA simulations on a chosen part family viz. brackets for the purpose of this study. Geometric features selected for this study include blind and through round holes and square pockets. The dataset consisted of variations in the sizes of these features, their location on the bracket and the direction of loading. The data from the FEA simulations was transformed to a form that can be processed by a classifier. Geometric features were labeled as suppressible or not suppressible based on the difference in the value of maximum von Mises stress in the bracket in the presence and absence of the feature. The dataset thus generated was further processed by a neural network. The neural network was trained on the training data and employed to predict the suppressibility of geometric features on the testing data.

The accuracy of the network in making these predictions was about 90 %. Through case studies it was demonstrated that defeaturing CAD models as suggested by the proposed system, achieves a reduction in the number of mesh elements without significant changes in the stress analysis results.

Approval Sheet

This dissertation entitled "**Defeaturing CAD models using Machine Learning**" by Krutika Patil (12D100005) is approved for the degree of Master of Technology in Mechanical Engineering with specialization in Computer Aided Design and Automation.



Internal Examiner

Prof. Parag U. Tandaiya

Dept. of Mechanical Engg, IIT Bombay



Internal Examiner

Prof. Salil Kulkarni

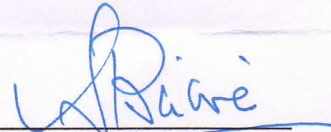
Dept. of Mechanical Engg., IIT Bombay



Guide

Prof. S. S. Pande

Dept. of Mechanical Engg., IIT Bombay



Chairman

Prof. Supratim Biswas

Dept. of Computer Science, IIT Bombay

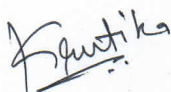
Date: July 10, 2017

Place: IIT Bombay

Declaration

I declare that this written submission is my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified my idea/data/fact/source in my submission. I understood that any violation of the above will cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has been taken when needed.

Signature:



Name of the student: Krutika Patil

Roll No.: 12D100005

Date: 5th July, 2017

Acknowledgments

I would first like to thank my thesis advisor Professor S. S. Pande of the Department of Mechanical Engineering at IIT Bombay. His guidance and motivation were critical in planning and completing my thesis.

I would also like to thank Professor Salil Kulkarni of the Department of Mechanical Engineering at IIT Bombay for the insightful discussions.

I am thankful to VIP Lab in the Electrical Engineering Department for allowing me access to a state of the art computer for running FEA simulations.

I am grateful to my parents and my friends at IIT Bombay for their unfailing support during the course of my study.

Contents

1	Introduction	1
1.1	Motivation for Defeaturing	1
1.2	Scope of the Project	2
1.3	Organization of the report	2
2	Literature Survey	4
2.1	Scope	4
2.2	Simplification Approaches	4
2.2.1	Based on explicit features	4
2.2.2	Based on surface entities	8
2.3	Summary and Conclusions	13
2.4	Objective of Project	13
3	Methodology	14
3.1	Overview	14
3.2	Generating the Dataset	15
3.2.1	CAD model creation	15
3.2.2	Finite Element Simulations	17
3.2.3	Simulation Runtime Details:	24
3.3	Pre-Processing Data	25
3.3.1	Inputs to the classifier	25
3.3.2	Identification of Suppressible Holes	29
3.4	Classification Model	31
3.5	Accuracy of the Classifier	37
3.5.1	Through Holes	37
3.5.2	Blind Holes	38
3.5.3	Square Holes	39
3.5.4	Blind and Through Holes Combined	40
3.5.5	Generalization	42

4	Case Studies	44
4.1	Overview	44
4.2	Case Study 1	44
4.3	Case Study 2	46
4.4	Case Study 3	48
5	Summary and Future Directions	51
5.1	Overview	51
5.2	Conclusions	52
5.3	Future Work	53

List of Figures

2.1	a) Edges constituting the sheet b) Detected free edges in black — Joshi and Dutta [5]	5
2.2	a) Original model with holes b) Defeatured model — Joshi and Dutta [5]	6
2.3	Fillet with constant parameter lines (Joshi and Dutta [5])	6
2.4	Equations to calculate curvature (Joshi and Dutta [5])	7
2.5	Point of intersection of supporting edges (Joshi and Dutta [5])	7
2.6	a) Adjacency graph b) merging of clusters c) removing duplicate edges (Sheffer [10])	8
2.7	a)Original model b) A face is split symmetrically into 3 parts c) new parts are merged with the adjacent faces (Sheffer [10])	8
2.8	Computing the boundary angle index Sheffer [10]	9
2.9	a) Original model 127 faces b) Simplified model 46 faces (Sheffer [10])	10
2.10	Color coded size field; in the red region the computed mesh size is less than the threshold	11
2.11	Edge collapse operator Quadros and Owen [8]	11
2.12	original model	12
2.13	defeatured model (Quadros and Owen [8])	12
2.14	Left - Tet mesh on defeatured model — Right - Tet mesh on original model (Quadros and Owen [8])	12
3.1	Summary of the approach to defeature CAD models.	14
3.2	Dimensions of the bracket in millimeter	15
3.3	Illustration of a bracket with a pattern of four holes / pockets	16
3.4	The surfaces highlighted are constrained to be fixed	17
3.5	Majority of the elements in the mesh are hexahedral elements with 20 nodes.	17
3.6	Region of interest in the mesh	18
3.7	Illustration of variation of diameter in a pattern of four through holes	19

3.8	Illustration of variation of location of holes in a pattern of four blind holes	20
3.9	The location of holes in the pattern was varied by changing the distance d	21
3.10	Illustration of variation of loading direction in a pattern of 4 blind holes	22
3.11	Polar plot: Number of training examples for each direction of load θ	23
3.12	Model of the original and defeatured bracket	23
3.13	Calculating Volume and Surface area gain	25
3.14	Calculating Normalized distance from boundary conditions	26
3.15	Holes with same descriptors	27
3.16	Division of XY plane into 8 regions	28
3.17	Stress with and without a hole feature.	29
3.18	Architecture of the neural network	31
3.19	Output of a neuron	32
3.20	Classification accuracy versus threshold p	35
3.22	Confusion Matrix	37
3.23	ROC curves: Through holes	38
3.24	Confusion matrix	38
3.25	ROC curve: Blind holes	39
3.26	Confusion matrix	39
3.27	ROC Curve: Square holes	40
3.28	Confusion matrices	41
3.29	ROC Curve: Blind and Through holes data	41
3.30	Direction of load in training and testing data	42
3.31	Classification performance on through holes	42
3.32	Classification performance on blind holes	43
4.1	Case Study 1: Geometry and boundary conditions	44
4.2	Case Study 1: Stress distribution in the original model	45
4.3	Case Study 1: Stress distribution in the defeatured model	45
4.4	Case Study 2: Geometry and boundary conditions	46
4.5	Case Study 2: Features marked as suppressible	46
4.6	Case Study 2: Stress distribution in the original model	47
4.7	Case Study 2: Stress distribution in the defeatured model	47
4.8	Case Study 3: Geometry and boundary conditions	48
4.9	Case Study 3: Stress distribution in the original model	48
4.10	Case Study 3: Stress distribution in the defeatured model (Classifier does not include loading direction information)	49

4.11 Case Study 3: Stress distribution in the defeatured model (Classifier includes loading direction information)	49
--	----

List of Tables

3.1	Diameter of holes in the four hole pattern versus maximum von Mises stress in the original and defeatured bracket when all the other simulation parameters are kept constant	19
3.2	Location of holes in the four hole pattern versus maximum von Mises stress in the original and defeatured bracket when all the other simulation parameters are kept constant	21
3.3	Direction of loading versus maximum von Mises stress in the original and defeatured bracket when all the other simulation parameters are kept constant	24
3.4	Encoding the direction of load	28
3.5	Sample of the dataset corresponding to through holes	30
3.6	Classification accuracy for various neural network architectures	32
3.7	Classification accuracy for various activation functions of the hidden layer neurons	33
3.8	Accuracy versus performance ratio parameter	34
3.9	Learning rate versus accuracy	34
4.1	Comparing original and defeatured Model	45
4.2	Case Study 2: Comparing original and defeatured Model	47
4.3	Case Study 3: Analysis of the effect of including forces in the classifier	49

Abbreviations

B-Rep Boundary Representation

CAD Computer Aided Design

CFD Computational Fluid Dynamics

FEA Finite Element Analysis

FEM Finite Element Method

NURBS Non Uniform Rational Basis Spline

PDE Partial Differential Equation

ROC Receiver Operating Characteristics

Chapter 1

Introduction

This chapter explains the motivation for defeaturing CAD models. The scope of the project is established and the organization of the report is mentioned.

1.1 Motivation for Defeaturing

CAD software is extensively used in applications from modeling parts and then evaluating the design through CFD analysis, FEM analysis to Product Life cycle management. Mechanical parts are designed in CAD software because of unambiguity in visualization and ease in manufacturing planning. Complex CAD models have a large number of geometric features and a complex topology with numerous faces, edges, vertices. A representation of the object with such a high level of detail occupies a large amount of computer memory too.

Such a detailed description is often not necessary for other downstream engineering applications like meshing and finite element analysis. Small geometric features can increase the size of a physical simulation to 10 times [12]. FEA is used to ensure that the designed part meets the specifications. FEM is a technique to find approximate solutions to boundary value problems for PDE's by dividing a large problem into simpler, smaller parts called finite elements. Meshing is an important operation in FEA where the complex object geometry is divided into a large number of small pieces, a grid of nodes is generated. Equations of equilibrium are solved at these nodes to determine the structural strength and deformation characteristics of the object.

When the complexity of the CAD model increases, the meshing procedure becomes more challenging. The meshing procedure takes a long time, algorithms generate poor quality meshes near small features like chamfers, holes, fillets, indentations etc. Poor mesh quality can lead to erroneous FEA results. Also, for a large mesh the finite element analysis time is long. Thus CAD models are simplified by suppressing detailed features

that don not affect the finite element analysis results and significantly decrease the mesh size and improve mesh quality.

In the field of computer graphics and virtual reality, mesh models are generally used. It is desirable to simplify these models to enable fast rendering. In this domain it is important to preserve the outer shape of the model after the simplification process.[3]

1.2 Scope of the Project

The objective of the present project was to use machine learning techniques and build a prototype system to identify suppressible holes / pockets in brackets subjected to FEA simulations for static structural analysis. Review of literature has been carried out to understand the relative advantages and limitations of existing defeaturing methods in solid models. Most of the existing literature to automate the defeaturing process focuses on using purely geometric measures like the dimensions of the geometric feature with respect to the dimensions of the part, the distance of the geometric feature from the displacement constraints and point of application of load to decide whether a geometric feature is suppressible. [9] [1].

Unlike all the other approaches, we also take into account the effect of direction of loading in determining suppressible features. A comprehensive dataset was generated through extensive FEA simulations on a chosen part family viz. brackets for the purpose of this study. Geometric features selected for this study include blind and through round holes and square pockets. The dataset consisted of variations in the sizes of these features, their location on the bracket and the direction of loading. The data from the FEA simulations was transformed to a form that can be processed by a classifier. Geometric features were labeled as suppressible or not suppressible based on the difference in the value of maximum von Mises stress in the bracket in the presence and absence of the feature. The dataset thus generated was further processed by a neural network. The neural network was trained on the training data and employed to predict the suppressibility of geometric features on the testing data.

1.3 Organization of the report

This report comprises of the following chapters:

Chapter 1 was a brief introduction about the background and listed the scope of the work.

Chapter 2 reviews the literature in the field of defeaturing and simplification of CAD models. Strategies to defeature CAD models based on surface entities and other methods

based on explicit features are studied. The chapter concludes with the objective of the current study.

Chapter 3 contains the details about the proposed prototype system.

Chapter 4 includes case studies to demonstrate that the proposed system achieves a reduction in the number of mesh elements without significant changes in the stress analysis results.

Chapter 5 summarizes the results of the current study and includes discussion about the considerations for future work

Chapter 2

Literature Survey

2.1 Scope

This chapter reviews the literature in the field of defeaturing and simplification of CAD models. Several papers were reviewed in this area, for the sake of brevity a few key ideas are discussed here. The chapter concludes with the objective of the project.

2.2 Simplification Approaches

Simplification operators take as input the model representation and return the simplified model. Based on the type of simplification operator the simplification approaches can be classified as surface entity based, volumetric entity based, explicit feature based and dimension reduction based [12]. Several approaches combine defeaturing with the meshing procedure or defeature models post the meshing procedure[8], these will not be discussed here.

2.2.1 Based on explicit features

Simplifying holes and fillets

In this section we discuss the approach proposed by Joshi and Dutta [5] to simplify holes and fillets in freeform surface models is discussed. The input to their system is a B-Rep model with NURBS representation for surface patches. They define a hole as a loop of edges with no surface on the inside.

The **Steps in the hole recognition algorithm** are:

1. Detect all free edges in the model F as shown in Fig.2.1 . A free edge is an edge with a surface on only one side.

2. For a free edge E_1 in F , find the start and end vertices $V_{1,start}$ and $V_{1,end}$
3. Find a free edge E_2 such that $V_{2,start}$ is $V_{1,end}$
4. Continue step 3 until an edge E_n is found such that $V_{n,end}$ is $V_{1,start}$. The edges E_1, E_2, \dots, E_n form a hole.
5. Steps 2-4 are repeated till all holes are detected.
6. Holes with perimeter below a certain threshold value are marked for suppression.

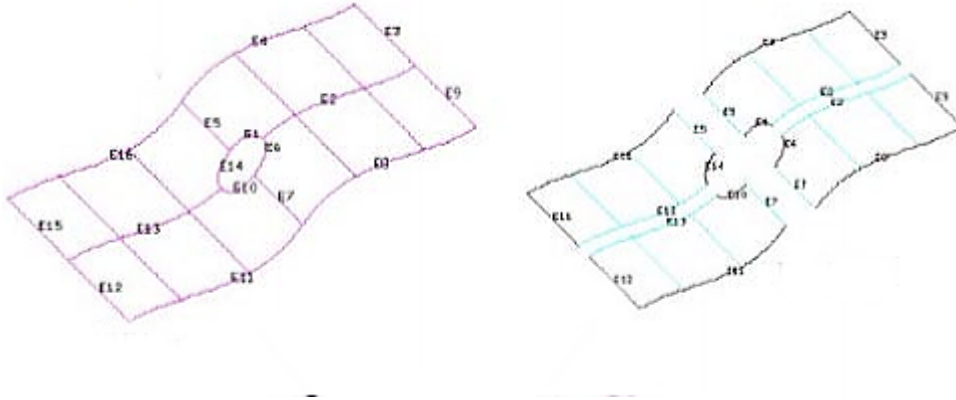


Figure 2.1: a) Edges constituting the sheet b) Detected free edges in black — Joshi and Dutta [5]

The next step is the suppression of holes. The hole is covered by a smooth surface patch. If the hole belongs to a single surface patch then, the hole is suppressed from its corresponding trimming curve. If the hole belongs to multiple surface patches and the underlying surfaces intersect, then the surfaces are trimmed at the intersection rather than the trimming curve specified by the hole.

The **results, advantages and limitations of the hole suppression module** are discussed now. In Fig 2.2 a) the model has 30 holes, the algorithm detected all holes. The holes at A & B did not meet the perimeter threshold for suppression and were hence not suppressed. However for the hole at C the surface patch creation algorithm failed and hence the hole still exists in the simplified model. An advantage is that this method is not limited to circular holes, however the suppression module can be further improved.

