Data Driven Design of Load Bearing Components

A report Submitted
In Partial Fulfilment of the Requirements
for the Degree of
Bachelors of Technology

by

Atharva Malwadkar (180103015)
Devanuj Baruah (180103023)

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
May 2022

CERTIFICATE

It is certified that the work contained in the report entitled "***Data Driven Design of Load Bearing Components***", by "*Mr. Atharva Malwadkar*" and "*Mr. Devanuj Baruah*" , has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Prof. Debabrata Chakraborty

Prof. Shyamanta Moni Hazarika

April, 2022

Department of Mechanical Engineering,

I.I.T Guwahati

Declaration

We declare that this written submission represents are our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be 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 not been taken when needed.

Atharva Malwadkar

Roll number: 180103015

Devanuj Baruah

Roll number: 180103023

Department of Mechanical Engineering,

I.I.T Guwahati, Assam, India

Dedicated to

Our Parents

and to
Teachers and Friends

Abstract

Mechanical design focuses primarily on the design of load bearing components with the aim of determining the best size, shape, and material to ensure the component's protection under load. The constitutive models which form the basis of mechanical properties used for traditional design problems, have their own limitations. These models are complex and are slow to process when dealing with complex problems for systems with high dimensional design variables. We propose the Data-Driven design as a way to eliminate epistemic uncertainties linked to traditional constitutive models in mechanical design. By eschewing empirical models, data driven design can eliminate modelling error and uncertainty, and no loss of experimental information is incurred. Data-driven model-free approach can not only eradicate the explicit postulation of a particular constitutive model, but it can also produce databases of material states that follow some relevant sampling guidelines. In the scope of our thesis project, We shall implement and discuss cost reduction methods using euclidean norms, such as linear regression as a baseline model, decision trees, boosting techniques with ensemble models and self organizing maps. We will analyse the results and infer our conclusions on the efficiency or the lack thereof. The methodology, results and discussions can be found in the following chapters.

Atharva Malwadkar and Devanuj Baruah

IIT Guwahati

May, 2022

Acknowledgment

We would like to express our sincere gratitude towards our supervisors, Prof. Debabrata Chakraborty and Prof. Shyamanta Moni Hazarika, and thank them for the most valuable guidance and supervision over the course of this project. From introducing us to the field of Data Driven Design to their exemplary technical advice, persistent monitoring and constant encouragement throughout the project, we are deeply indebted to them for giving us this opportunity.

Atharva Malwadkar
Devanuj Baruah
IIT Guwahati
April, 2022

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In engineering, design refers to the method or process of solving problems. A proposed design offers solutions to an issue that leads to the creation of a product or process. Engineering design is the method of using our knowledge of mathematics and natural sciences to devise a device, part, or process to meet desired needs.

Mechanical design focuses primarily on the design of load bearing components with the aim of determining the best size, shape, and material to ensure the component's protection under load. As a result, a mechanical design procedure has often been carried out using relevant constitutive models in the past. These constitutive models have historically been important for structural design and material certification, and have been accomplished primarily through uni-axial tests aimed at determining elastic material properties.

1.2 Limitations of constitutive models

The aforementioned constitutive models which form the basis of mechanical properties used for traditional design problems, have their own limitations. These models are complex and slow to process when dealing with complex problems for systems with high dimensional design variables. In certain cases, the design domain is much too large for a constitutive model to handle.

The nonexistence of knowledge of a design problem also makes getting to the correct constitutive model almost impossible. Constitutive models bring with them material modelling empiricism, modelling error and uncertainty. This modeling error and uncertainty arise from imperfect knowledge of the functional form of the material laws, the phase space in which they are defined, and from scatter and noise in the experimental data. Furthermore, often the models used to fit the data are ad hoc, without a clear basis in physics or a mathematical criterion for their selection, and thus the process of modeling is mired in empiricism and arbitrariness. The entire process of empirical material modeling, and model validation thereof, is open-ended and no rigorous mathematical theory exists to date that makes it precise and quantitative.

1.3 Data Driven Design

Previous work in using large unstructured data-sets to infer solutions for boundary value problems have been categorized under *Data Science*, but typically with the aim of parametric identification, or augmenting and automating, rather than replacing, the use and generation of material models. Material informatics uses database techniques to first identify parameters of correlation and then use machine-learning regression techniques to ultimately provide predictive quantitative models. Principal-component analysis provides methods of dimensional reduction that allow such modeling techniques to be applied. **Fig 1.1** shows generalised flow for data driven engineering includes the dynamic flow of data to and from different process points, thereby modelling the problem around statistical manipulation and machine learning techniques with relevant data. This can be used with or without the postulation of empirical models, to aid in the accuracy or results inferred from analytical methods.

In more recent research, Data-Driven Design is proposed as a way to remove the epistemic uncertainties associated with conventional constitutive models. Data driven design can minimise modelling error and uncertainty by avoiding empirical models, and there is no loss of experimental knowledge. A data-driven model-free approach can not only eradicate the explicit postulation of a particular constitutive model, but it can also produce databases of material states that follow some relevant sampling guidelines. Further, for orthosis and prostheses, the designs that are data –driven are more

engaging and tailored to users' preferences. In **Fig 1.2**, we can see how data driven design operates where a large number of relevant data is collected and processed to be used as the model (guiding the behavior of the component instead of constitutive model) for design, instead of using data just a means for automating the analytical processes of already established design methodology using constitutive models.

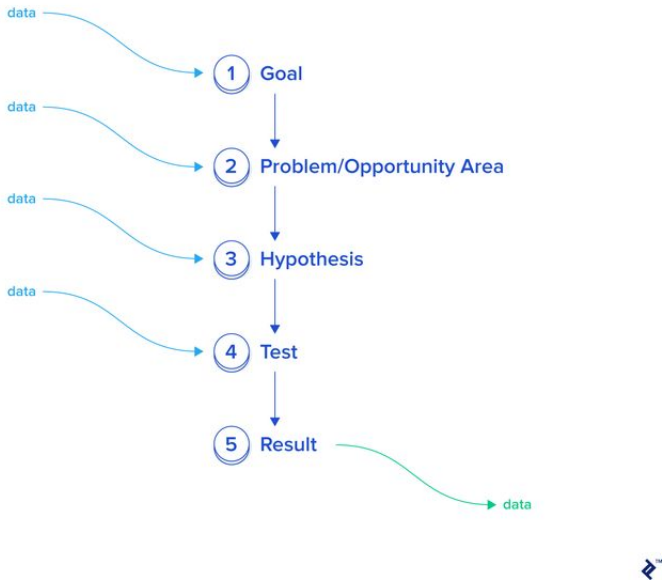

Figure 1.1: Data Driven Design process flow [3]

1.4 Objectives

The aforementioned papers shed more light into the conditions and criterion under which a model-free data driven design methodology would be more preferred over traditional modeling using the constitutive relations. In our report, we shall discuss one such methodology where we generate validated synthetic datasets using ANSYS, which conform to the correct set of constraint space and equilibrium condition. We further implement and discuss Machine Learning algorithms such as Decision Tree, stacking techniques of ensemble models and Self-Organizing Maps. We will analyse the results and infer our conclusions on the efficiency or the lack thereof. The methodology, results and discussions can be found in the following chapters.

