PRIFYSGOL
BANGOR
UNIVERSITY

School of Computer Science and Electronic Engineering
College of Environmental Sciences and Engineering

Artificial neural networks in the context of additive manufacturing

---

Jorge Francisco Oca Ramirez

Submitted in partial satisfaction of the requirements for the
Degree of Bachelor of Science
in Computer Science

Supervisor Dr. Abdullah Mamun

May 2022

Acknowledgements

I would like to thank Dr. Abdullah Mamun for his guidance throughout the project, and Dr. Gauri Mahalle for her advice using ANN and MATLAB. Also, Dr. Yanhua for organising the module and providing help.

Statement of Originality

The work presented in this thesis/dissertation is entirely from the studies of the individual student, except where otherwise stated. Where derivations are presented and the origin of the work is either wholly or in part from other sources, then full reference is given to the original author. This work has not been presented previously for any degree, nor is it at present under consideration by any other degree awarding body.

Student:

Jorge Francisco Oca Ramirez

Statement of Availability

I hereby acknowledge the availability of any part of this thesis/dissertation for viewing, photocopying or incorporation into future studies, providing that full reference is given to the origins of any information contained herein. I further give permission for a copy of this work to be deposited with the Bangor University Institutional Digital Repository, and/or in any other repository authorised for use by Bangor University and where necessary have gained the required permissions for the use of third party material. I acknowledge that Bangor University may make the title and a summary of this thesis/dissertation freely available.

Student:

Jorge Francisco Oca Ramirez

Abstract

Additive manufacturing (AM) is a manufacturing technology used to create 3D objects and parts. AM has been mentioned before in what is known as the new Industry 4.0 where humans are only involved in essential processes and computers are interconnected to make decisions. The goal for many industries is to create the most precise parts while maintaining safety and cutting down on material waste and cost. To achieve these goals, data needs to be present and analysed so that data driven-decisions can be made. Material scientists are starting to use machine learning (ML) algorithms such as artificial neural networks (ANNs) to find complex relationships and predict material properties. Tuning the parameters in AM can be hard, but ANN is capable of making it an easier and quicker task. The objective of this research is to find the correlation between machine parameters (print speed, layer height, fan speed, etc) and mechanical properties (roughness, % elongation, and ultimate tensile strength) in the context of AM. An ANN model was built to find this correlation. The ANN model proposed was capable of predicting the mechanical properties with an accuracy of 83% overall, thus indicating an existing correlation between the machine parameters and mechanical properties studied. It can be said that ANNs can be an effective way of predicting mechanical properties if the parameters are tuned effectively, which can help avoid having to print and test parts.

Contents

List of Figures

Chapter 1

Introduction

1.1 Additive manufacturing

Additive manufacturing (AM), also known as 3D printing, rapid prototyping, and additive fabrication is a manufacturing technology used to create 3D objects and parts [15]. It works by extruding the chosen material layer upon layer, usually plastic or metal often in the form of a filament, wire, or powder. A laser can also melt powdered material deposited on the bed. The extruder nozzle is heated up and deposits multiple 2D layers to form a 3D object. This manufacturing process requires to have a Computer-Aided Design (CAD) model where the object is designed, to be later converted into an standard triangulation language (STL) file. The STL file is made up of approximated triangles and is sliced into layers containing all the information of the object to be printed [3].

Intelligent manufacturing is mentioned multiple times in what is known as Industry 4.0. This new term proposes only essential human engagement by having interconnected computers, smart materials, and intelligent machines able to communicate with each other and make decisions [16]. In the current age of science and technology, material science plays an essential role. New materials are being used in industries such as housing, manufacturing, transportation, aviation, medicine, etc, and scientists are working very hard to develop these materials with specific properties. They study the mechanical properties, strength, rigidity, and ductility of each material to come up with the best options for each practical application [19].

Selective laser sintering (SLS), selective laser melting (SLM), and electron beam melting (EBM) use a powder that is melted using a laser. Other processes like fused deposition modelling (FDM) use thin filaments which are melted and extruded by the extruder nozzle [13]. Plastics, metals, polymers (nylon, copper reinforced polymers, etc), and ceramics are some of the materials that could be used in SLS, SLM, and EBM. The EMB process occurs in a high vacuum chamber to avoid oxidation and is a potential candidate for use in outer space. Materials used in FDM are polycarbonate (PC), acrylonitrile butadiene styrene (ABS), polyphenylsulfone (PPSF), and PC-ABS blends. The FDM additive manufacturing process does not require chemical post-processing, or resins to cure, and uses cheaper machines and materials [3]. The future of the additive manufacturing industry relies on the automotive, aerospace, and medical industry where the goal is to create the lightest and most precise parts while maintaining safety [9]. Another advantage of AM is that it cuts down on material waste, which saves on costs for the industry. Some of the particular defects present in AM are an increase in porosity and distortion due to the rapid rate of cooling [5].