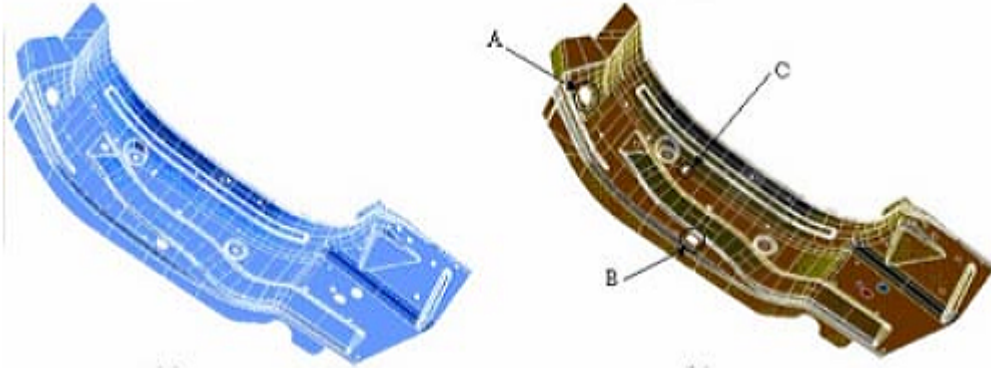


Figure 2.2: a) Original model with holes b) Defeatured model — Joshi and Dutta [5]

The **Blend suppression module** is discussed now:

A fillet or a blend is a transition between two faces with a sharp edge as shown in Fig.2.3. The method proposed in this paper uses variation in curvature across a surface to detect fillets. Each face is represented by a NURBS surface parametrised in two directions U and V . The **steps to detect a blend** are

1. For each face; traverse along the constant U and constant V curves and calculate the curvature at various points. The curvature of a surface $X(U,V)$ is calculated using equations in Fig.2.4.
2. If while traversing a constant U line along the V direction; K_v remains constant and is greater than a threshold; then this is a potential fillet in the V direction.
3. If the curvatures along multiple constant U lines is the same then it is marked as a constant radius fillet; otherwise it is marked as a variable radius fillet.
4. Additional steps ensure that surfaces like cylinders are not falsely classified as fillets.

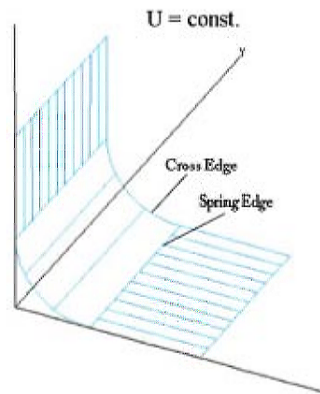


Figure 2.3: Fillet with constant parameter lines (Joshi and Dutta [5])

$$K_U = \frac{\left| \frac{\partial X}{\partial U} \times \frac{\partial^2 X}{\partial U^2} \right|}{\left| \frac{\partial X}{\partial U} \right|^3} \quad \& \quad K_V = \frac{\left| \frac{\partial X}{\partial V} \times \frac{\partial^2 X}{\partial V^2} \right|}{\left| \frac{\partial X}{\partial V} \right|^3}$$

Figure 2.4: Equations to calculate curvature (Joshi and Dutta [5])

The threshold in step 2 above takes into account the bounding box of the model so that the size of the model does not affect the algorithm.

The next step is to suppress fillets. For this purpose the intersecting edge of the two surfaces before the fillet was made is constructed. The procedure is as follows: At each cross -section of the fillet the point of intersection of the two support faces is constructed as shown in fig 2.5. A curve is constructed based on the points of intersection at various cross-sections along the length of the fillet. New surfaces are then constructed between the new curve and the support edges.

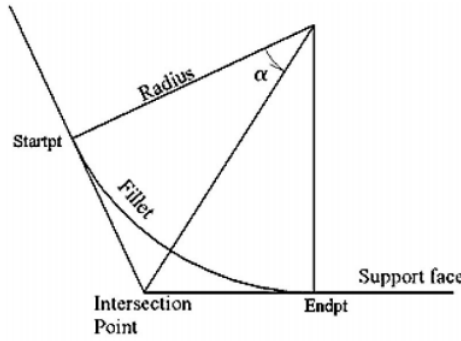


Figure 2.5: Point of intersection of supporting edges (Joshi and Dutta [5])

A limitation is that algorithm was restricted to blends with circular cross section. Also the blend faces were parametrised such that the blend radius was along a parametric direction.

2.2.2 Based on surface entities

Face Clustering based simplification

We discuss the face clustering approach reported by Sheffer [10]. It is to be noted that this is a simplification technique as opposed to defeaturing. The steps in the algorithm are as follows:

1. Cluster faces into regions and construct an adjacency graph as shown in Fig 2.6 a). Each node of the graph represents a cluster, an arc is used to connect adjacent faces.
2. Arcs are assigned weights as a measure of the improvement in shape properties on collapsing the arc. The procedure to assign weights is described below.
3. A priority queue is created and the arc with the highest weight is removed as shown in Fig 2.6 b). The graph is updated and the above procedure is carried out iteratively until no more arcs can be contracted.
4. Once the face clustering algorithm has terminated, the collapsible faces are detected. Faces that can be symmetrically split (collapsed) into adjacent clusters are collapsed as shown in Fig. 2.7 b).
5. the newly split faces are merged with the adjacent faces as shown in Fig. 2.7 c).

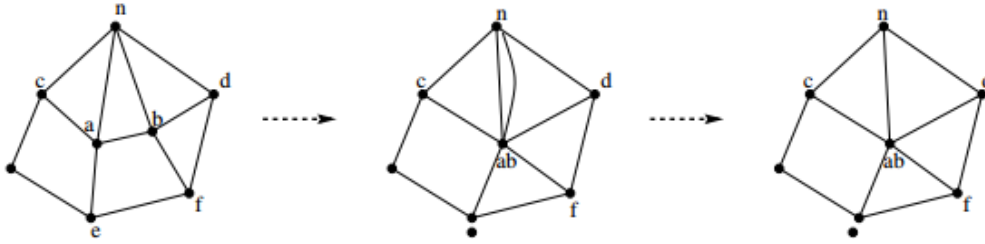


Figure 2.6: a) Adjacency graph b) merging of clusters c) removing duplicate edges (Sheffer [10])

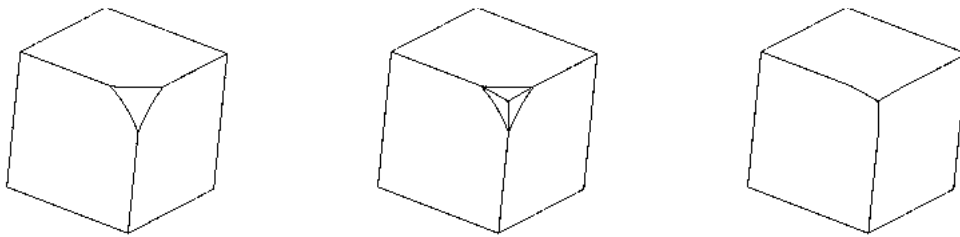


Figure 2.7: a)Original model b) A face is split symmetrically into 3 parts c) new parts are merged with the adjacent faces (Sheffer [10])

We will now discuss the **decision criteria** used for clustering. This amounts to the computation of weights for each arc in step 2 above. The weight for each arc is a weighted average of some geometric indices. These geometric indices quantify properties desirable for mesh generation and are as follows:

1. **Boundary preservation:** Edges at sharp boundaries between faces have to be retained to create good meshes. The index for edge angle is defined as

$$E_{a,b} = \begin{cases} -1 & \alpha_i < E_{min} \\ \frac{\sum_i^K \alpha_i - E_{min}}{K} / (\pi - E_{min}) & otherwise \end{cases}$$

where K is the number of shared edges between clusters, α_i is the dihedral angle between two clusters along edge i. Fig 2.8 illustrates how this index is calculated. The value of E will be 1 when the two faces are parallel. The two faces can be merged if the value of E is close to 1.

The remaining indices are described briefly.

2. **Region boundary shape** Smooth region boundary shapes are desirable for high mesh quality
3. **Region Smoothness** Regions with gradual change in curvature are desirable for high mesh quality

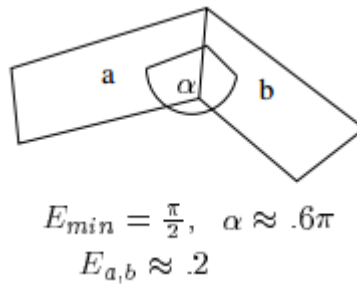


Figure 2.8: Computing the boundary angle index Sheffer [10]

We will now discuss the **results** of this approach. As seen in Fig. 2.9 faces corresponding to bends and fillets in the original model have been clustered thus reducing the total number of faces from 127 to 46.

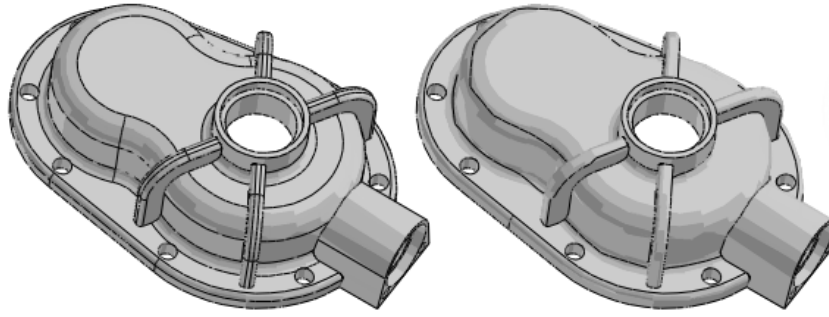


Figure 2.9: a) Original model 127 faces b) Simplified model 46 faces (Sheffer [10])

The **advantages and limitations** of the method are discussed below: The advantage of this method over feature based methods is that it can suppress small features that are not prominent. Also there is no need to define separate suppression rules for each type of feature present. The collapse detection algorithm allows for symmetrical partitioning of features like blends and fillets. An advantage of this method over previous clustering approaches is the ability to cluster faces into non-planar surfaces. It can also be applied to non-manifold topologies.

A limitation of face clustering approaches arises from the fact that the face clustering algorithm is greedy, and hence not necessarily optimal. Also the threshold values for geometric indices are heuristic. This approach is a model simplification approach and does not defeature a model. Hence it cannot be used to fill holes for instance.

Geometry-based size field

We now discuss the geometry based size field and facet based reduction operator by Quadros and Owen [8]. This method achieves automatic defeaturing. The input is model with B-rep topology and NURBS- based geometry definition without any feature data. The model is discretised before further processing.

There are two stages in this approach:

1. Unnecessary features are identified using a geometry based size field
2. Marked facets are removed using an edge collapse operator as shown in Fig. 2.11

The steps involved in **detecting features** are as follows

1. Identify geometric factors used to measure the complexity of surfaces and curves. These include 2D proximity, surface curvature for surfaces and 1D proximity, curve curvature, curve twist for curves.
2. The next step is to measure the geometric factors identified above. Proximity is measured by the distance, curvature is quantified by the radius of curvature.
3. The above quantities are used to calculate the maximum mesh size at a given point (x,y,z) . A size field function $s = f(x, y, z)$ is obtained as shown in Fig.2.10. If the computed mesh size less than a user defined threshold, the region is marked for suppression.

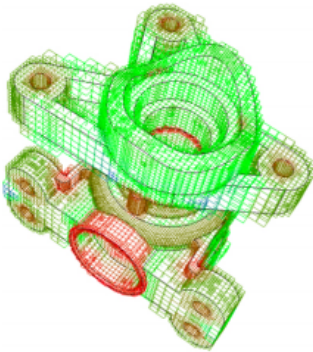


Figure 2.10: Color coded size field; in the red region the computed mesh size is less than the threshold

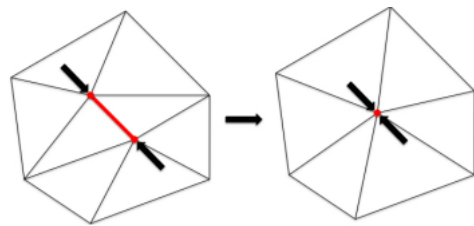


Figure 2.11: Edge collapse operator Quadros and Owen [8]

It is reported that measures based on curvature can be used to detect blends, holes and fillets.

The **results** are shown in Fig. 2.12 - 2.14. The original model in Fig. 2.12 has a long cylindrical hole, fillets at both convex and concave edges, multiple slots, and a protruded slab. These are not visible in Fig. 2.13. Fig. 2.14 shows the meshed models. The original model has 51,109 elements with average quality 0.75 the defeatured model has 3,490 elements with average quality 0.825

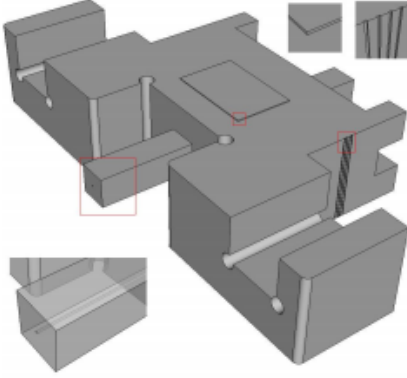


Figure 2.12: original model

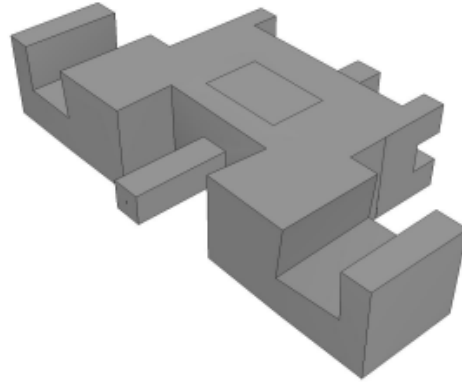


Figure 2.13: defeatured model (Quadros and Owen [8])

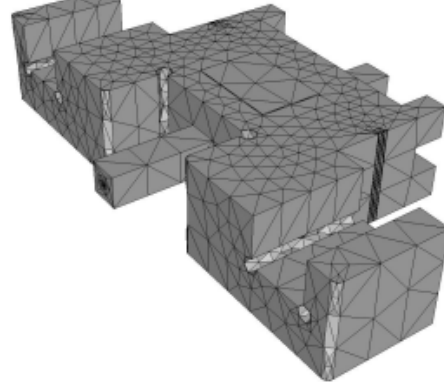
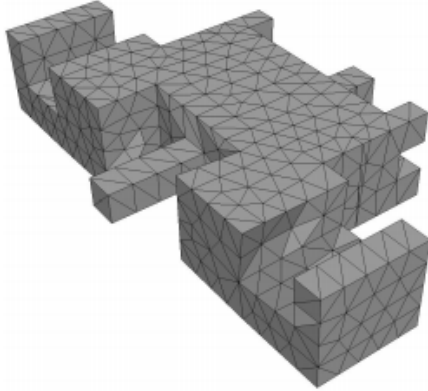


Figure 2.14: Left - Tet mesh on defeatured model — Right - Tet mesh on original model (Quadros and Owen [8])

The **advantages and limitations** of this method are as follows:

An advantage over feature based methods is that there is no need to define individual features to be suppressed. An important limitation is that it does take into account the boundary conditions and features with proximity to the boundary conditions or point of application of load. These have to be retained by manual intervention.

2.3 Summary and Conclusions

- Feature based approaches simplify a limited number of features and lack generality. To be completely effective, all problem causing features need to be removed. To achieve this goal, additional work is required in defining such features and incorporating them into the common framework [5]. This can be achieved by tailoring a simplification strategy to features from a particular functional domain.
- However not all features can be defined and techniques like the one proposed by [8] highlight the advantages of more generic measures to detect suppressible entities over techniques based on explicit feature definitions.
- Although these methods discuss ways in which a model can be simplified, the literature does not provide error estimations between the analyses of the original and simplified models.
- There is a lack of physics based error measures in the literature.
- There is a lack of physics based defeaturing rules in the literature and most of the work focuses on geometry based measures to identify suppressible features.

2.4 Objective of Project

The objective of the present project was to use machine learning techniques and build a prototype system to identify suppressible holes / pockets in brackets subjected to FEA simulations for static structural analysis. A comprehensive dataset was generated through extensive FEA simulations on brackets. The dataset consisted of brackets with holes (blind and through) and square pockets of various sizes. It also consisted of variations in the location of these geometric features on the bracket and the direction of loading.

The data from the FEA simulations was transformed to a form that can be processed by a classifier. This step is called feature engineering and involves using domain knowledge of the data to determine predictors that will help in classification. The predictors used in most studies are purely geometric measures. This study analyzed the effect of including the direction of loading in the set of predictors.

Chapter 3

Methodology

3.1 Overview

In this chapter, the proposed prototype system predicting the suppressibility of geometric features is described in detail. The system takes as input certain attributes about a geometric feature and the geometry of the parent part and suggests whether the feature should be removed from the 3D model of the part under consideration. The geometric features included in this study are blind and through holes and square pockets. Figure 3.1 summarizes the approach for the current study.