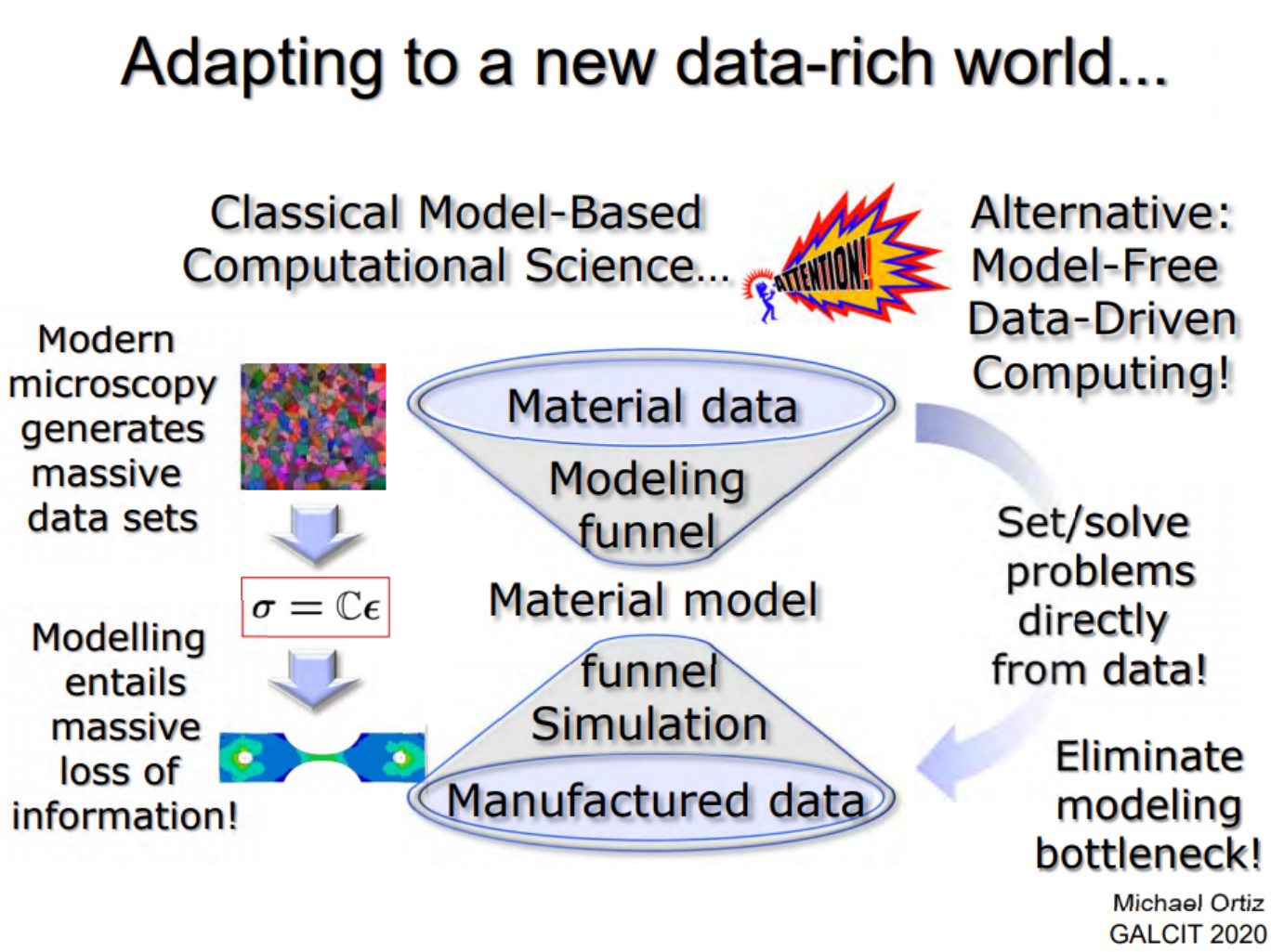

Figure 1.2: Use of Model-Free design practices using material data[4]

Chapter 2

Literature Review

Stainier et al [1] propose an integrated model-free data-driven approach to solid mechanics and show how principles and techniques of data driven computational mechanics leads to elimination of epistemic uncertainty linked to traditional constitutive models. Qualitative as well as quantitative results obtained using a data-driven approach were more accurate. The methodology proposed in the paper is to generate a material database using the Data-Driven Identification method, which is used to solve the classical boundary value problem. This is done by employing a distance minimization algorithm to find a compatible mechanical state (stress-strain field) in equilibrium, broadly termed as Data-Driven Computational Mechanics (DDCM). Acquired data is used to train a model to predict responses from different geometric structures of the same material. This method thus eliminates the postulation of a constitutive model entirely, thus getting rid of its uncertainties.

The paper follows this procedure and applies the DDI+DDCM methodology along with computational techniques such as FEM, to generate a robust procedure for response prediction. They work on two structural examples of the same material and document the performance of the constitutive models v.s the model-free technique, noting the errors as well as deviance and convergence of error distances with the difference in data points. The paper concludes to note that the performance of the data-driven technique is noteworthy for linear systems corresponding to standard elasticity problems, and would work just as fine for material non-linearity in the data-set, as our linear system of equations in the DDI+DDCM algorithms remain unaffected. We shall explore this paper further in detail throughout our report.

Ponmalai et al. [5] discuss and evaluate the working performance of Self-Organizing Maps (Kohonen Maps) as a form of neural network that are used for function approximation methods, visualization and exploratory data analysis of high dimensional dataset. They also demonstrated that the Self-Organizing Map is a time efficient method for data sets with both high numbers of samples and also high dimensionality. A SOM is trained by updating the locations of the weight vectors. The goal of this update step is to move the weight vectors to be representatives of the centers of clusters in the data and to mirror the order of the lattice. This is very much in line with the scope of our endeavours in this thesis project.

Shin et al. [2] presents a data-driven approach for developing a one-dimensional thin-walled beam model. In order to analyze complicated deformations occurring in a thin-walled beam by a beam theory, it is important to identify core cross-sectional deformations that are the bases for the one-dimensional beam analysis. For data processing, they have used principal component analysis to obtain the desired core cross-sectional deformations without making any assumptions about the behaviour of the beam's sectional deformations. The core cross-sectional deformations can then be expressed in explicit functional forms (shape functions) to make the one-dimensional higher-order beam analysis construction easier. This paper discusses in detail the formulation of a robust Data-Driven pipeline to solve for design problems, and also the various levels and stages of processing raw material data for optimised use-cases. These practices were put to use by comparing numerical results with the processed shape functions obtained earlier, proving to show accurate results for static problems as well as vibration and buckling analyses. They have gone on to note the importance of correct procedure for the generation of data and further preprocessing. They have also stated that while their results have been obtained exclusively on tests for linear isotropic materials, they should be effective while dealing with composite materials as well.

Kirchdoerfer et al. [3] developed a new computing paradigm known as data-driven computing, in which calculations are performed directly from experimental material data as well as relevant constraints and conservation laws, such as compatibility and equilibrium, bypassing the empirical material modelling phase of traditional computing entirely. Data-driven solvers try to allocate the state from a predefined data set that comes closest to satisfying the conservation laws to each material point. These

try to find the state that is nearest to the data set while satisfying the conservation laws. The resulting data-driven problem thus consists of the minimization of a distance function to the data set in phase space subject to constraints introduced by the conservation laws. They have checked the performance of this model-free approach by employing two examples- static equilibrium of 3D trusses and finite-element discretized linear-elastic solids. After the formulation of the constraint equations using Lagrange multipliers, they go on to capture the intersection of the constraint set with the global phase-space. With the use of Big Data, this paper discusses the scope of extending these design problems beyond linear isotropic materials, as well as exploring a data-driven approach for tackling dynamic problems since the addition of non-inertial forces will not affect the material behaviour.

Kotsiantis [6] in his paper titled *"Decision Trees: an Overview"* discusses the motivation behind using decision trees for various machine learning purposes. They proceed to discuss the steps to build and deploy a decision tree based model- namely building and inference gathering (training). They begin by initializing an empty decision tree and then selecting for each decision node the correct set of attributes using an attribute selection measure. The selected attribute must maximally diminish the mixture of classes between each subset created. The result inferences are gathered by we start with the root of the constructed tree and follow the path corresponding to the observed value of the attribute in the interior node of the tree. This process is continued until a leaf is encountered.

Chapter 3

Methodology

3.1 Overview

The model-free approach we will look at, known as data-driven computational mechanics (DDCM) [3], involves formulating calculations directly from experimental material data and relevant critical constraints and conservation rules, bypassing the need for empirical material modelling in traditional computing. The data-driven solver aims to allocate the closest possible state from a predefined material-data set to and material point of the computational model while also meeting the necessary constraints and conservation rules. The local state assignment's best fit is calculated using a figure of merit that penalises the distance from the data set in phase space.

The resulting data-driven problem thus consists of the minimization of a distance function to the data set in phase space subject to constraints set forth by the essential constraints and conservation laws. We can also generate reference solutions using Finite-Element Analysis and constitutive models to benchmark the DDCM solver's performance. DDCM has also been documented to be robust in its capacity to perform simulations on various geometries and loadings using a given database. Qualitatively, the method requires only limited data to predict the major features of strain and stress fields (e.g. location of maximal values). Quantitatively, the precision obtained depends directly on the quality of the database (number and sampling of data points). Every dataset that we have generated in the scope of this project satisfies all equilibrium and constraint equations. The data has been generated in ANSYS software, which helps us formulate an initial condition-boundary value problem satisfying all conditions in the constraint space. We generated meshes of various sizes for all the modelling test

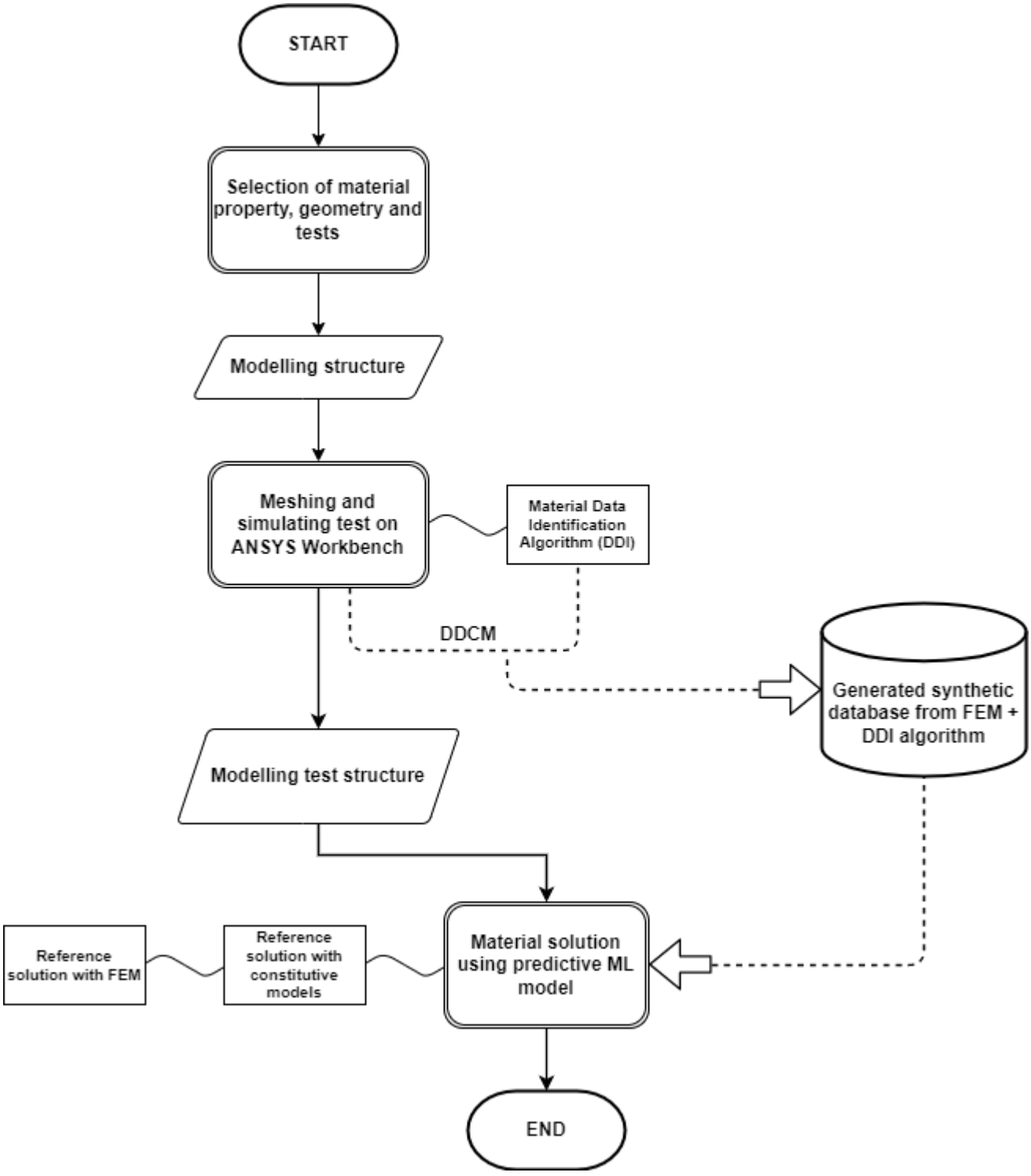

Figure 3.1: Flowchart for the model-free method of solving a given mechanical design problem.

structures and refined them to the point of validation. These set of appropriate finite element models were optimal for validating our predicted results as well as showcasing the non-constitutive behaviour of DDCM.

The Machine Learning formulation of the solution to our design problem relies on understanding the motivation behind the selection of different algorithms and the pipeline architecture. We use a stacking regressor where different function approximation methods (Decision Tree, Self-Organizing Map) are used in tandem with a final regressor function (RandomForestRegressor) to give us accurate material data prediction. We shall look at the methodology now. Considering our design problem, equilibrium constraints, computational power and qualitative knowledge domain at hand, we have formulated a complete pipeline for data-driven simulation methodology, as seen in Fig 3.1

3.2 Learning Algorithms

3.2.0.1 Self-Organizing Maps

A Self Organizing Map is a type of artificial neural network which is trained using competitive learning rather than error-correction learning (e.g., backpropagation with gradient descent) used by other artificial neural networks. It is an unsupervised machine learning technique to produce the low dimensional representation of any high dimensional data while preserving its topological structure. A data set with p variables measured in n observations could be represented as clusters of observations with similar values for the variables. These clusters then could be visualized as a two-dimensional "map" such that observations in proximal clusters have more similar values than observations in distal clusters. This can make high-dimensional data easier to visualize and analyse. This powerful paradigm that is extensively applied for clustering and visualization purpose and for regression learning, due to its ability to provide a topological projection of high dimensional non linear data. In this case, data extracted from the SOM are usually restricted to the best matching unit (BMU). A SOM is trained by updating the locations of the weight vectors. The goal of this update step is to move the weight vectors which are representatives of the centers of clusters in the data and to mirror the order of the lattice. Fig 3.2 demonstrate the training of a SOM on uniform random 2-D data. The weight vectors of a SOM are initialized with random values. As the weight vector of each lattice node moves, its neighbors move with it.. The amount that a neighbor of a node moves is controlled by a function called the neighborhood