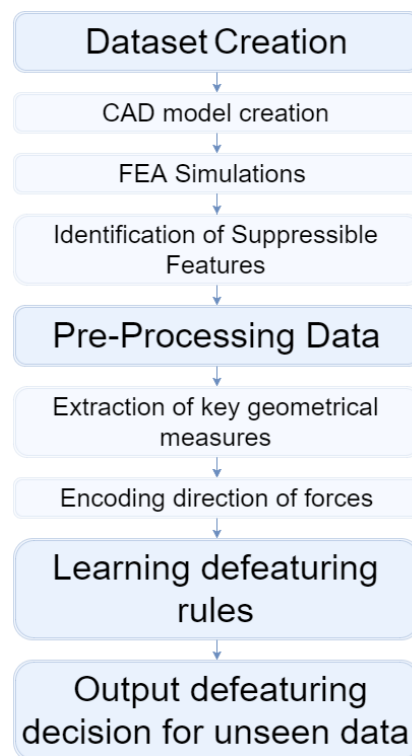


Figure 3.1: Summary of the approach to defeature CAD models.

Since the system is based on machine learning techniques- classification techniques to be more specific, it will automatically learn the conditions under which a feature is suppressible given some training data. The first step in the current study was gathering data. While some studies in this area have used expert knowledge in the construction of the dataset, this study relies on finite element simulations to assess which features are suppressible. In Section 3.2 the process for creating the dataset is explained. Section 3.3 contains the details of the pre-processing stage. It involves extracting key information from the finite element analysis studies and transforming the data such that it can then be processed by a classifier. This step is also called feature engineering in machine learning terminology and is more domain specific than the classifier. Once the data was pre-processed, a neural network classifier were used for learning the defeaturing rules. Section 3.4 contains the details about the classifier. The performance of the classifier was then tested on unseen data points. Section 3.5 contains the details about these experiments.

3.2 Generating the Dataset

3.2.1 CAD model creation

In the absence of expert knowledge about defeaturing decisions, this study employed finite element simulations to determine whether a geometric feature under consideration is suppressible. Since finite element simulations are time consuming the part for the current study was restricted to a geometric part family viz. bracket. The dimensions of the bracket are shown in figure 3.2. Through holes, blind holes and square pockets are the geometric features under consideration in this study. Figure 3.3a shows a pattern of 4 holes, figure 3.3b shows a pattern of 4 blind holes and figure 3.3c shows a pattern of 4 square pockets on a bracket part.

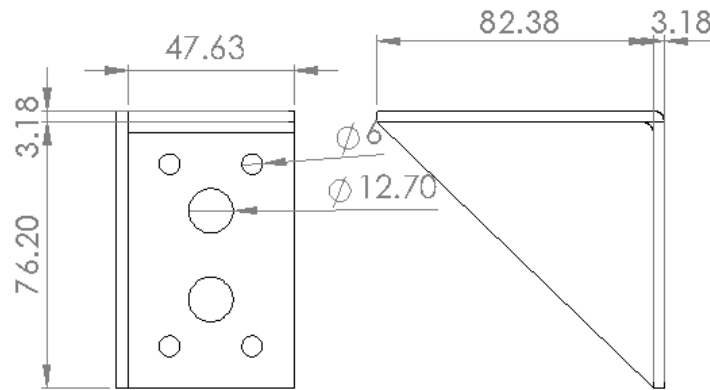


Figure 3.2: Dimensions of the bracket in millimeter

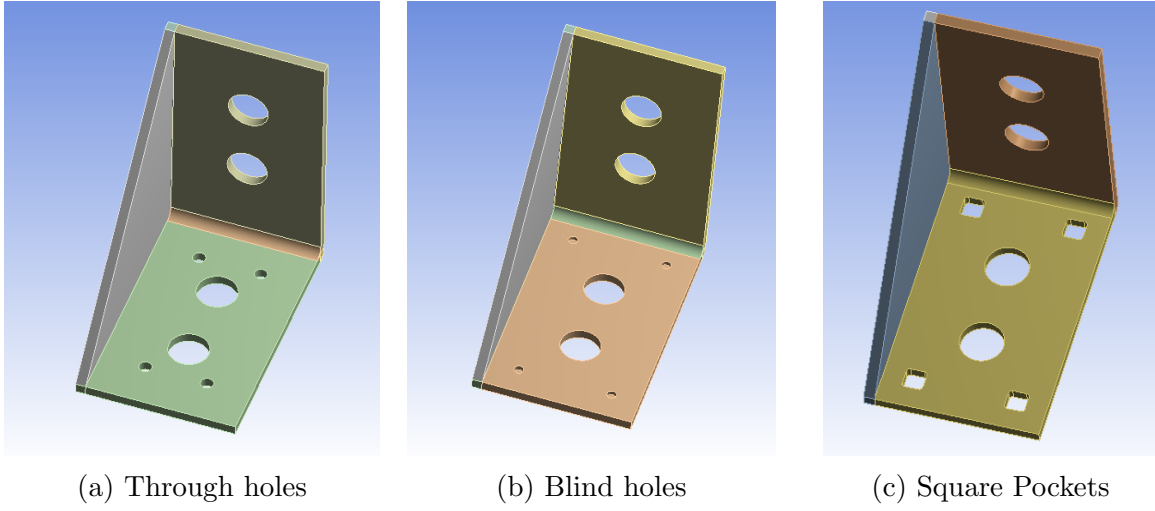


Figure 3.3: Illustration of a bracket with a pattern of four holes / pockets

The proposition is that each individual hole / pocket in the pattern, can be suppressed if the difference in the maximum von Mises stress in the bracket in the presence and absence of the hole / pocket is small. It is reported in literature that the size of a geometric feature relative to the size of a part and the distance of the geometric feature from the boundary conditions have a significant effect on the suppressibility [1] [9]. Previous studies have not considered the effect of the direction of loading on the suppressibility of features. A large dataset would help in creating a better classification model. Since generating a large dataset is not within the scope of the current study, it was attempted to generate the most informative data set by varying the direction of loading, the size of the hole / pocket and the distance of the hole / pocket from the boundary conditions in the brackets as discussed in the next section.

3.2.2 Finite Element Simulations

The simulations were run as parametric studies in ANSYS Workbench. The simulation parameters that were constant for all the simulations are mentioned below:

- **Displacement Constraint:**

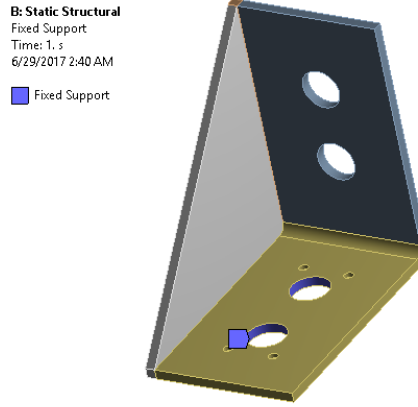


Figure 3.4: The surfaces highlighted are constrained to be fixed

- **Meshing Method:** The hex-dominant meshing method in ANSYS was used to generate the mesh. In regions where hexahedral elements cannot be used, other type of elements are used. These are 10 node tetrahedral elements, 15 node wedge elements and 13 node pyramid elements.

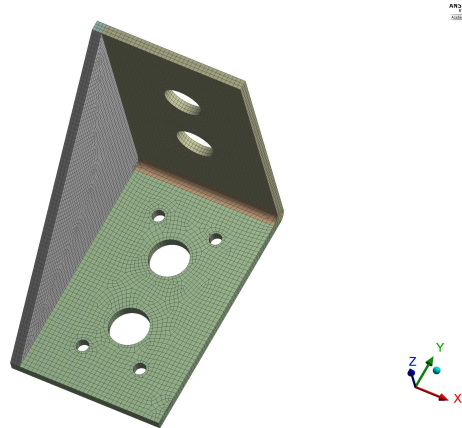


Figure 3.5: Majority of the elements in the mesh are hexahedral elements with 20 nodes.

- **Dealing with Stress Singularities:** Stress singularity occurs in the mesh near the fixed support boundary condition. Because of this it was observed that the maximum value of von Mises stress in the bracket increased when the size of mesh

elements was reduced. Also the location of the maximum stress was at the elements where this boundary condition was applied. Mesh convergence was not observed. To deal with stress singularity occurring at the fixed support, results at mesh elements close to the fixed support boundary condition were ignored. The rule of thumb to ignore results within a 2-3 mesh elements distance was used. Only results in the region highlighted in figure 3.6 are considered

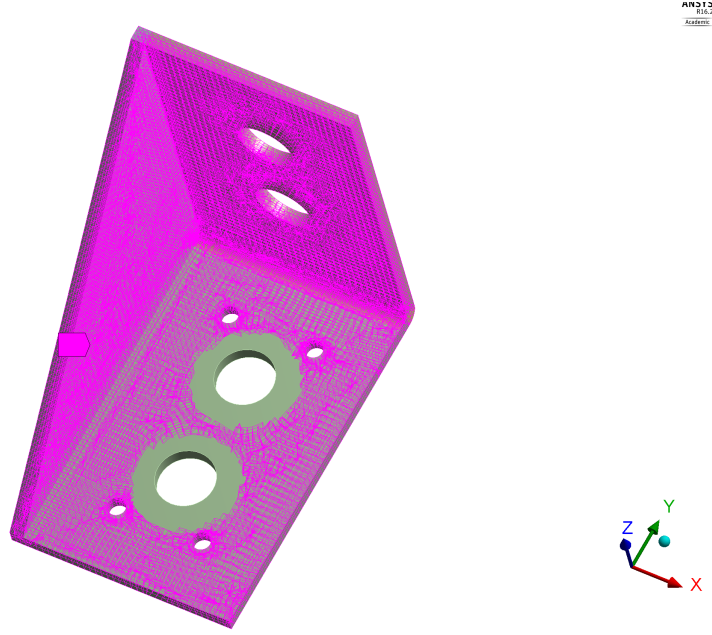


Figure 3.6: Region of interest in the mesh

F_x , F_y , F_z -the forces in the X, Y and Z direction (figure 3.10), diameter (round holes) / side length (square hole) (figure 3.7), distance d (the location of the hole / pocket) as shown in figure 3.9 were the input parameters in the parametric ANSYS studies. The output parameter in the study was the maximum von Mises stress in the region shown in the figure 3.6 above. This section illustrates how variations in the size of the geometric features, their location on the bracket and the direction of loading affect the stress analysis results.

- **Variation of diameter of holes:** Simulations were run for geometries with the diameter of holes in the 4 hole pattern taking the values 3, 4, 6, 8, 10, 12 millimeter. Figures 3.7a, 3.7b illustrates a bracket in which the location of the centroid of the 4 hole pattern is fixed but the diameter is varied. Figures 3.7c, 3.7d illustrates a bracket in which one of the holes is suppressed.

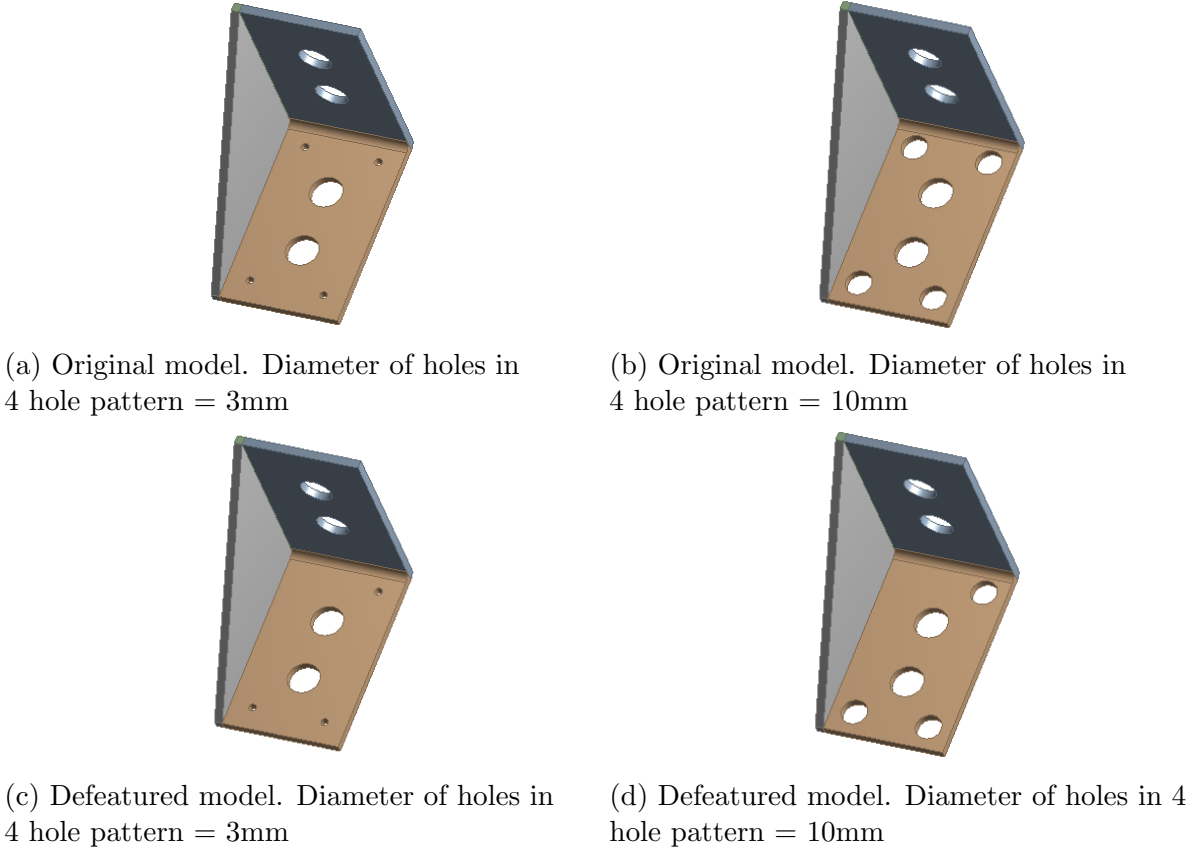


Figure 3.7: Illustration of variation of diameter in a pattern of four through holes

Sr no	Diameter of holes (mm)	$\sigma_{v,max}$ in original bracket (Pa)	$\sigma_{v,max}$ in defeatured bracket (Pa)	$\frac{\sigma_{v,max} \text{ with feature}}{\sigma_{v,max} \text{ without feature}}$
1	3	$11,314 \times 10^3$	$12,450 \times 10^3$	0.908
2	4	$11,219 \times 10^3$	$11,875 \times 10^3$	0.944
3	6	$11,777 \times 10^3$	$11,941 \times 10^3$	0.986
4	8	$11,812 \times 10^3$	$12,303 \times 10^3$	0.960
5	10	$13,449 \times 10^3$	$14,384 \times 10^3$	0.930
6	12	$16,365 \times 10^3$	$14,937 \times 10^3$	1.095

Table 3.1: Diameter of holes in the four hole pattern versus maximum von Mises stress in the original and defeatured bracket when all the other simulation parameters are kept constant

Table 3.1 compares the maximum von Mises stress in the original model-figures 3.7a, 3.7b and defeatured model- figures 3.7c, 3.7d when the diameter of holes is varied keeping all other simulation parameters constant. $\sigma_{v,max}$ denotes the maximum von Mises stress in the bracket. We see that when the diameter of the holes is 12 mm the defeatured model gives a misleading low value of the von Mises stress (about 10

% lower). Hence suppressing a hole with this diameter is not advisable. Defeating holes with diameters from 3 mm to 8 mm results in stress values higher than the original model. As explained in section 3.3.2, such holes are labeled as suppressible in this study.

- **Variation of location of holes:** Simulations were run for geometries in which the location of holes in the four hole pattern was varied by the changing the distance d highlighted in figure 3.9. The values taken by d are 8, 9, 10, 11, 12, 13, 14, 16 millimeter.