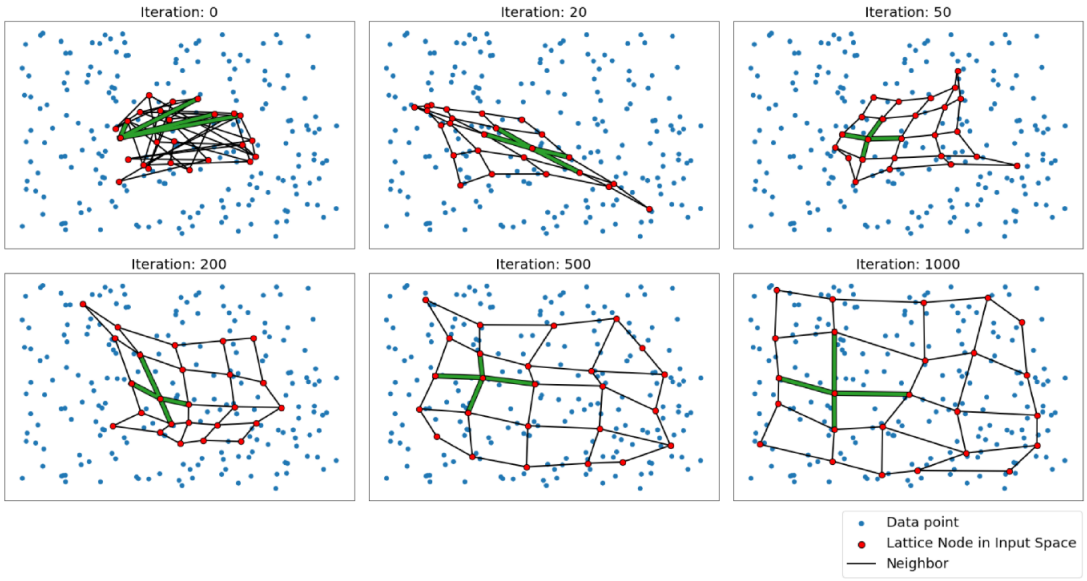

Figure 3.2: Training of SOM on uniform random 2-dimensional data[5]

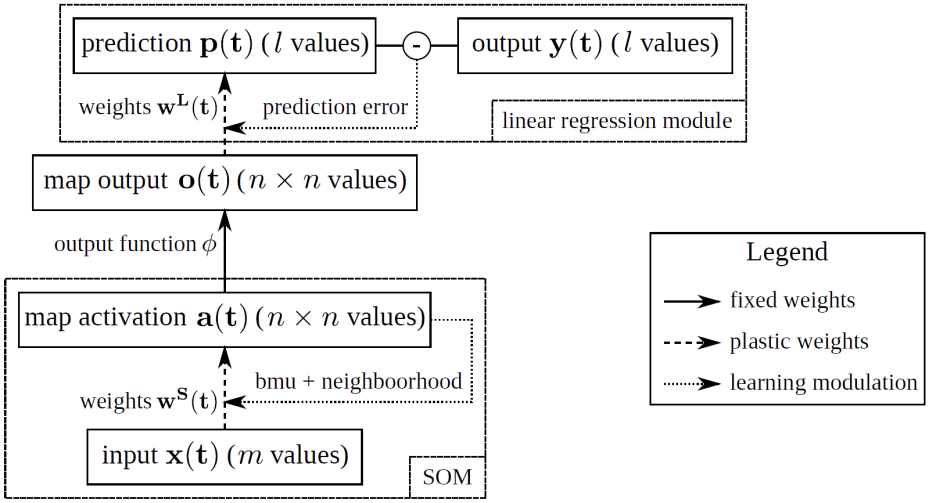

Figure 3.3: SOM architecture with a regression approximator feed [7]

function and it is proportional to the distance of the two nodes in the lattice space.

SOM architecture:

To study the influence of the output activity of the SOM when using it in an integrated model, we consider the architecture depicted in Fig 3.3. This architecture is designed for supervised learning of an input/output relationship. SOM as a regressor is used to easily generalize our data, where the stress-strain field takes up a complex high dimensional relationship for areas with stress concentration.

3.2.1 Decision Tree

A decision tree as a part of Classification and regression trees is a machine-learning method for constructing prediction models from data. The model is obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. As a result, the partitioning can be represented graphically as a decision tree. Decision trees are designed for dependent variables that take a finite number of unordered values, with prediction error measured in terms of misclassification cost. Regression trees are for dependent variables that take continuous or ordered discrete values, with prediction error typically measured by the squared difference between the observed and predicted values. Decision trees have an advantage that it generalizes easily to the data, lesser data cleaning is required, non-linearity does not affect the model's performance, lesser number of hyper-parameters to tune and it is robust to the distribution of the dataset. The tree-structured classifier with three types of nodes:

- The ***Root Node*** is the initial node which represents the entire sample and may get split further into further nodes.
- The ***Interior Nodes*** represent the features of a data set and the branches represent the decision rules.
- Finally, the ***Leaf Nodes*** represent the outcome.

There are two major phases of the Decision Tree induction process: the growth phase and the pruning phase. The growth phase involves a recursive partitioning of the training data resulting in a Decision Tree such that either each leaf node is associated with a single class or further partitioning of the given leaf would result in at least its child nodes being below some specified threshold. The pruning phase aims to generalize the Decision Tree that was generated in the growth phase by generating a sub-tree that avoids over-fitting to the training data. Pruning can be done by restricting the maximum depth(levels) of the tree. Decision tree algorithms need to recursively partition dataset into subsets according to some splitting criteria i.e. they still have to repeatedly compute the records belonging to a node and then compute the splits for the node. In each iteration, the algorithm considers the partition of the training set using the outcome of a discrete function of the input attributes. The selection of the most appropriate function is made according to some splitting measures. After the selection of an appropriate split, each node further subdivides the training set into smaller subsets, until no split gains sufficient splitting measure or a stopping criteria is

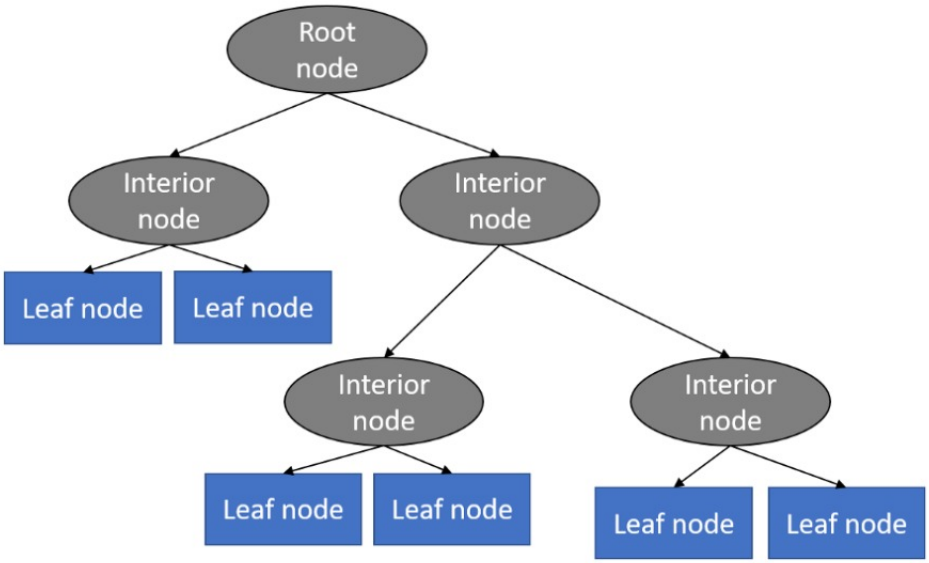

Figure 3.4: General Decision Tree design [6]

satisfied. Several impurity measures can be used to check goodness of the split at the node. For the regressor we have used **gini impurity**.

$$i_c = \sum_i f_i \cdot (1 - f_i) = 1 - \sum_i f_i^2 \tag{3.1}$$

where f_i represents the fraction of records belonging to class i.

3.2.2 Stacking Regressor

Ensemble learning is a machine learning paradigm where multiple models (often called "weak learners") are trained to solve the same problem and combined to get better results. Stacking Regressor, is an ensemble learning technique where the predictions of multiple regressors (referred as level-one regressors) as linearly combined to use as new features to train a meta-classifier. Stacked generalization consists in stacking the output of individual estimator and use a regressor to compute the final prediction. Stacking allows to use the strength of each individual estimator by using their output as input of a final estimator.

We have trained the dataset using Self Organising Map Regressor and Decision Tree Regressor as individual estimators. The predictions from both the estimators are fur-

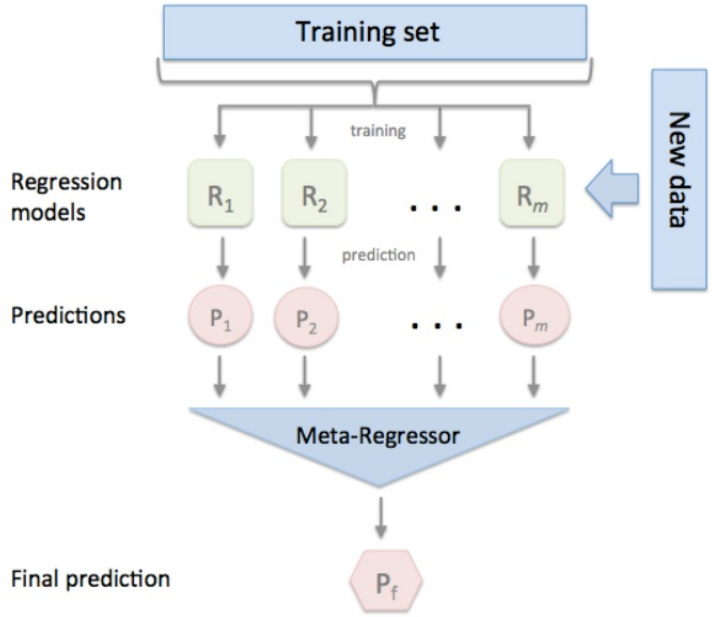

Figure 3.5: Stacking model architecture[8]

ther fed to RandomForest Regressor which acts as meta-regressor to produce the final output for our model. The RandomForest Regressor is an another bagged ensemble technique of pruned Decision Trees. It uses the data bootstrapping and aggregation, to generate varied Decision Trees. The final prediction is the combined mean of all the individual Decision Tree model's output.

Algorithm: Stacking Regression[8]

Input: Training data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{m}$ $(\mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathcal{Y})$ **Output:** An ensemble regressor H

1. Step 1: Learn first-level regressors

   for $t \leftarrow 1$ to T **do:**

   Learn a base regressor h_t based on $\mathcal{D}$

   end for

2. Step 2: Construct new data sets from $\mathcal{D}$

   for $i \leftarrow 1$ to m **do:**

Construct a new data set that contains $\{\mathbf{x}'_i, y_i\}$, where $\mathbf{x}'_i = \{h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), \ldots, h_T(\mathbf{x}_i)\}$

end for

3. Step 3: Learn a second level regressor

   Learn a new regressor h' based on the newly constructed data set

   return $H(\mathbf{x}) = h'(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x}))$

Chapter 4

Result and Discussion

4.1 Data generation and implementation

The material data-set is generated through simulations of load bearing members on computer software, namely ANSYS. For the purpose of this project, we have evaluated our machine learning model for load prediction.

4.1.1 Plate

A finite element simulation was run on a thick plate(E=217.5E+09, Poisson's ratio= 0.3) with dimensions 200 x 128 x 50 mm with a central hole in it. It was uni-axially loaded in the vertical and horizontal plane under four different loading conditions (100N, 120N, 130N, 140N). We have used this generated data-set, whereas you can model this yourself using the DDI algorithm. In an ideal case, we should use all of these synthetic databases to compare our solutions and infer the best results. We have meshed the said plate with 300K+ nodes. from the loading and displacements, stress-strain field was formed. This IVBP problem conformed to the set of constraint and equilibrium relations (Fig 4.1) and gave us the same validated results with increasingly finer meshes. The sampled 2-D stress and strain invariants of these specimen are plotted along with a reference linear solution (Fig 4.2 - 4.5).

We also plot the strain space of these specimen which conform to the DDCM/FEM solutions while being a synthetic dataset. Thus we can check the validity of the behaviour of the generated data (Fig 4.6 - 4.9)

We employ our tandem SOM and Decision Tree model in a stacking regressor.