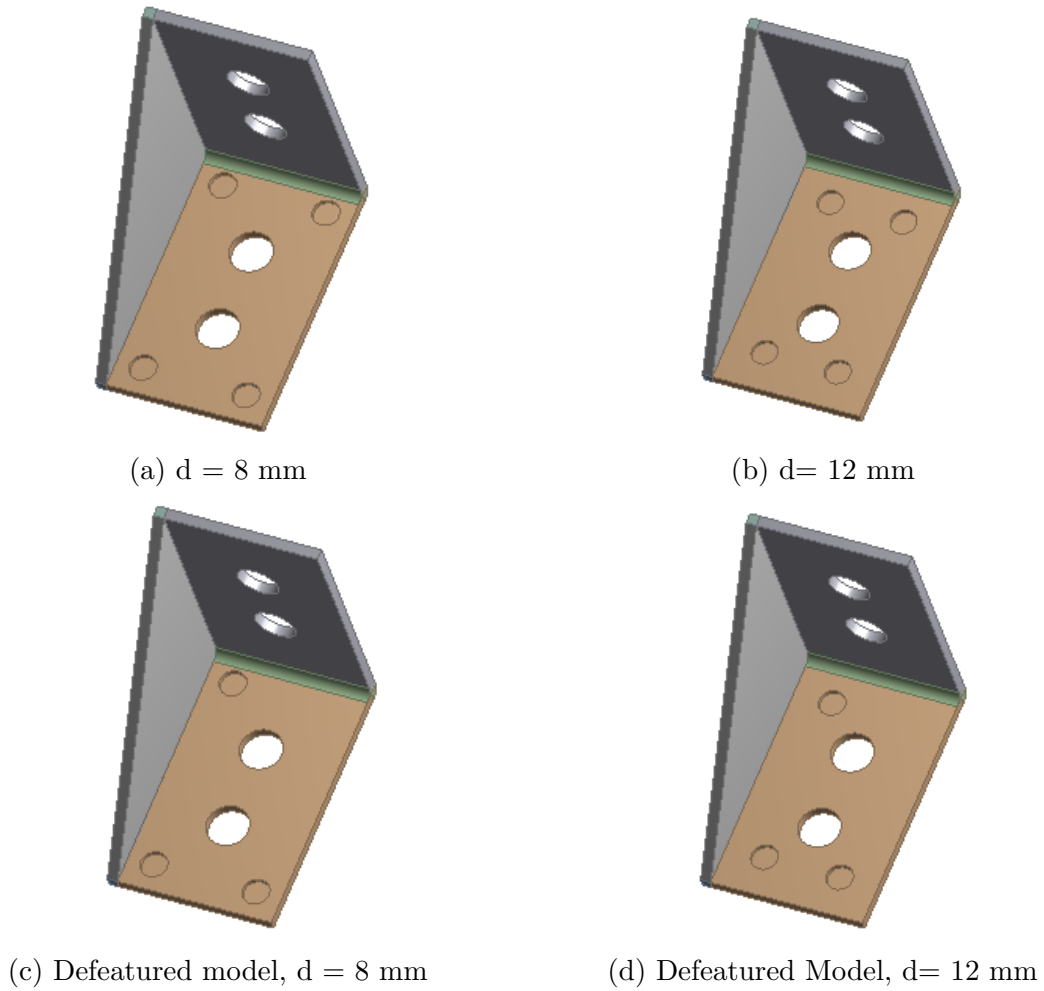


Figure 3.8: Illustration of variation of location of holes in a pattern of four blind holes

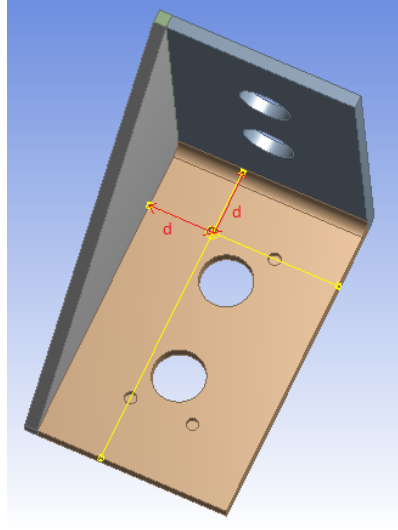


Figure 3.9: The location of holes in the pattern was varied by changing the distance d

Sr no	location parameter: d (mm)	Maximum von Mises Stress in original bracket (Pa)	Maximum von Mises Stress in defeatured bracket (Pa)	$\frac{\text{maximum stress with feature}}{\text{maximum stress without feature}}$
1	8	$12,655 \times 10^3$	$12,676 \times 10^3$	0.9983
2	9	$12,756 \times 10^3$	$12,154 \times 10^3$	1.049
3	10	$12,632 \times 10^3$	$11,934 \times 10^3$	1.058
4	11	$12,699 \times 10^3$	$11,509 \times 10^3$	1.103
5	12	$14,716 \times 10^3$	$11,498 \times 10^3$	1.279

Table 3.2: Location of holes in the four hole pattern versus maximum von Mises stress in the original and defeatured bracket when all the other simulation parameters are kept constant

Table 3.2 compares the maximum von Mises stress in the original model-figures 3.8a, 3.8b and defeatured model- figures 3.8c, 3.8d when the location of holes is varied (by changing the parameter d shown in figure 3.9) keeping all other simulation parameters constant. We see that when the parameter d increases, the defeatured model gives a misleading low value of the von Mises stress. Suppressing the hole feature in such situations is not advisable.

This example illustrates the importance of considering the location of the geometric feature when making defeaturing decisions.

- **Varying the direction of force:** In all the simulations the force is applied at the boundary of the holes shown in the figure 3.10 below. The direction of loading is varied in the X-Y plane

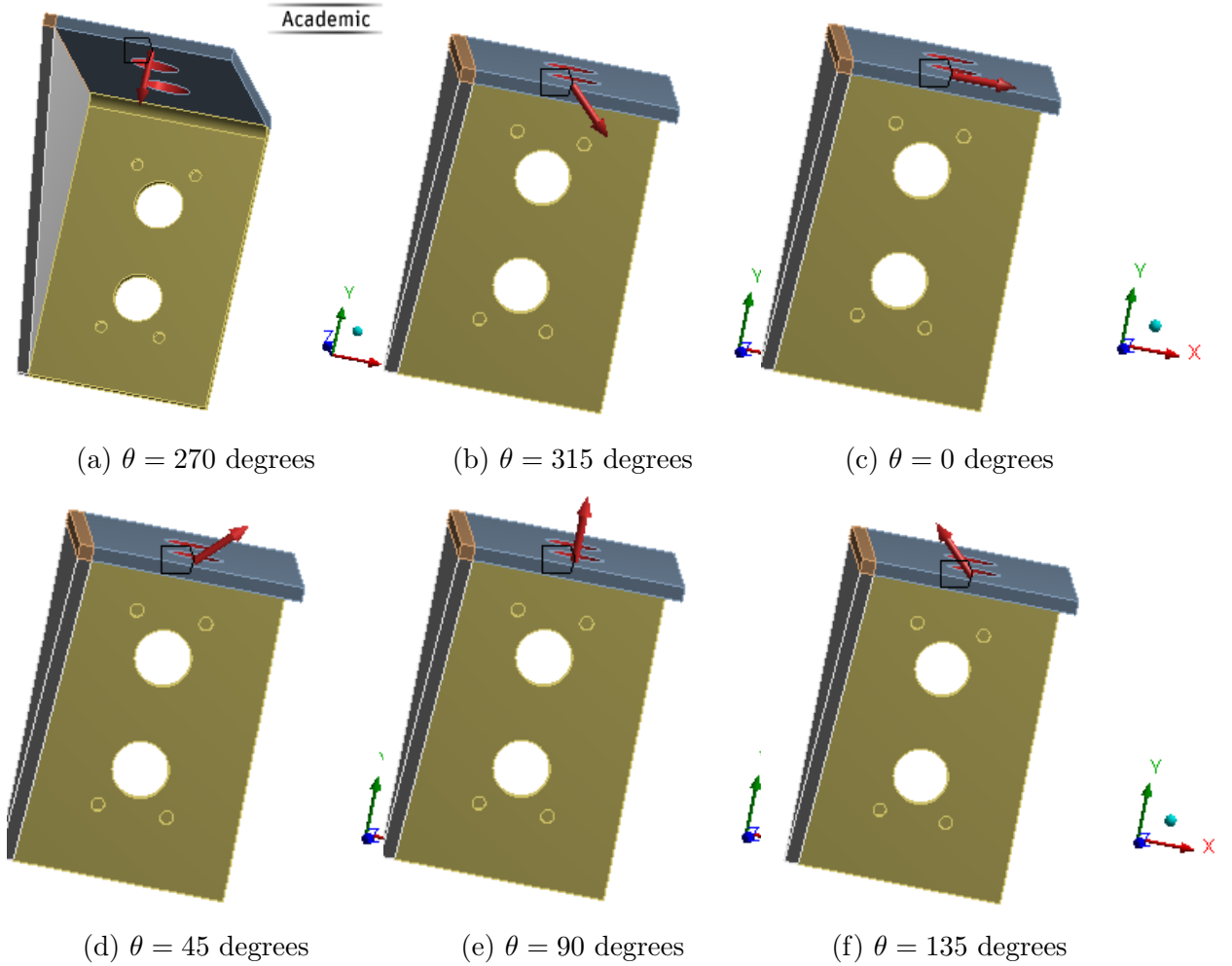


Figure 3.10: Illustration of variation of loading direction in a pattern of 4 blind holes

The figure 3.11 shows the number of data points generated for each direction of loading. θ is the angle measured from the positive X- axis in the global co-ordinate system shown in figure above.

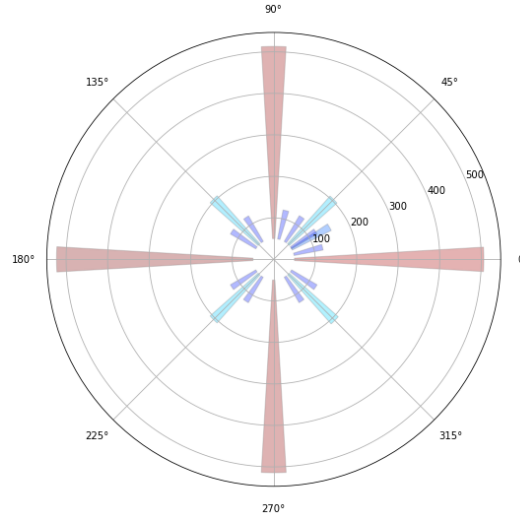


Figure 3.11: Polar plot: Number of training examples for each direction of load θ

Table 3.3 compares the maximum von Mises stress in the original model-figure 3.12a and defeatured models- figures 3.12b, 3.12c when the direction of load is varied (by changing the parameter θ shown in figure 3.10 keeping all other simulation parameters constant, where $\sigma_{v,max}$ is the maximum von Mises stress in the bracket and y_0 is given by

$$y_0 = \frac{\text{maximum stress with feature}}{\text{maximum stress without feature}} \quad (3.1a)$$

$$y_0 = \frac{\sigma_{v,max} \text{ with feature}}{\sigma_{v,max} \text{ without feature}} \quad (3.1b)$$

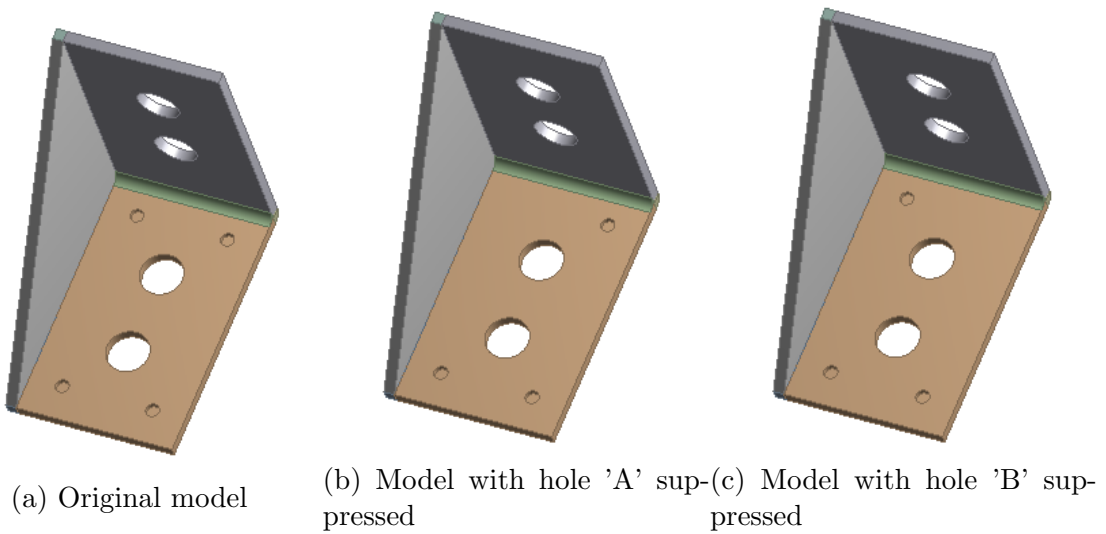


Figure 3.12: Model of the original and defeatured bracket

Sr no	θ (degrees)	$\sigma_{v,max}$ (Pa) in original bracket	$\sigma_{v,max}$ (Pa) in defeatured bracket (Hole 'A' suppressed)	y_0 (Hole 'A' suppressed)	$\sigma_{v,max}$ (Pa) in defeatured bracket (Hole 'B' suppressed)	y_0 (Hole 'B' suppressed)
1	0	1.1325×10^7	1.1317×10^7	1.00	1.128×10^7	1.00
2	45	1.297×10^7	1.323×10^7	0.98	1.186×10^7	1.09
3	90	1.190×10^7	1.207×10^7	0.98	1.032×10^7	1.153
4	135	1.924×10^7	2.09×10^7	0.92	2.031×10^7	0.947
5	180	1.131×10^7	1.226×10^7	0.922	1.22×10^7	0.922

Table 3.3: Direction of loading versus maximum von Mises stress in the original and defeatured bracket when all the other simulation parameters are kept constant

We see that suppressing hole 'A' does not lead to a lower value of $\sigma_{v,max}$, in fact the value of $\sigma_{v,max}$ is higher. Hence, hole 'A' can be suppressed for the cases considered in the table 3.3. However, in the case of hole 'B' for $\theta = 45$ and $\theta = 90$ suppressing the hole leads to lower values of $\sigma_{v,max}$ than the original model. It was observed that the location of $\sigma_{v,max}$ is at hole 'B' for these values of θ . Hence suppressing hole 'B' is not advisable. This example illustrates the importance of considering the direction of loading when making defeaturing decisions.

3.2.3 Simulation Runtime Details:

About 2500 FEA simulations were run for each type of feature i.e through holes, blind holes and square pockets. In total about 7500 simulations were run. The simulations were run in ANSYS Workbench as parametric studies. Generating the dataset required about 650 hours (26 days) of FEA simulations on a Windows 10 PC with the following technical specifications:

CPU: Intel(R) Core(TM) i5 CPU M56@2.67GHz

Installed RAM: 8.00 GB

3.3 Pre-Processing Data

3.3.1 Inputs to the classifier

This step involves transforming the data from the FEA simulations into attributes that can be processed by the classifier. In machine learning terminology, these attributes are called features. In classification problems, features describe an individual instance whose class is to be predicted. This is not to be confused with the geometric features subject to the suppressibility analysis in this study. Each feature used to describe an individual hole / pocket instance is explained in detail below:

1. **Volume Gain:** This is calculated as the ratio of the volume of the part after defeaturing to the volume of the part before defeaturing. Since ANSYS Workbench does not output the volume of the part in the parametric study, this had to be computed in the pre-processing stage. The volume of the part without the pattern (fig 3.13a) is $V = 33,542 \text{ mm}^3$. The volume of defeatured part as a function of the diameter (D) is computed using the equations 3.2. The thickness of the part, $t = 3.175 \text{ mm}$.

$$\text{Volume gain} = \frac{\text{Volume of defeatured model}}{\text{Volume of original model}} \quad (3.2a)$$

$$\text{Volume of defeatured model} = V - 3\pi \left(\frac{D}{2}\right)^2 t \quad (3.2b)$$

$$\text{Volume of original model} = V - 4\pi \left(\frac{D}{2}\right)^2 t \quad (3.2c)$$

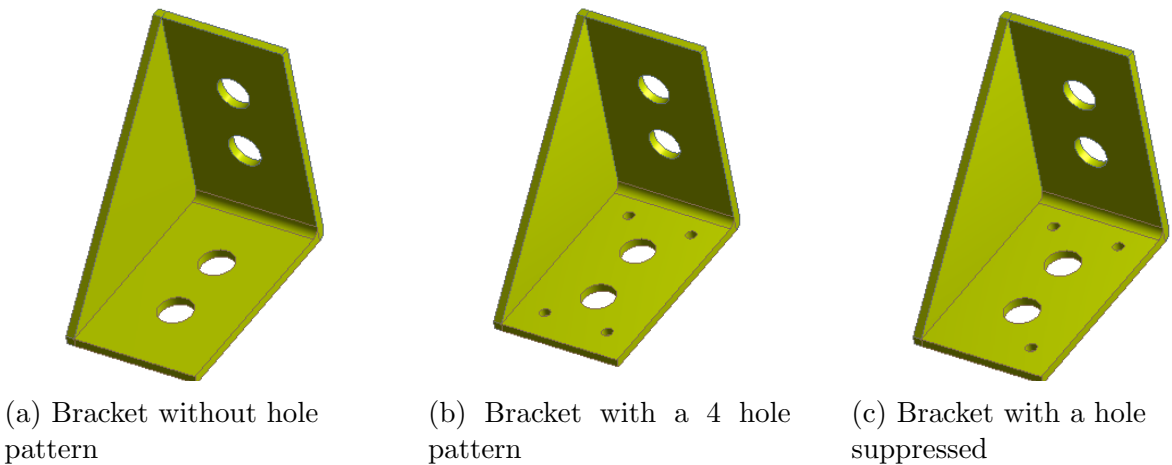


Figure 3.13: Calculating Volume and Surface area gain

2. **Surface area gain:** This is calculated as the ratio of the surface area of the part after defeaturing to the surface area of the part before defeaturing. Similar to volume gain the surface area gain also had to be computed as a function of diameter D . The surface area of the part without the pattern was found to be $S = 22,694mm^2$. The equations 3.3 show the computation of Surface area gain for through holes.

$$\text{Surface area gain} = \frac{\text{Surface area of defeatured model}}{\text{Surface area of original model}} \quad (3.3a)$$

$$\text{Surface area of defeatured model} = S - 3 \times 2\pi \left(\frac{D}{2}\right)^2 + 3 \times \pi Dt \quad (3.3b)$$

$$\text{Surface area of original model} = S - 4 \times 2\pi \left(\frac{D}{2}\right)^2 + 4 \times \pi Dt \quad (3.3c)$$

3. Ratio of dimensions of the feature:

For through holes this is given by the ratio diameter / thickness of the part. For blind holes this is given by the ratio diameter / depth of hole. All blind holes in this study had depth equal to half of the thickness of the part.