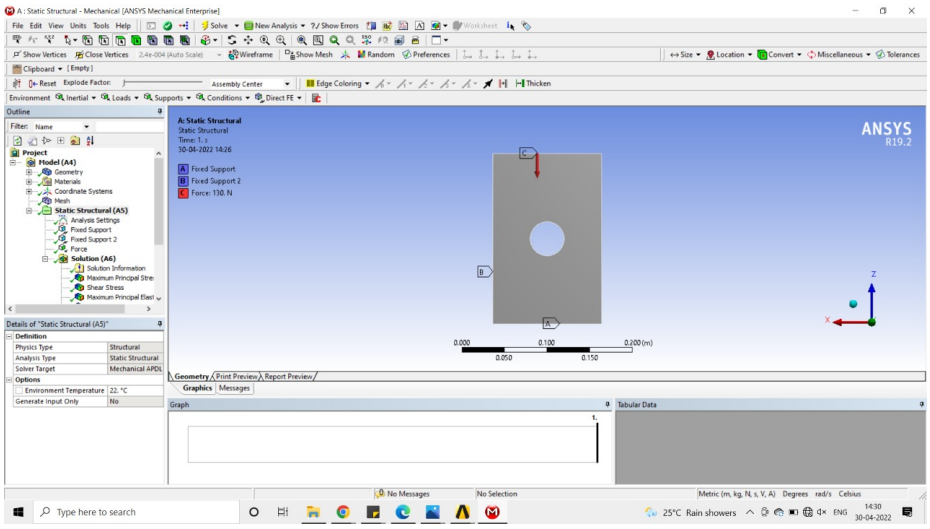

Figure 4.1: Singular hole plate modelling

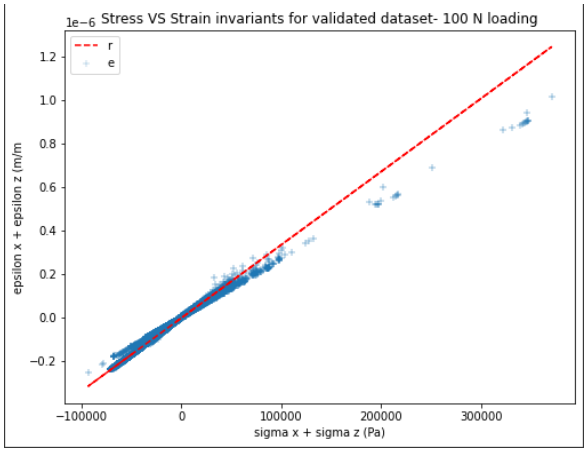

Figure 4.2: stress-strain invariant 100 N

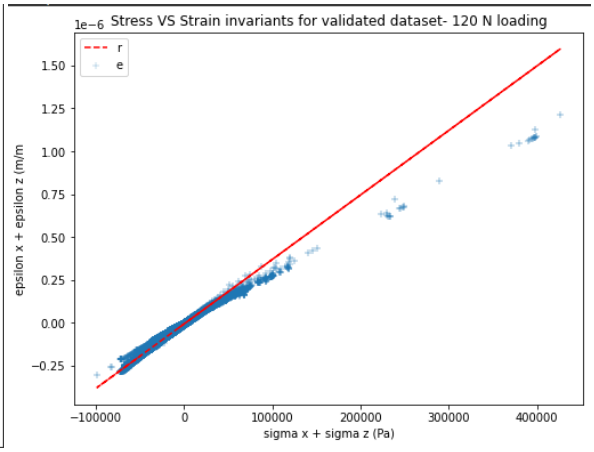

Figure 4.3: stress-strain invariant 120 N

Firstly, the data of three plates- 100N, 120N and 140N are concatenated and then a fraction of it it randomly sampled. This data will be our training dataset for the stacking regressor. We are focused on predicting the Normal Stress (Pa) along the Z axis, which will be our target variable. The rest of the independent variables are- *'Normal Elastic Strain (m/m) (X)','Normal Elastic Strain (m/m) (Y)', 'Normal Elastic Strain (m/m) (Z)', 'Shear Elastic Strain (m/m) (XY)', 'Shear Elastic Strain (m/m) (XZ)','Shear Elastic Strain (m/m) (YZ)'*. We use these same features as dependent and independent variables in future experiments as well. The data is standard scaled for the pipeline along with our test data (plate load 130 N). It is subsequently fed onto the stacking regressor where it acts as a raw input to the Decision Tree, and then into the SOM regressor. A final function estimator is used to reduce the variance in the form of a RandomForest regressor. The hyperparameters for the Decision Tree is as follows-

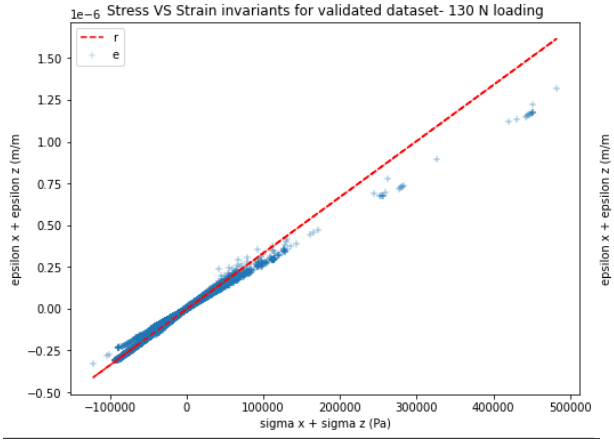

Figure 4.4: stress-strain invariant 130 N

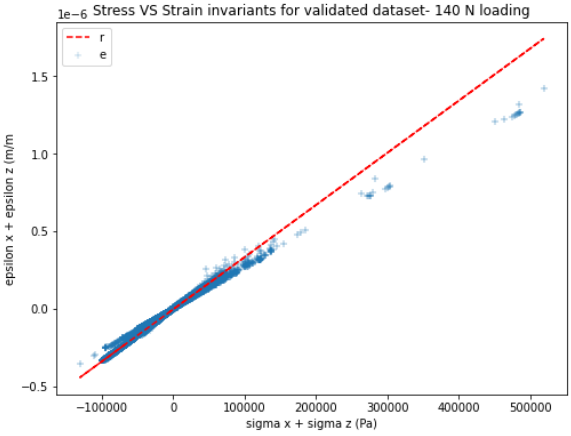

Figure 4.5: stress-strain invariant 140 N

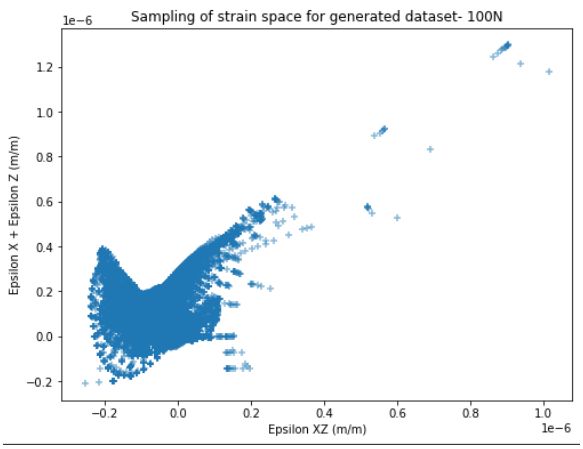

Figure 4.6: Strain Space; load-100N

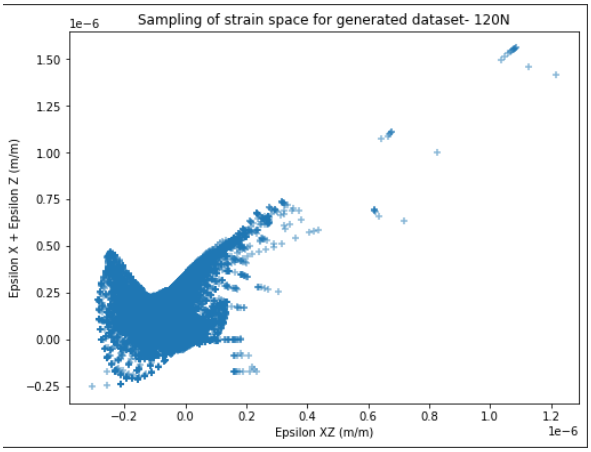

Figure 4.7: Strain Space; load-120N

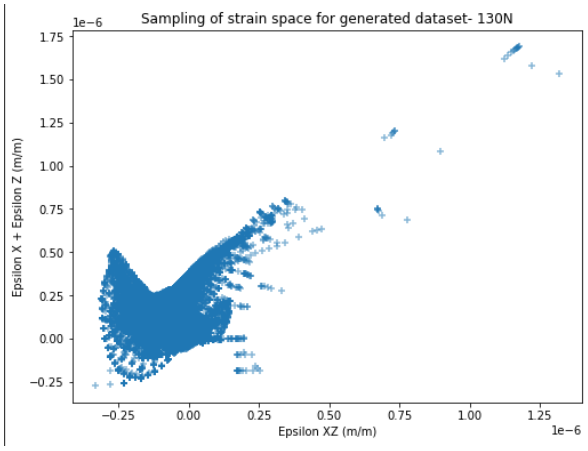

Figure 4.8: Strain Space; load-130N

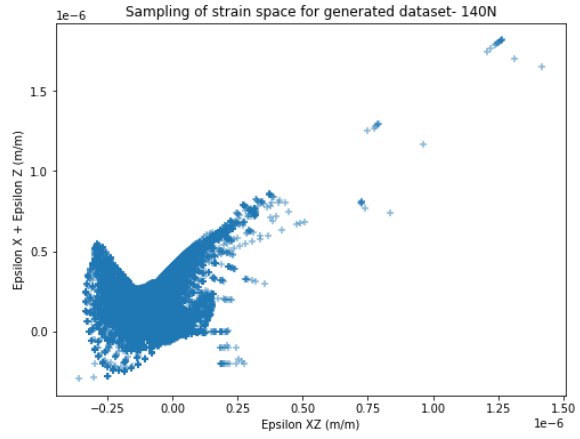

Figure 4.9: Strain Space; load-140N

max-depth = 15, *max-features* = sqrt(6) and impurity measurement method = Gini. For implementation of the SOM, we used a python package that helps us build SOM classifiers and regressors called **SuSi**. For Decision Tree and assembling the stacking regressor, we used Sci-Kit Learn's decision tree library along with their pipeline and stacking model builder. We used the compute power of Google Colab hosted on a non-local runtime. We save the following model and load the trained hyperparameters for testing our behaviour on the testing dataset.