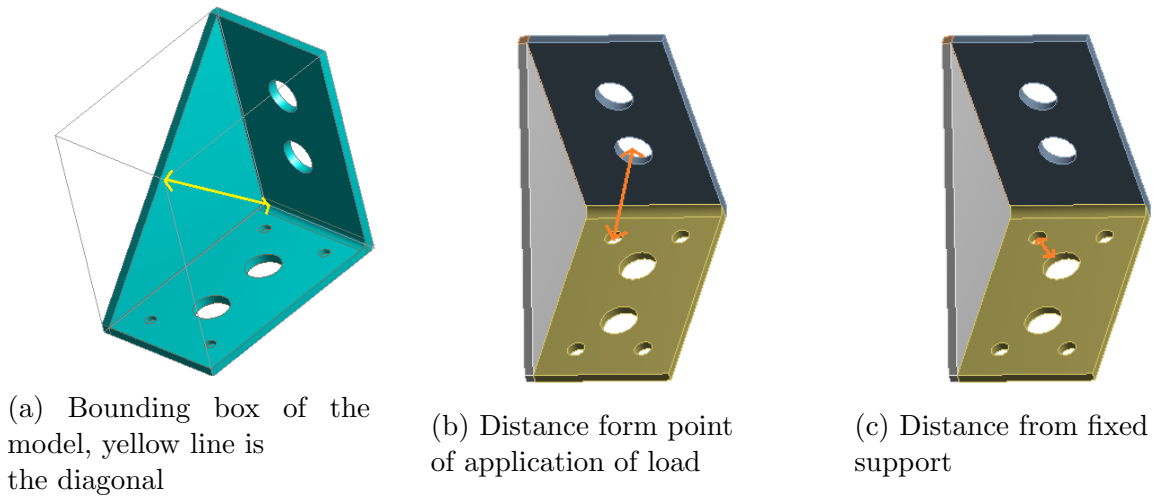


Figure 3.14: Calculating Normalized distance from boundary conditions

4. **Normalized distance from the point of application of load** The distance from the point of application of load is calculated as the Euclidean distance between the centroid of the geometric feature and the nearest point of application of load as shown in figure 3.14b. This distance however is only useful when it is normalized by some measure of the size of the part. The normalizing measure used here is the length of the body diagonal of the bounding box of the model as shown in figure 3.14a. The purpose of normalization is to make this input feature relevant to parts irrespective of their size.

5. **Normalized distance from the nearest displacement constraint** The distance from the nearest displacement constraint is calculated as the Euclidean distance between the centroid of a geometric feature and the nearest displacement constraint as shown in figure 3.14c. It is also normalized by the length of the body diagonal of the bounding box of the model.

6. **Normalized distance from the center of gravity** It was observed that because of the symmetry in the model all the above features were equal for two distinct holes shown by red and green in the figure 3.15. Hence to break the symmetry this distance was added to the feature set.

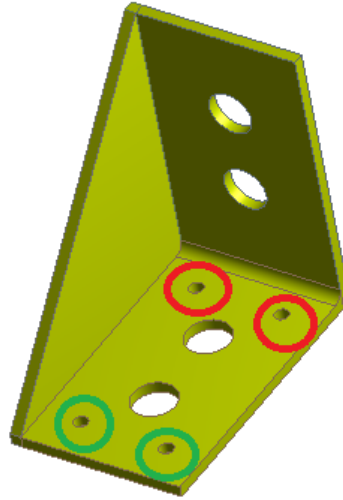


Figure 3.15: Holes with same descriptors

7. **Direction of loading** As mentioned in the section 3.2.1 for the purpose of this study the direction of forces was varied in the XY plane. The XY plane was divided into 8 regions shown in figure . The training data was generated with loading along the following directions. In the polar plot in fig . the magnitude of the radius represents the number of training data points.

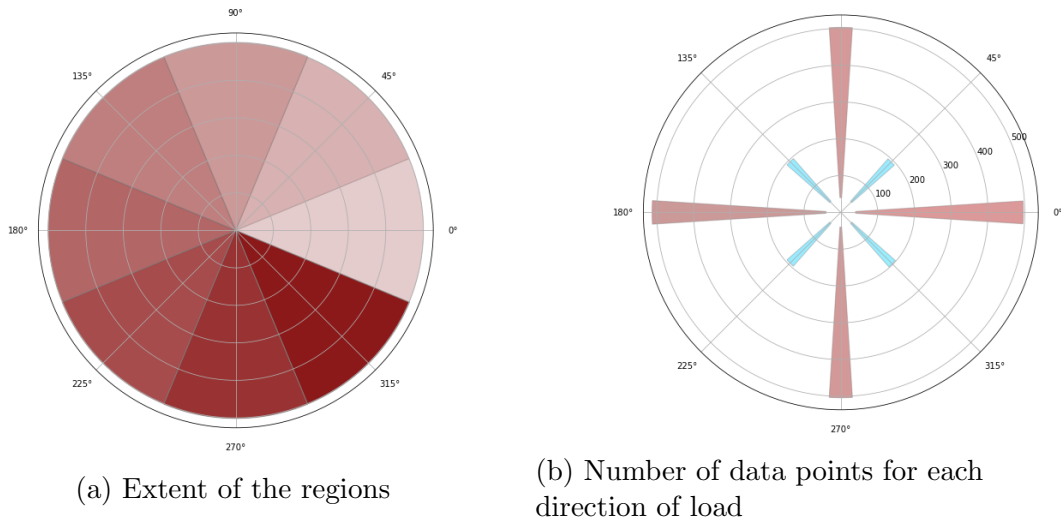


Figure 3.16: Division of XY plane into 8 regions

The direction of loading was encoded as a categorical variable to be fed to the classifier. The table 3.4 shows the encoding

Sr. no	Range of θ (degrees)	Categorical encoding							
1	$[-22.5, 22.5)$	1	0	0	0	0	0	0	0
2	$[22.5, 67.5)$	0	1	0	0	0	0	0	0
3	$[67.5, 112.5)$	0	0	1	0	0	0	0	0
4	$[112.5, 157.5)$	0	0	0	1	0	0	0	0
5	$[157.5, 202.5)$	0	0	0	0	1	0	0	0
6	$[202.5, 247.5)$	0	0	0	0	0	1	0	0
7	$[247.5, 292.5)$	0	0	0	0	0	0	1	0
8	$[292.5, 337.5)$	0	0	0	0	0	0	0	1

Table 3.4: Encoding the direction of load

3.3.2 Identification of Suppressible Holes

To determine the effect of suppressing a feature, the FEA simulation was run with and without the feature.

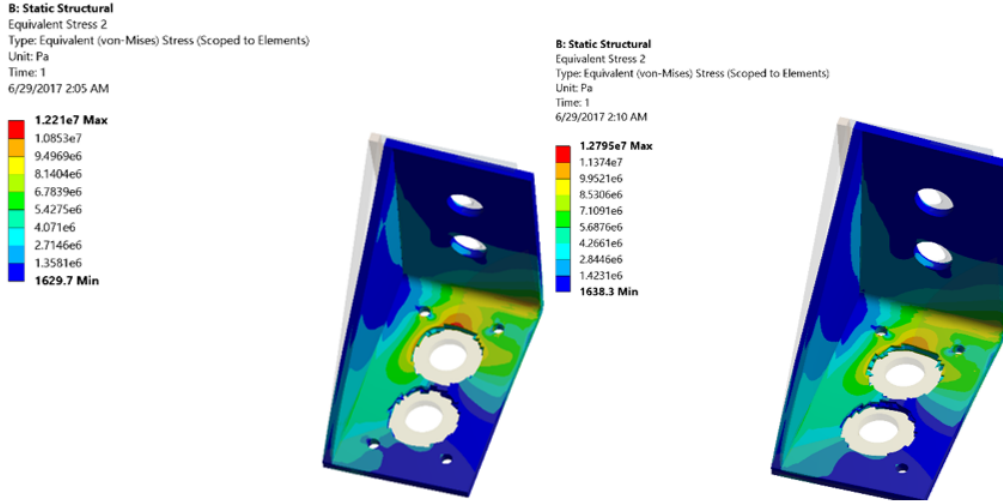


Figure 3.17: Stress with and without a hole feature.

The ratio of the maximum von Mises stress in the above two cases was used as a decision parameter.

$$y_0 = \frac{\text{maximum stress with feature}}{\text{maximum stress without feature}} \quad (3.4)$$

If this ratio was less than a certain threshold value the feature was labeled as suppressible. The threshold used in this study was 1.05

$$y = \begin{cases} 1, & \text{if } y_0 \leq 1.05 \\ 0, & \text{otherwise} \end{cases} \quad (3.5)$$

In the case shown in figure 3.17 the ratio is less than the threshold and hence the feature is labeled suppressible.

Table 3.5 shows a part of the dataset corresponding to through holes. This will be processed by the neural network.

Sr. no	Predictors														Response	
	X1	X2	X3	X4	X5	X6-X13								X14	y	1-y
1	1.0013	0.998	3.779	0.2478	0.0737	1	0	0	0	0	0	0	0	0.1120	1	0
2	1.0055	0.997	7.559	0.2435	0.0948	1	0	0	0	0	0	0	0	0.1004	0	1
3	1.0003	0.999	1.889	0.2478	0.0857	1	0	0	0	0	0	0	0	0.1120	1	0
4	1.0003	0.999	1.889	0.2446	0.1082	1	0	0	0	0	0	0	0	0.1039	0	1
⋮	⋮	⋮	⋮	⋮		⋮									⋮	
1951	1.0003	0.999	1.889	0.2446	0.1082	1	0	0	0	0	0	0	0	0.1039	0	1

Table 3.5: Sample of the dataset corresponding to through holes

where,

- X1: Volume gain
 X2: Surface area gain
 X3: D/t diameter of hole / thickness of bracket
 X4: Normalized distance from point of application of load
 X5: Normalized distance from fixed support
 X6-X13: Direction of loading
 X14: Normalized distance from center of gravity
 y: Defeaturig decision $y = 1$ indicates the feature is suppressible,
 $y = 0$ indicates the feature is not suppressible

3.4 Classification Model

The dataset was divided randomly into the training set, validation set and testing set. 70 % of the data was used for training, 15% was used for validation and 15% was used for testing. A feedforward neural network with two hidden layers was chosen for the classification model (figure 3.18). The network consists of 14 input neurons, two hidden layers with 64 neurons each, and an output layer with 2 neurons. The output layer has two neurons since the geometric features can be classified into two classes- suppressible and not suppressible. The output of each output neuron is the score for that particular class. Each neuron in a layer is connected to all the neurons in the subsequent layer. The strength of these connections is given by the weights W . These weights are computed by minimizing a loss function in the training phase. The training process stops when there is no significant improvement in the classification accuracy on the validation set.

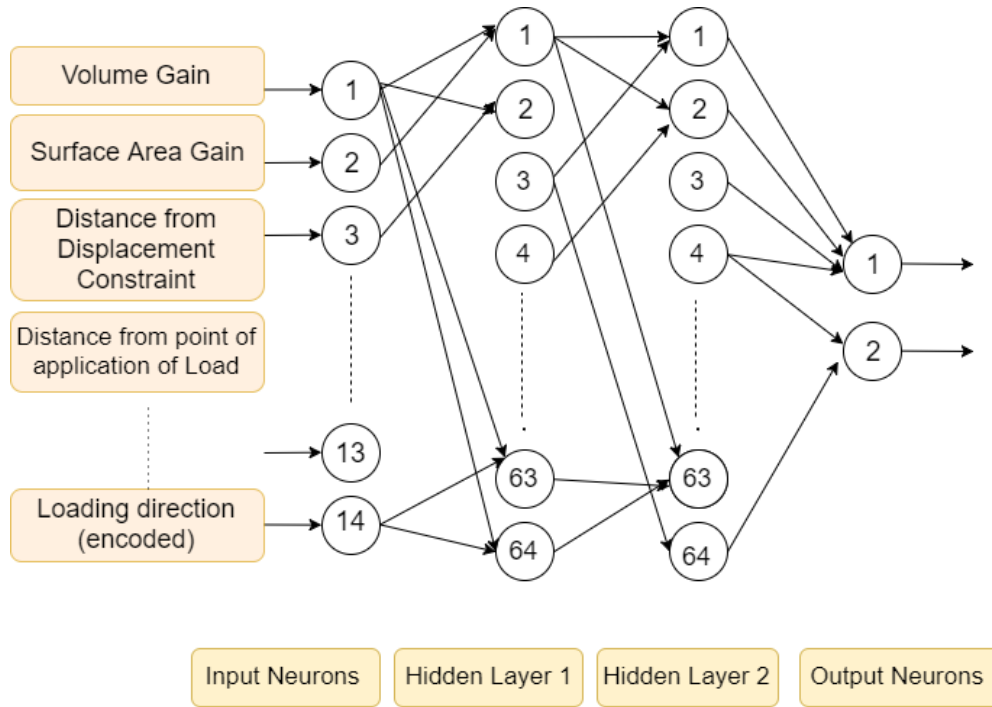


Figure 3.18: Architecture of the neural network

Choosing the neural network architecture

Classification accuracy is defined as the fraction of total samples that were correctly classified. The below table 3.6 summarizes the classification accuracy for various architectures of the network. Two hidden layers with 64 neurons each were used since this architecture outperformed other architectures tested.

Sr. no	Number of neurons	Accuracy (single hidden layer)	Accuracy (two hidden layers)
1	2	0.781	0.774
2	4	0.802	0.788
3	8	0.808	0.873
4	16	0.825	0.887
5	32	0.897	0.890
6	64	0.914	0.924
7	128	0.897	0.877
8	256	0.894	0.890
9	512	0.870	0.883
10	1024	0.894	0.894

Table 3.6: Classification accuracy for various neural network architectures

Choosing activation function for the hidden layers

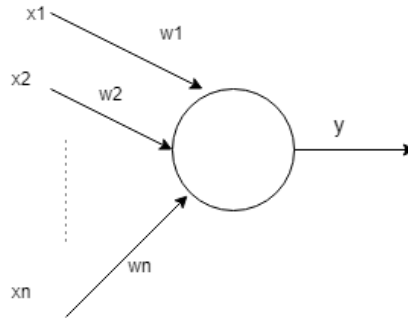


Figure 3.19: Output of a neuron

The output of a neuron is given by $y(x) = f(W^T x)$, (figure 3.7) where W denotes the weights and f is the activation function. The below table 3.7 summarizes the classification accuracy for various activation functions available in the MATLAB neural network toolbox. Since the weights are initialized randomly and the dataset is split into training, testing and validation set randomly, the accuracy of the network is not the same in every run. Thus, mean accuracy over 10 iterations is used.

Sr no	Activation function	Mean accuracy over 10 iterations
1	Transig	0.884
2	Elliot'sig	0.890
3	Logsig	0.900
4	Hardlims	0.777
5	Tribas	0.882
6	Radbas	0.878

Table 3.7: Classification accuracy for various activation functions of the hidden layer neurons

It can be seen that the using the all the activation functions tested have comparable performance other than the hardlims function. The tansig (hyperbolic tangent sigmoid) activation function was used for neurons in the hidden layers. The output layer uses the soft-max activation function.

Setting the performance ratio parameter for regularization:

Overfitting occurs when a network performs well on training data but not on testing data. Regularization is used to prevent overfitting. L_i in 3.6a is the cross-entropy loss for each training example, L is the average loss over all training data points, N is the number of training data points. y_i is the true label, f_j is the j^{th} element of the vector of class scores f [11]. Equation 3.6b shows the loss function used in training neural networks with the performance ratio parameter γ , where W_m are the weights of the network.

$$L = \frac{1}{N} \sum_{i=1}^N L_i = \frac{1}{N} \sum_{i=1}^N -\log \left(\frac{e^{f_{y_i}}}{\sum_{j=1}^2 e^{f_j}} \right) \quad (3.6a)$$

$$L \text{ using regularization} = \gamma \frac{1}{N} \sum_{i=1}^N L_i + (1 - \gamma) \sum_{m=1}^M W_m^2 \quad (3.6b)$$

The below table 3.8 summarizes the classification accuracy for various values of the performance ratio parameter.

Sr no	Parameter	Mean accuracy over 10 iterations
1	0.2	0.8939
2	0.3	0.8809
3	0.4	0.8805
4	0.5	0.8836
5	0.6	0.8980
6	0.7	0.8959
7	0.8	0.9068
8	0.9	0.8884

Table 3.8: Accuracy versus performance ratio parameter

The performance ratio parameter was set to 0.8

Setting the learning rate:

The learning rate determines by how much weights are updated in each learning epoch. The table 3.9 summarizes the classification accuracy for each learning rate. The learning rate chosen was 0.025/64

Sr no	Learning rate	Mean accuracy over 10 trials
1	0.4	0.9027
2	0.1	0.8942
3	0.025	0.8976
4	0.025/4	0.8925
5	0.025/16	0.8959
6	0.025/64	0.9020

Table 3.9: Learning rate versus accuracy

Output of the Classifier

The output of each neuron in the output layer is the score for that particular class. It is normalized so that for a given sample data point, the sum of scores over all output neurons is equal to 1. If the score at output neuron 1 is greater than a threshold value p , then the sample point is assigned class 0 otherwise it is assigned class 1. Class 1 means that the geometric feature corresponding to the sample data point is suppressible. Figure 3.20 below summarizes the classification performance for various values of the threshold value p . True positive refers to features that are not suppressible and correctly classified by the network. True negative refers to features that are suppressible and correctly classified by the network. False positive refers to features that are suppressible but are incorrectly

classified as unsuppressible. False negative refers to features that are unsuppressible but are incorrectly classified as suppressible.

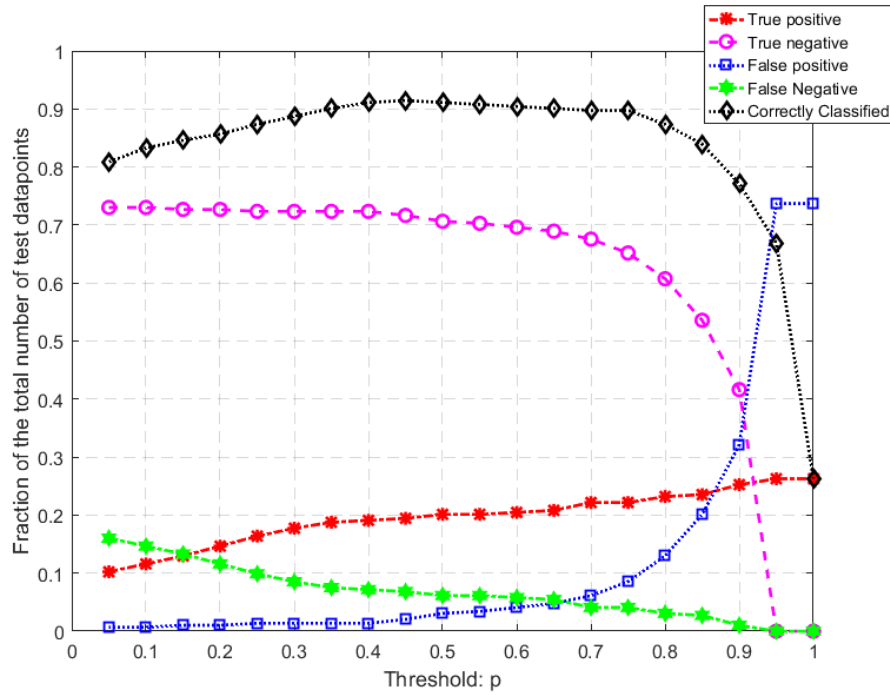


Figure 3.20: Classification accuracy versus threshold p

It can be seen that for $p > 0.35$ and $p < 0.75$ the classification accuracy is about 90 %. As p is increased in this range the number of false negatives decreases and the number of false positives increase. The confusion matrices for $p = 0.35$ and $p = 0.75$ are shown below:

Confusion matrix (p = 0.35)			
Output Class	0	1	
	52 17.7%	4 1.4%	92.9% 7.1%
	25 8.5%	212 72.4%	89.5% 10.5%
	67.5% 32.5%	98.1% 1.9%	90.1% 9.9%
	0	1	Target Class

(a) Confusion matrix: $p = 0.35$

Output Class	0	<div>65 22.2%</div>	<div>18 6.1%</div>	<div>78.3% 21.7%</div>
	1	<div>12 4.1%</div>	<div>198 67.6%</div>	<div>94.3% 5.7%</div>
		<div>84.4% 15.6%</div>	<div>91.7% 8.3%</div>	<div>89.8% 10.2%</div>
		0	1	
		Target Class		

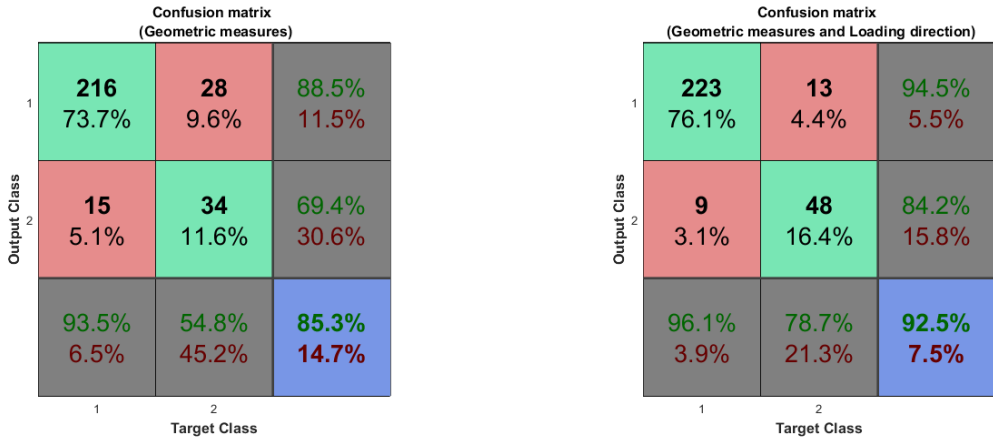
(b) Confusion matrix: $p = 0.75$

Columns correspond to target class. Rows correspond to output class. It can be seen that $p = 0.35$ leads to lower false positives and higher false negatives. Hence, features that are suppressible but are incorrectly classified as unsuppressible is lower (1.4 % of the total number of test data points). Features that are unsuppressible but are incorrectly classified as suppressible is higher (8.5 % of the total number of test data points). $p = 0.75$ leads to higher false positives and lower false negatives. Hence, features that are suppressible but are incorrectly classified as unsuppressible is higher (6.1 % of the total number of test data points). Features that are unsuppressible but are incorrectly classified as suppressible is lower (4.1 % of the total number of test data points). In both cases the overall classification accuracy is close to 90%. The value of the parameter p depends on the relative cost of false positives and false negatives. A value of $p = 0.5$ means that the cost of false positives and false negatives is identical.

3.5 Accuracy of the Classifier

3.5.1 Through Holes

When the above classification model was tested on the through holes dataset, including the direction of loading was found to improve the performance. This can be seen in the figure 3.22 and 3.23. Figure 3.22 shows the confusion matrices and figure 3.23 shows the ROC curve. Each column in a confusion matrix corresponds to a target class. Here class 1 corresponds to suppressible holes and class 2 corresponds to holes that are not suppressible. Each row of the confusion matrix corresponds to an output class.



(a) Confusion matrix:
Classification with geometric features

(b) Confusion matrix: Classification with
geometric features and loading direction

Figure 3.22: Confusion Matrix

We infer that taking into account the direction of load has increased the prediction accuracy on unseen data from 85.3 % to 92.5 %. The above classification was made with the assumption that the cost of false positives and false negatives is identical. The ROC curve enables us to examine the performance of the classifier without this assumption. The area under the ROC curve is a measure of the performance of a classifier. For an ideal classifier this area is equal to 1. In figure 3.23 we see that the area under the curve is 0.96 which is very close to an ideal classifier. Also taking into account the loading direction has led to an increase in the area under the curve.

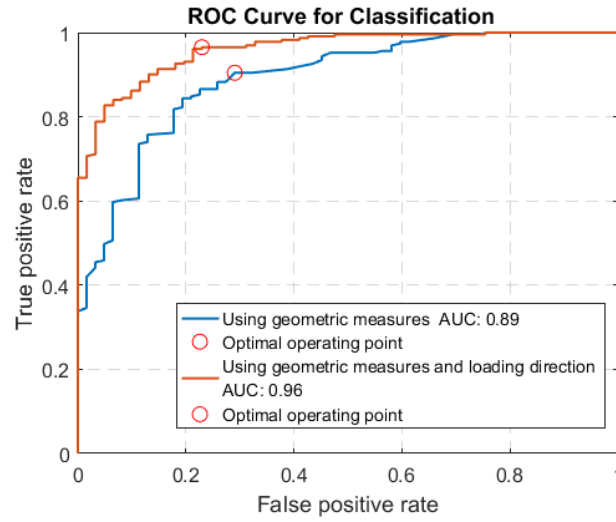


Figure 3.23: ROC curves: Through holes

3.5.2 Blind Holes

When the above classification model was tested on the blind holes dataset, including the direction of loading was found to improve the performance. This can be seen in the figure 3.24 and 3.25. Figure 3.24 shows the confusion matrices and figure 3.25 shows the ROC curve. As with through holes, the overall classification accuracy on the test set has improved from 83.8 % to 93.1% when the loading direction was included in the predictors. The area under the ROC curve has also improved from 0.88 to 0.94

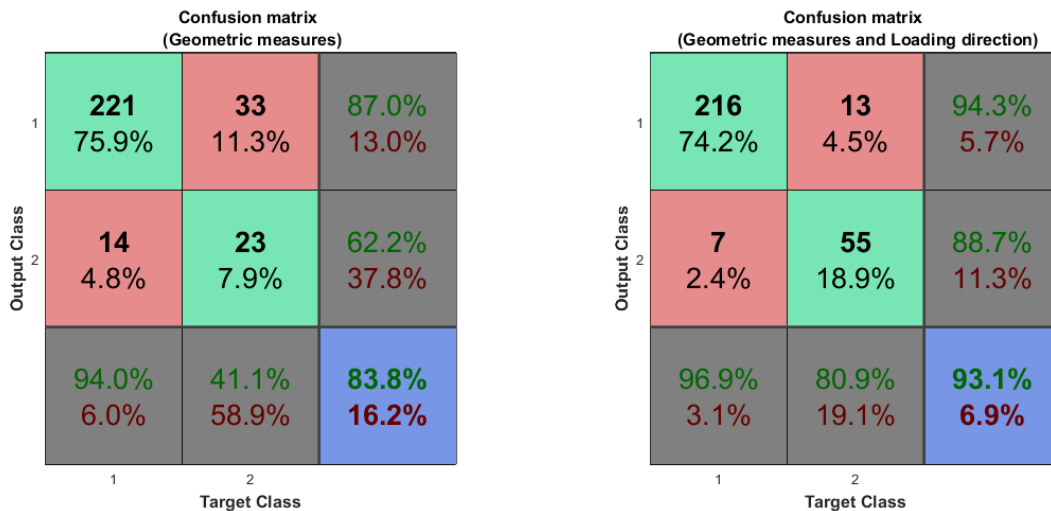
(a) Confusion matrix:
Using geometric measures(b) Confusion matrix: Including loading
direction

Figure 3.24: Confusion matrix

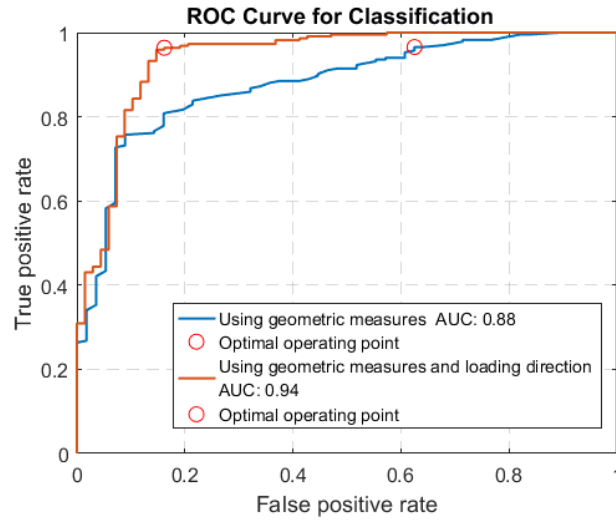


Figure 3.25: ROC curve: Blind holes

3.5.3 Square Holes

When the above classification model was tested on the square pockets dataset, including the direction of loading was found to improve the performance. The performance can be seen in the figure 3.26 and 3.27. Figure 3.26 shows the confusion matrices and figure 3.27 shows the ROC curve.

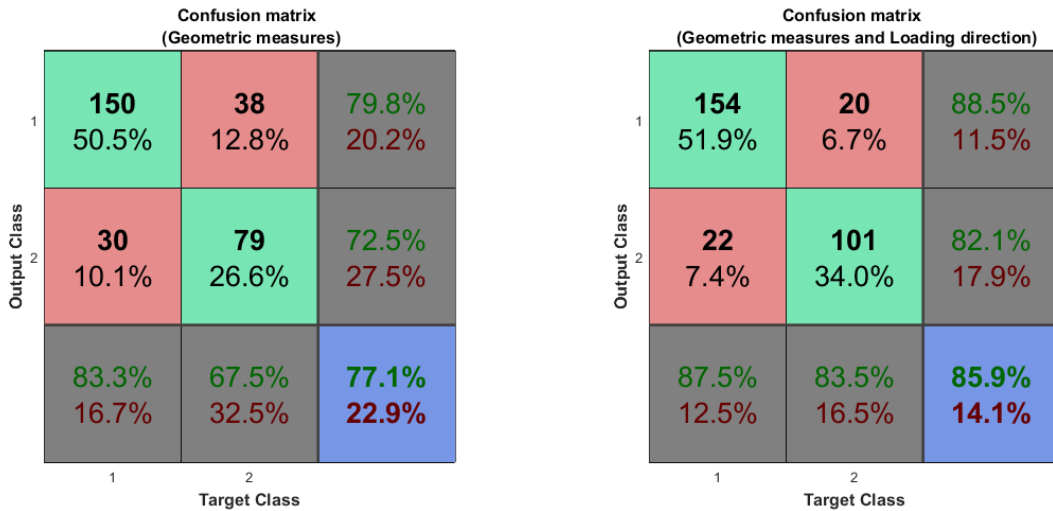
(a) Confusion matrix:
Using geometrical measures(b) Confusion matrix: Including loading
direction

Figure 3.26: Confusion matrix

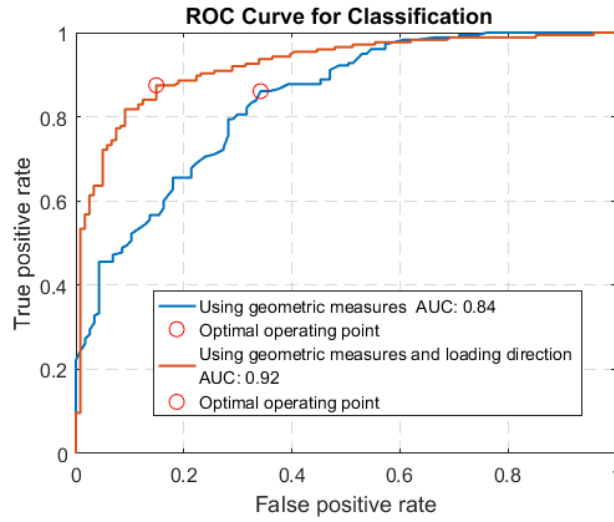


Figure 3.27: ROC Curve: Square holes

We see that the overall classification accuracy has improved from 77.1 % to 85.9 % by the inclusion of information about the loading direction. The area under the ROC curve has increased from 0.84 to 0.92.