This trained stacking model is further used on another test specimen, a same dimension stainless steel plate with multiple eccentric holes in it with 120 N loading condition, as seen in Fig 4.10. We evaluate our model's performance on the test datasets with R2 score and distance-wise error metrics. R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. The correlation explains the strength of the relationship between an independent and dependent variable.

$$R^2 = 1 - \frac{SSR}{SST} \tag{4.1}$$

where SSR (Sum of Squares of Residuals) is the sum of the squares of the difference between the actual observed value (Y) and the predicted value (Y'). SST (Total Sum of Squares) is the sum of the squares of the difference between the actual observed value (Y) and the average of the observed y value (Yavg). 0 indicates that the model explains none of the variability of the response data around its mean. 1 indicates that the model explains all the variability of the response data around its mean.

For the 130 N loading single holed plate, we get an R2 score of **0.9742726**, which signifies that the model has been able to conform to most of the variance in the test dataset. The error percentages are given in Fig 4.11 and the corresponding euclidean distance error map is shown in Fig 4.12

For the multiple eccentric hole test specimen, the R2 score comes out to be **0.921504**. Similar error percentages and error heatmap are shown in Fig 4.13 and 4.14

4.1.2 L shaped beam

With the satisfactory performance of our stacking SOM + Decision Tree model, we expand the problem to another example where we test and evaluate the prediction of

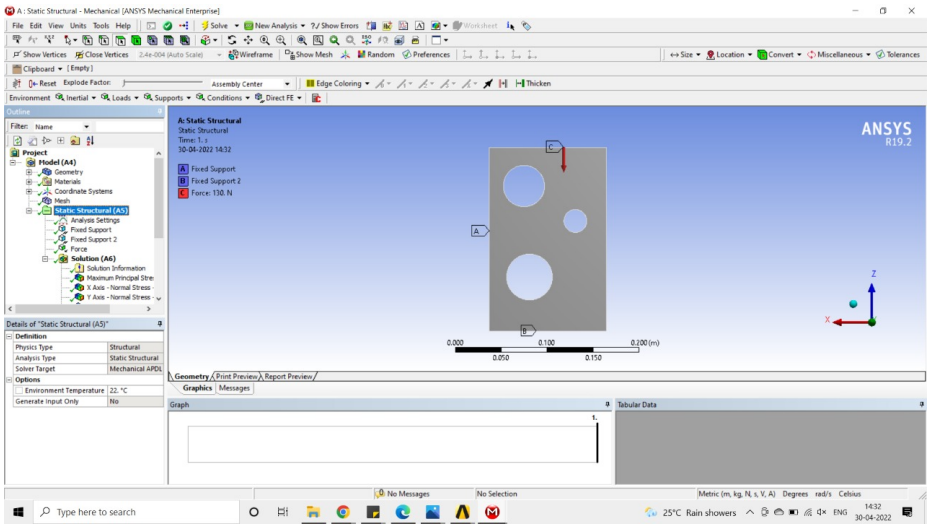

Figure 4.10: Multi hole plate modelling

	y_CV	predicted	error_dist	Error_Percentage
count	342375.000000	342375.000000	342375.000000	342375.000000
mean	-21062.788820	-21058.500929	999.784304	10.019191
std	13296.184712	13255.516758	1088.528962	341.762046
min	-98723.000000	-98128.900000	0.000000	0.000000
25%	-30957.000000	-30820.400000	282.000000	1.781567
50%	-21521.000000	-21562.700000	664.200000	4.366400
75%	-8677.150000	-8673.260000	1367.500000	8.655209
max	132280.000000	126488.000000	53981.200000	129105.754456

Figure 4.11: Error percentage for 130 N loading plate

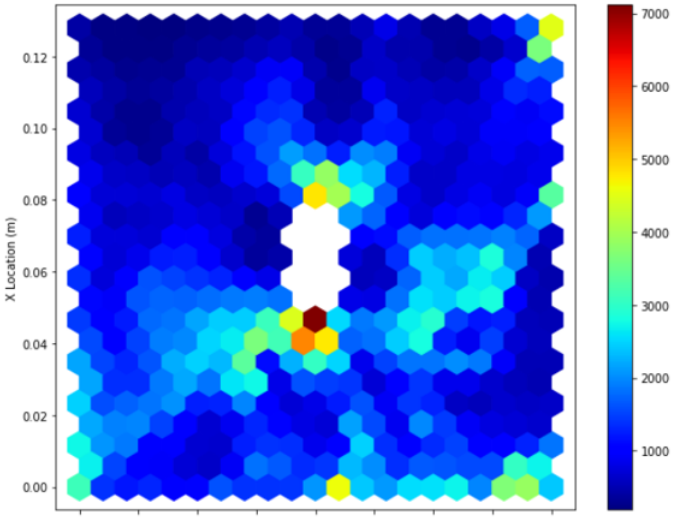

Figure 4.12: Error distance between predicted and actual values; 130 N loading plate

stress on an L shaped beam (Fig 4.15). The dimensions of the beam are as such: Total height (H)= 1m, Total width (W) = 0.6m, Width of vertical branch (w1) =0.2m, Width of horizontal branch (w2)= 0.3m. The beam has a circular hole of radius=0.075m at location (0.2m,0.15m). Material properties, as mentioned before, are the same as that of our rectangular plates (E=200 GPa and =0.3). The base of the beam is fixed and the top is horizontally displaced in the x-direction. The two material datasets were generated from classical simulations from ANSYS on a mesh of 12624 elements (65474 elements). The first dataset was generated by displacing the top by 0.002m and the

	y_CV	predicted	error_dist	Error_Percent
count	319777.000000	319777.000000	319777.000000	3.197770e
mean	-29004.437357	-27810.169194	3281.721299	3.831175e
std	22378.493334	21324.144858	5342.344901	2.491691e
min	-153000.000000	-98141.200000	0.000000	0.000000e
25%	-40783.000000	-39786.900000	457.890000	3.413429e
50%	-34088.000000	-30849.500000	1422.100000	8.258973e
75%	-7104.900000	-6556.460000	3954.700000	1.639534e
max	152000.000000	126439.000000	66481.900000	1.144588e