3.5.4 Blind and Through Holes Combined

In this experiment, the data for blind and through holes was combined and processed by the classifier. There was no additional information about whether a datapoint corresponds to a through hole or a blind hole. The same set of predictors mentioned in table 3.5 was used.

The performance can be seen in the figure 3.28 and 3.29. Figure 3.28 shows the confusion matrices and figure 3.29 shows the ROC curve.

The classification accuracy is found to be comparable to that found when the classification model was used on the through holes data and blind holes data alone. It was 92.5 % for through holes and 93.1 % for blind holes when trained individually, when trained and tested together it was observed to be 92 %. As in the previous cases, an improvement in the classifier performance is observed by the inclusion of information about loading direction.

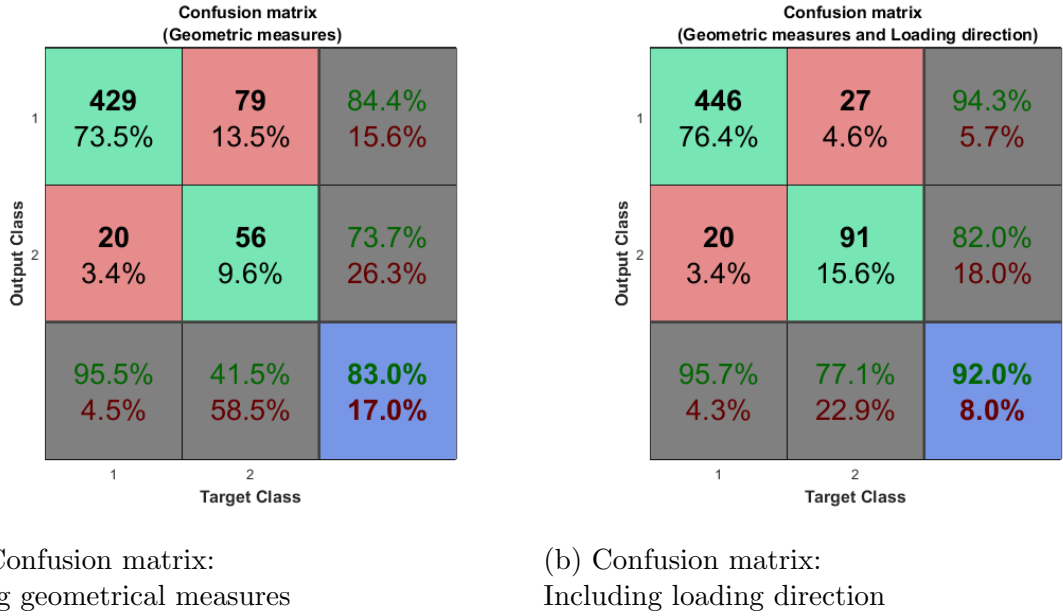


Figure 3.28: Confusion matrices

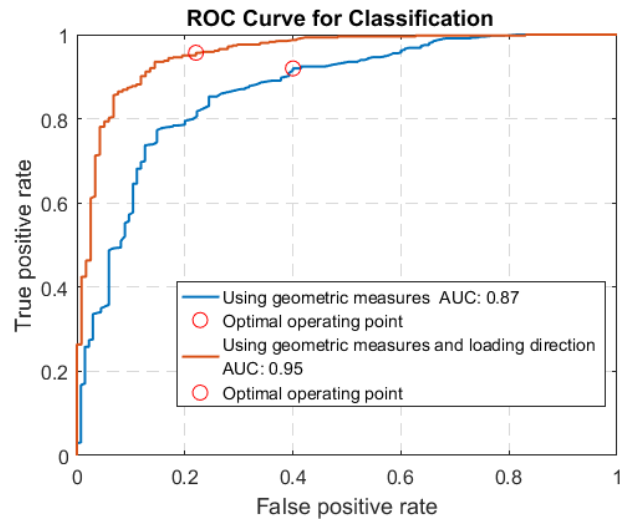
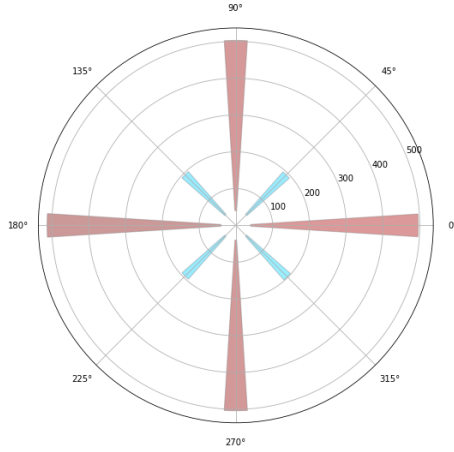


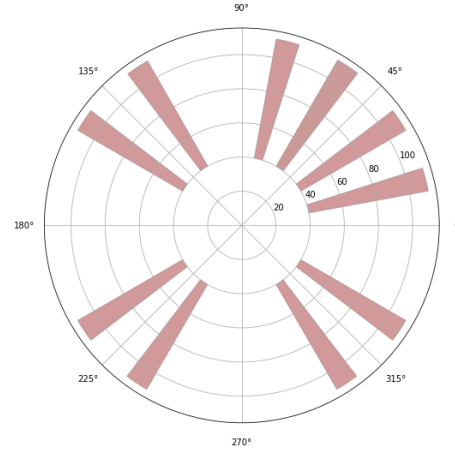
Figure 3.29: ROC Curve: Blind and Through holes data

3.5.5 Generalization

A classifier tries to minimize the classification error on the training dataset, but the usefulness of the classifier depends on how well it generalizes on examples beyond the training dataset. In this experiment the direction of the load in the training data was different from the direction of load in the testing data Figure 3.30.



(a) Direction of load in training data



(b) Direction of load in testing data

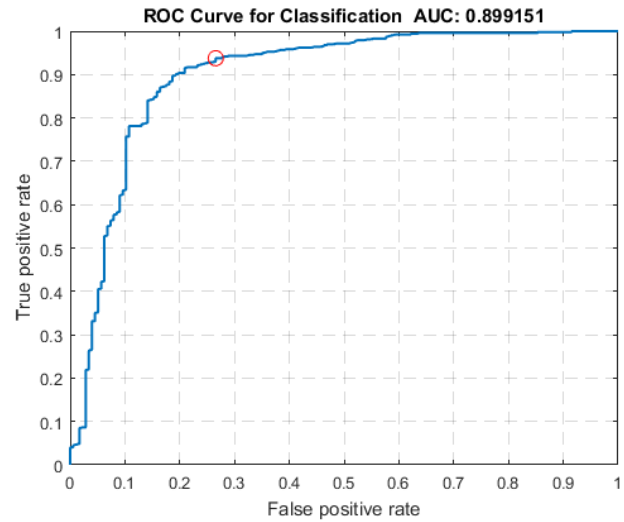
Figure 3.30: Direction of load in training and testing data

The prediction accuracy is summarized in figures 3.31 and 3.32:

Confusion Matrix

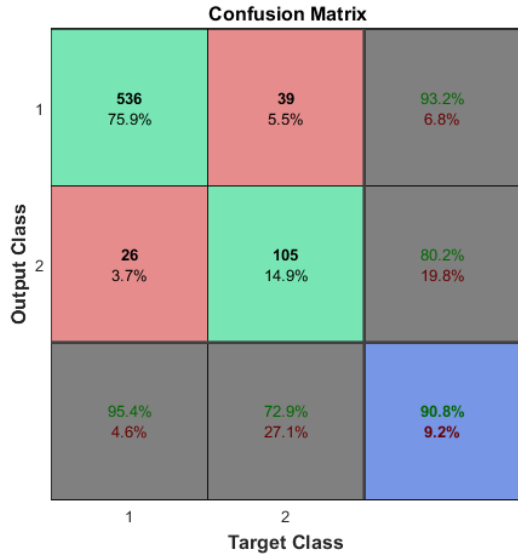
	1	2	
1	501 70.8%	51 7.2%	90.8% 9.2%
2	30 4.2%	126 17.8%	80.8% 19.2%
	1	2	
	94.4% 5.6%	71.2% 28.8%	88.6% 11.4%
	1	2	
	Target Class		

(a) Confusion Matrix

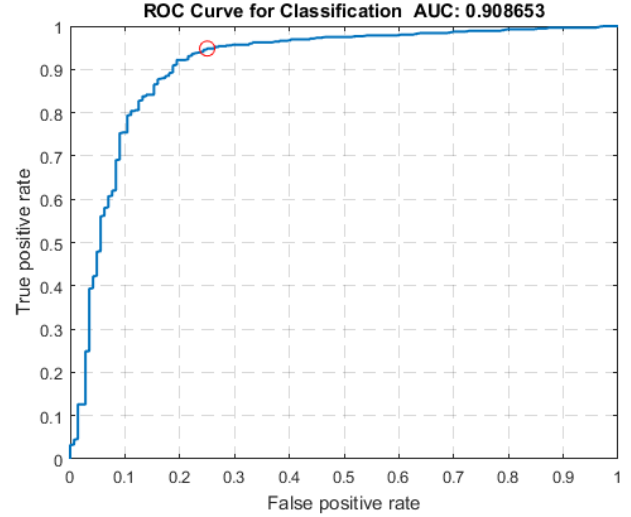


(b) ROC curve

Figure 3.31: Classification performance on through holes



(a) Confusion Matrix



(b) ROC Curve

Figure 3.32: Classification performance on blind holes

It can be seen that the performance in figure 3.31 and figure 3.32 is comparable to the performance mentioned in sections 3.5.1 and 3.5.2 when there was an overlap in the direction of loading in the training and testing data. The overall classification accuracy for through holes is 88.6 % compared to 92.5 % in section 3.5.1. The area under the curve is 0.89 as opposed to 0.96 earlier. The overall classification accuracy for blind holes is 90.8 % compared to 93.1 % in section 3.5.2. The area under the curve is 0.908 as opposed to 0.94 earlier. Hence, the model shows good generalization with respect to the direction of loading.

The next chapter discusses some case studies where the classification model was used to make defeaturing decisions on parts.

Chapter 4

Case Studies

4.1 Overview

In this chapter, the performance of the proposed prototype system predicting the suppressibility of geometric features is demonstrated through case studies. The system takes as input certain attributes about a geometric feature and the geometry of the parent part and suggests whether the feature should be removed from the 3D model of the part under consideration.

4.2 Case Study 1

The geometry of the part with a pattern of four through holes is shown in figure 4.1. The diameter of holes in the four hole pattern is 4mm. The location parameter $d = 9mm$. The dimensions of the parent bracket are shown in figure 3.2. The prototype system proposed was used to determine if each individual hole in the pattern was suppressible. The 14 predictors mentioned in table 3.5 describing each hole in the pattern were processed by the neural network one by one. These were part of the test set. The neural network assigned the class "suppressible" to each hole.

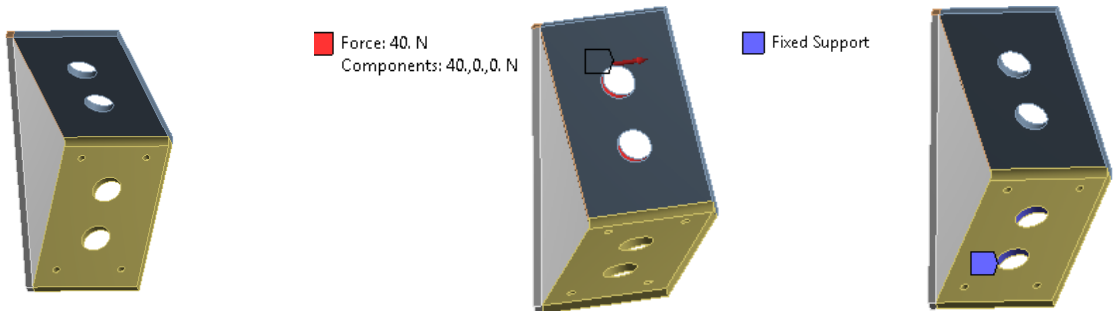


Figure 4.1: Case Study 1: Geometry and boundary conditions

Figure 4.2 shows the von Mises stress in the original model. Figure 4.3 shows the von Mises stress in the defeatured model when all holes in the four hole pattern were suppressed .

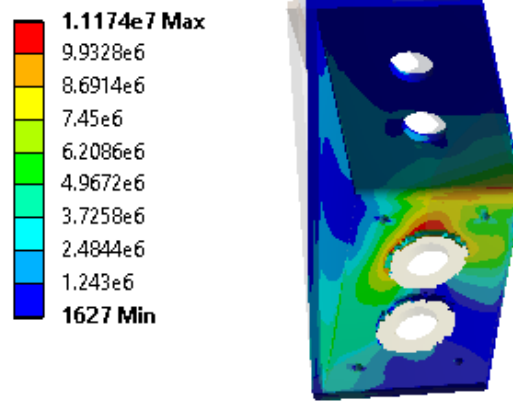


Figure 4.2: Case Study 1: Stress distribution in the original model

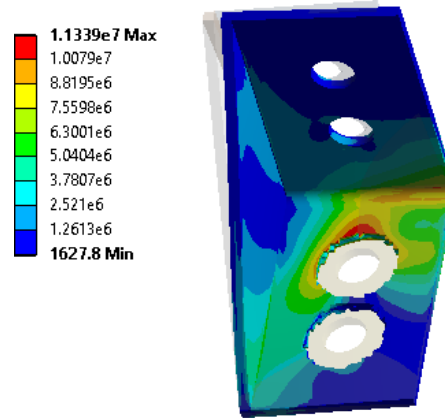


Figure 4.3: Case Study 1: Stress distribution in the defeatured model

Table 4.1 compares the FEA simulation details for the original and defeatured models.

	Number of elements	Average Element Quality	Maximum von Mises Stress in the Model
Original Model	24,676	0.687	$1.1174 \times 10^7 Pa$
Defeatured Model	23,659	0.702	$1.1339 \times 10^7 Pa$

Table 4.1: Comparing original and defeatured Model

It can be seen that there is a 4.1 % reduction in the number of mesh elements, 2.1 %

improvement in the average mesh quality without any significant change in the maximum von Mises stress (1.43 %).

4.3 Case Study 2

The geometry of the part with a pattern of 4 blind holes is shown in figure 4.4. The diameter of holes in the four hole pattern is 12mm. The location parameter $d = 10mm$. The dimensions of the parent bracket are shown in figure 3.2. The prototype system proposed was used to determine if each individual hole in the pattern was suppressible. The 14 predictors mentioned in table 3.5 describing each hole in the pattern were processed by the neural network one by one. These were part of the test set. The classification model predicted that two of the four holes were suppressible -figure 4.5.

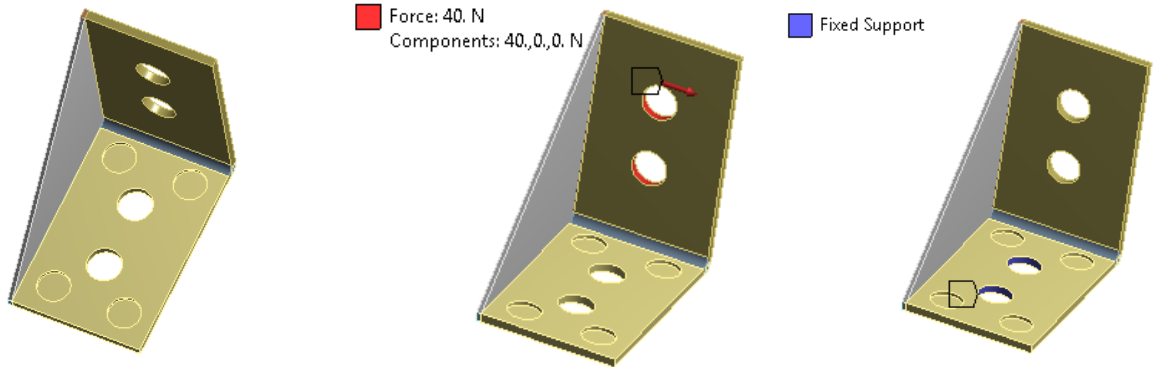


Figure 4.4: Case Study 2: Geometry and boundary conditions

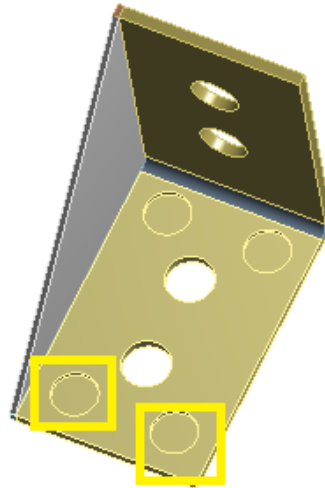


Figure 4.5: Case Study 2: Features marked as suppressible

Figure 4.6 shows the von Mises stress in the original model. Figure 4.7 shows the von Mises stress in the defeatured model when the features marked suppressible are removed.

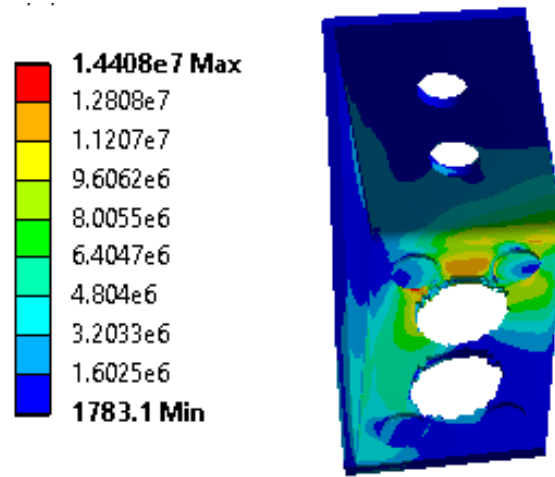


Figure 4.6: Case Study 2: Stress distribution in the original model

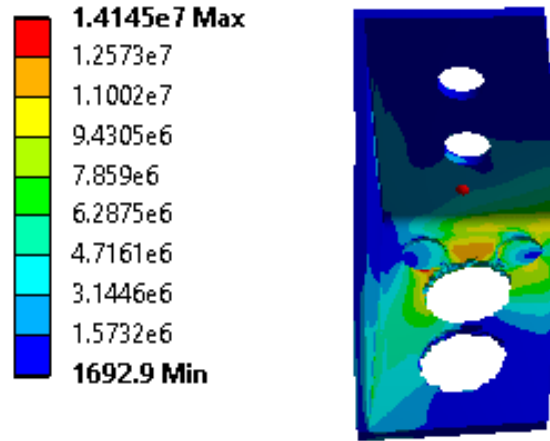


Figure 4.7: Case Study 2: Stress distribution in the defeatured model

Table 4.2 compares the FEA simulation details for the original and defeatured models.

	Number of elements	Average Element Quality	Maximum von Mises Stress in the Model
Original Model	27,113	0.6549	$1.4408 \times 10^7 Pa$
Defeatured Model	25,251	0.6680	$1.4145 \times 10^7 Pa$

Table 4.2: Case Study 2: Comparing original and defeatured Model

It can be seen that there is a 6 % reduction in the number of mesh elements, improvement in the average mesh quality without any significant change in the maximum von

Mises stress (1.8 % change).

4.4 Case Study 3

This case study compares the defeaturing decision when the loading direction is included in the classifier with the decision when the direction of loading is not included in the classifier.

The geometry of the part with a pattern of four through holes is shown in figure 4.8. The dimensions of the parent bracket are shown in figure 3.2. The prototype system proposed was used to determine if each individual hole in the pattern was suppressible. The datapoints corresponding to the holes were in the test set.

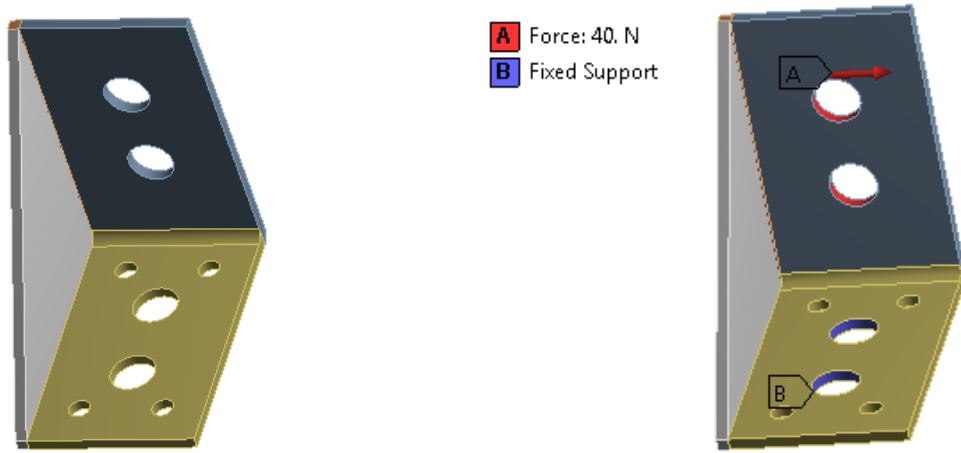


Figure 4.8: Case Study 3: Geometry and boundary conditions

Figure 4.9 shows the von Mises stress in the original model. Figures 4.10 and 4.11 show the von Mises stress in the defeatured model.

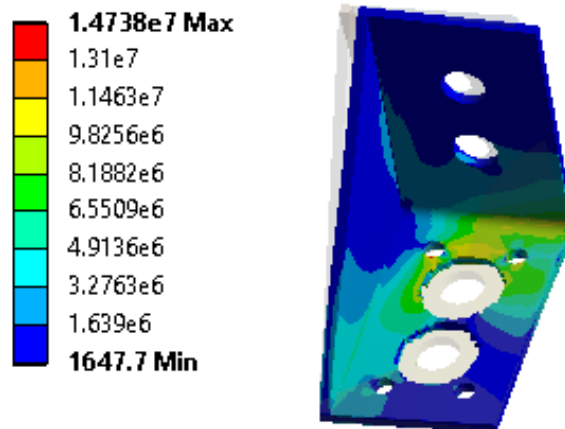


Figure 4.9: Case Study 3: Stress distribution in the original model

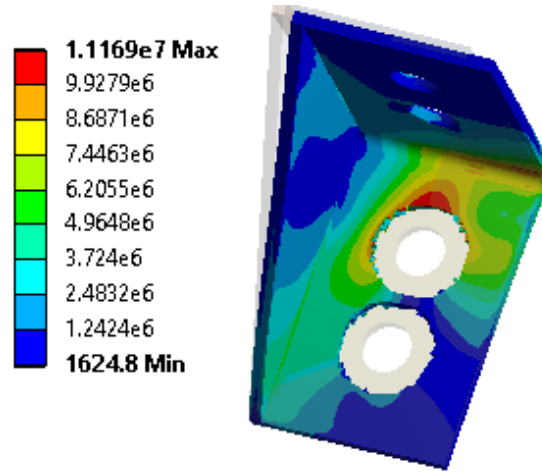


Figure 4.10: Case Study 3: Stress distribution in the defeatured model (Classifier does not include loading direction information)

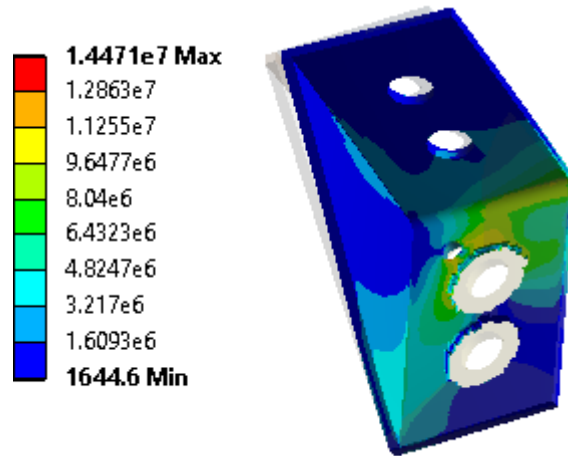


Figure 4.11: Case Study 3: Stress distribution in the defeatured model (Classifier includes loading direction information)

Table 4.3 compares the FEA simulation details for the original and defeatured models.

	Number of mesh elements	Average Element Quality	Maximum von Mises Stress in the Model
Original Model	23,954	0.680	$1.4738 \times 10^7 Pa$
Defeatured Model (Classifier includes forces)	23,877	0.687	$1.4471 \times 10^7 Pa$
Defeatured Model (Classifier does not include forces)	23,238	0.706	$1.1169 \times 10^7 Pa$

Table 4.3: Case Study 3: Analysis of the effect of including forces in the classifier

It can be seen that the decision by the classifier made when the information about

loading direction was included is acceptable. The difference in the maximum von Mises stress in the part is 1.8 %, the number of mesh elements have also decreased. The decision made by the classifier without including the loading direction is not acceptable. The difference in the maximum von Mises stress in the part is 28 %. This value is misleadingly low and might lead to lower factor of safety than required.

The case studies illustrated in this chapter show the advantages of using the proposed classification model. The number of mesh elements in the defeatured models was lower than the original model. The average mesh quality was higher. The importance of including the direction of loading in the classification model was shown in case study 3.

Fail Cases

It was observed that for a significant fraction of the misclassified datapoints in the test phase, the value of y_0 was very close to the threshold of 1.05

$$y_0 = \frac{\text{maximum stress with feature}}{\text{maximum stress without feature}} \quad (4.1)$$

$$y = \begin{cases} 1, & \text{if } y_0 \leq 1.05 \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

where $y = 1$ means the geometric feature is suppressible. These type of mis classifications are not expected to affect the FEA results significantly. No other trend was identified in the fail cases.

Other important conclusions from this study are discussed in the next chapter.

Chapter 5

Summary and Future Directions

5.1 Overview

The objective of the current study was to use machine learning and build a prototype system to identify suppressible holes / pockets in brackets subjected to FEA simulations for static structural analysis. The performance of any machine learning algorithm depends on the quality of the dataset[2]. A comprehensive dataset was generated through extensive FEA simulations on brackets. The dataset consisted of brackets with holes (blind and through) and square pockets of various sizes. It also consisted of variations in the location of these geometric features on the bracket and the direction of loading.

The data from the FEA simulations was transformed to a form that can be processed by a classifier. This step is called feature engineering and involves using domain knowledge of the data to determine predictors that will help in classification. The predictors used in most studies are purely geometric measures. This study analyzed the effect of including the direction of loading in the set of predictors. The performance of the classifier was tested through case studies. Important conclusions from this work are summarized in the next section.

5.2 Conclusions

- CAD models often have small geometric features like holes / pockets which have little effect on the stress distribution in the model but substantially increase the size of the mesh. Large meshes are undesirable as they increase the solution time in a finite element study and are more memory intensive. A prototype system employing neural networks was proposed in this work to automatically predict which geometric features are suppressible one at a time. The results of the system are prototypical in nature and hence restricted in their prediction and domain specific.
- Most of the existing literature to automate the defeaturing process focuses on using purely geometric measures like the dimensions of the geometric feature with respect to the dimensions of the part, the distance of the geometric feature from the displacement constraints and point of application of load to decide whether a geometric feature is suppressible. This study demonstrates that the direction of loading has a significant effect in determining which geometric features are suppressible. Including the direction of loading among the predictors of the classification model was shown to improve the accuracy of the classifier.
- A classifier tries to minimize the classification error on the training dataset, but the usefulness of the classifier depends on how well it generalizes on examples beyond the training dataset. To test the generalization performance, the classifier was trained on a dataset in which the loading was along eight uniformly spaced directions in a plane. The classifier was then tested on examples in which the direction of loading was different from that in the training set. The accuracy of the classifier was 88.6 % on the through holes data and 90.8 % on the blind holes data. This is comparable to the performance achieved when both the training and the testing dataset had loading in the same direction as reported in Sections 3.5.1 and 3.5.2. The accuracy of the classifier was 92.5 % on the through holes data and 93.1 % on the blind holes data.

5.3 Future Work

While this study successfully demonstrates a classification model to predict suppressible holes / pockets in a CAD model family it can be improved further as follows:

- The geometric features considered in this study are limited to holes (blind and through) and square pockets. Features like fillets, chamfers, protrusion and depression have not been studied. Also the geometry was restricted to brackets. More parts can be added to the dataset.
- The current study proposes a prototype in which information about the geometry of the part and feature, direction of loading and the boundary conditions were extracted and pre-processed by the user before being processed by the classifier to determine the suppressibility of geometric features. It can be expanded to extract information about the geometry automatically from standard CAD file formats.
- This study analyzed the effect of suppressing features individually and did not take into account the relationship of the geometric feature under consideration with other geometric features that may be suppressible. Future work can analyze how the decision to suppress one feature depends on the decision to suppress another feature.

Bibliography

- [1] Florence Danglade, Jean-Philippe Pernot, and Philippe Véron. “On the use of machine learning to defeature CAD models for simulation”. In: *Computer-Aided Design and Applications* 11.3 (2014), pp. 358–368.
- [2] Pedro Domingos. “A few useful things to know about machine learning”. In: *Communications of the ACM* 55.10 (2012), pp. 78–87.
- [3] Jihad El-Sana and Amitabh Varshney. “Topology simplification for polygonal virtual environments”. In: *IEEE Transactions on Visualization and Computer Graphics* 4.2 (1998), pp. 133–144.
- [4] Lionel Fine, L Remondini, and J-C Leon. “Automated generation of FEA models through idealization operators”. In: *International Journal for Numerical Methods in Engineering* 49.1-2 (2000), pp. 83–108.
- [5] Nikhil Joshi and Debasish Dutta. “Feature simplification techniques for freeform surface models”. In: *Journal of Computing and Information Science in Engineering* 3.3 (2003), pp. 177–186.
- [6] Jae Yeol Lee, Joo-Haeng Lee, and Hyung-Sun Kim. “A cellular topology-based approach to generating progressive solid models from feature-centric models”. In: *Computer-Aided Design* 36.3 (2004), pp. 217 –229. ISSN: 0010-4485. DOI: [http://dx.doi.org/10.1016/S0010-4485\(03\)00094-0](http://dx.doi.org/10.1016/S0010-4485(03)00094-0). URL: <http://www.sciencedirect.com/science/article/pii/S0010448503000940>.
- [7] Yong-Gu Lee and Kunwoo Lee. “Geometric detail suppression by the Fourier transform”. In: *Computer-Aided Design* 30.9 (1998), pp. 677–693.
- [8] William Roshan Quadros and Steven J Owen. “Defeaturing CAD models using a geometry-based size field and facet-based reduction operators”. In: *Engineering with Computers* 28.3 (2012), pp. 211–224.
- [9] Brian H Russ. *Development of a CAD model simplification framework for finite element analysis*. Tech. rep. DTIC Document, 2012.

- [10] Alla Sheffer. “Model simplification for meshing using face clustering”. In: *Computer-Aided Design* 33.13 (2001), pp. 925–934.
- [11] “Stanford University CS231n: Convolutional Neural Networks for Visual Recognition by Andrej Karpathy”. In: ().
- [12] Atul Thakur, Ashis Gopal Banerjee, and Satyandra K. Gupta. “A Survey of CAD Model Simplification Techniques for Physics-based Simulation Applications”. In: *Comput. Aided Des.* 41.2 (Feb. 2009), pp. 65–80. ISSN: 0010-4485. DOI: [10.1016/j.cad.2008.11.009](https://doi.org/10.1016/j.cad.2008.11.009). URL: <http://dx.doi.org/10.1016/j.cad.2008.11.009>.