Figure 4.13: Error percentage for multiple holed plate

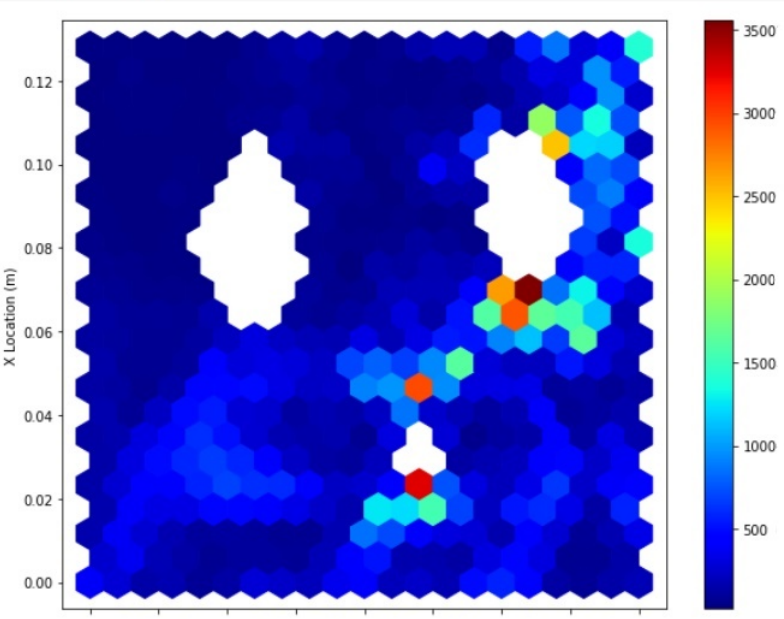

Figure 4.14: Error distance between predicted and actual values; multiple holed plate

second dataset was generated with 0.0045m displacement. As in the previous examples,

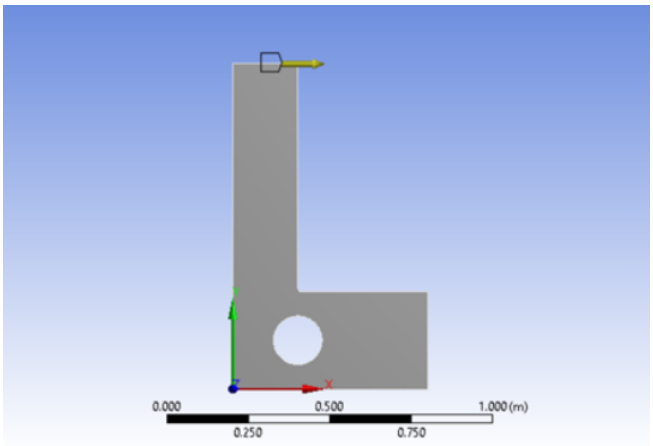

Figure 4.15: L-shaped beam modelling

we meshed the structure to 300k + nodes. We trained our stacking architecture on a part of the data and used the rest of the data-points for testing, using a test train split. The model achieved the R-squared score of **0.986** on the train dataset. Further, when the model was tested on the second data set a R-square score of 0.981 was observed

	y_CV	predicted	error_dist	Error_Percentage
count	2.160700e+04	2.160700e+04	2.160700e+04	2.160700e+04
mean	-3.192069e+06	-3.193925e+06	9.542662e+04	2.579223e+03
std	1.420473e+07	1.415861e+07	6.048119e+05	1.035974e+05
min	-3.720000e+08	-3.398000e+08	0.000000e+00	0.000000e+00
25%	-3.240000e+06	-3.248000e+06	4.816428e+01	2.631579e-01
50%	-2.858300e+02	-3.175142e+02	3.875400e+03	1.444390e+00
75%	1.891600e+04	1.709332e+04	3.666667e+04	9.721088e+00
max	1.080000e+08	1.070000e+08	3.220000e+07	1.137619e+07

Figure 4.16: Error percentage for L-shaped beam

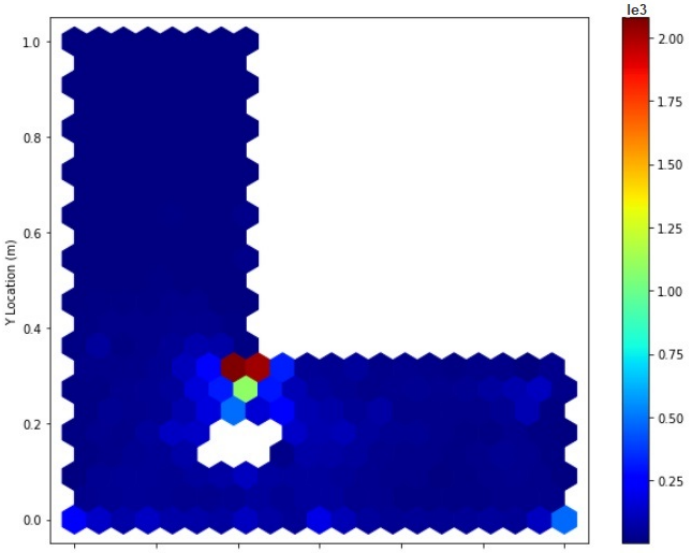

Figure 4.17: Error distance between predicted and actual values; L-shaped beam

which indicates a good generalization score by the model.

Chapter 5

Conclusion

5.1 Conclusion

Stress and Strain, although linear in relationship, exhibit a complex polynomial form to define a non-linear loading with eccentricities, deformations and non-uniform load bearing members. The data derived from the DDCM methodology concurs with our cross-validation results since ANSYS simulates a boundary value problem with the appropriate constraints and equilibrium satisfied.

Use of Self-Organizing Maps is a very apt choice of learning algorithm in our case since the principles of dimensionality reduction takes into effect in the presence of either very high dimensional variable datasets or datasets with a large sample. Together with Decision Tree, we use these algorithms to feed into a stacking regressor with a final estimator of the type RandomForest regressor. Unlike other polynomial function approximation methods, using a stacking model with appropriate algorithms running in tandem does not let the model overfit on the training data. Therefore in our examples of rectangular plates and L-shaped beam, we could generalize the model to understand the variability of the data, thus getting R2 scores close to 1. Finally, our data-driven methodology opens new avenues such as publicly shared material data repositories as well as software services which can enable the prediction of isotropic and anisotropic material behaviour under different loading, even dynamic force systems; since this methodology would work due to it's inherent nature irrespective of the physical state of the system. In the scope of this project, we have not looked at using Deep Learning techniques and training appropriately built Neural Networks to predict

material responses. Data driven design of load bearing members is still open to a lot of model-free techniques that can be ventured into in more detail.

References

[1] Laurent Stainier, Adrien Leygue, Michael Ortiz, *"Model-free data-driven methods in mechanics: material data identification and solvers"*, Springer-Verlag GmbH Germany, 4 June 2019

[2] Dongil Shin, Yoon Young Kim, *"Data-driven approach for a one-dimensional thin-walled beam analysis"*, Computers & Structures, Volume 231, 15 April 2020

[3] T.Kirchdoerfer, M.Ortiz, *"Data-driven computational mechanics"*, Computer Methods in Applied Mechanics and Engineering Volume 304, 1 June 2016

[4] Michael Ortiz, *"(Model-Free) Data-Driven Computational Mechanics"*, GALCIT Colloquium – (Advanced seminar course Ae 208 abc), April 17 2020

[5] R. Ponmalai, C. Kamath, *"Self Organizing Maps and their Applications to Data analysis"*, Lawrence Livermore National Security, 24 September 2019

[6] S.B Kostiantis, *"Decision Trees: a recent overview"*, Springer Science+Business Media B.V. 2011, 29 June 2011

[7] Thomas Hecht, Mathieu Lefort, Alexander Gepperth, *"Using self-organizing maps for regression: the importance of the output function"*, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 22-24 April 2015

[8] Bohdan Pavlyshenko, *"Using Stacking Approaches for Machine Learning Models"*, IEEE Second International Conference on Data Stream Mining and Processing (DSMP), 2018