1.2 Data science

Nowadays, the concepts of big data, data science, and data analytics are gaining interest from the public and researchers, and scientists are working together to try and make the most out of all the data available. It should be of high importance that companies within the AM industry start collecting and storing all the tests carried out so material scientists can use the data for developing new materials.

In the AM industry, there is a constant improvement in the methods used and objects are continuously created using various materials. To see if the objects built are appropriate for certain applications, tests need to be carried out. These tests can produce hundreds to thousands of outputs containing machine parameters and mechanical properties. Most of these tests are never entered into any datasets and therefore never analysed. If all the tests were collected and available to use, material scientists would make faster improvements in the material field, and for them to use all these data, they would need to apply big data and data analysis methods.

Big data is better defined by the ‘Four Vs’. These are volume, velocity, variety, and veracity. The AM industry can achieve large volumes of data by collecting all the tests being performed. Velocity means that data is constantly arriving, such as when new tests are performed and new values are stored. Variety is related to the number of different coding or formats that data arrive in. It can be in .txt, .csv, or .por and they all need to be changed before performing any actions on the data. Finally, veracity refers to the quality and truthfulness of the data that is being obtained. Sometimes, sensors are collecting the data, and these sensors can not be properly calibrated, break, or are measuring the wrong thing.

Big data does not tell us anything on its own, it needs to be analysed to get information out of it so data-driven decisions can be made (See Figure 1.1). This is where data analysis and data science play a big role. These two terms are very different. Data analysis is used to answer questions with the data collected, while data science tries to predict what might happen [24]. In the AM industry, data analysis is used, for example, to see if there is a correlation between some parameters and a part breaking more easily. In cases such as how to create models capable of predicting the mechanical properties an object will have without having to do those tests, data science is used.

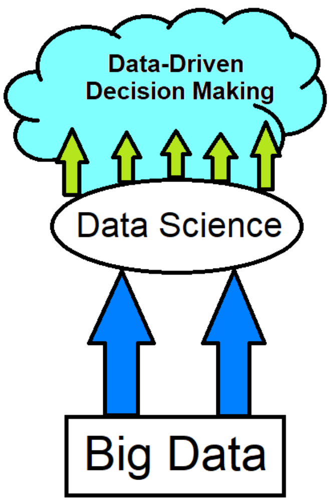

Figure 1.1: Data science for decision making in the AM industry

1.3 Machine learning

Machine learning (ML) is a type of Artificial Intelligence (AI) that focuses on the use of data and algorithms to predict new outcomes. It simulates the process of how humans learn (by using historical data to learn) and makes its own decisions that improve over time [18]. We can classify ML algorithms into supervised, unsupervised, and reinforcement learning.

Supervised learning consists of providing the programme with a set of labelled data which is used as training for the algorithm so it can later be used to make predictions. The weight of the neurons can be modified in the training process to give a more accurate prediction. In unsupervised learning, the programme is given a set of unlabelled data and it finds patterns that might not have been seen or discovered yet within that set of data. Lastly, reinforcement learning works by allowing the model to learn from trial and error by getting rewards every time it makes the right decision.

The availability of bigger datasets, paired with advances in algorithms and exponential increases in computer power, has sparked unprecedented interest in ML. ML is widely used in regression, classification, clustering, or dimensionality reduction tasks. Examples of real-world use cases are image and speech recognition, web search engines, fraud detection, and email filtering.

1.3.1 KNN and SVM algorithms

The K-Nearest Neighbors (KNN) algorithm is a type of supervised ML algorithm used to solve classification and regression problems. The KNN algorithm works by storing all possible cases and then classifying the new cases by counting how many k-neighbours are around. It then places this new case into the group with the most common cases around. This algorithm is computationally expensive and all variables have to be normalised to avoid one variable being more dominant than others [2].

Another type of supervised ML algorithm is the Support Vector Machine (SVM) algorithm used to solve classification and regression problems. All data items are plotted in an n-dimensional space (n being equal to the number of features), and then the classification is performed by finding the hyper-plane between the two classes. In the SVM algorithm, all variables have to be normalised to avoid one variable being more dominant than others [33].

For a better understanding and getting to know how simple ML algorithms work, we experimented with using the KNN and SVM algorithms in a classification problem using the dataset 'Social_Networks_Ads.csv'. This dataset determines whether a user purchased a particular product by clicking on an advertisement on a website based on their salary, age, and gender. If the product has been purchased, it will appear as a '1'. If the user did not purchase the product, a '0' will appear. The data are split into 70% for the training data, and 30% for the test data.

For the KNN algorithm, '6' is the number of neighbours used by the algorithm to classify each value, and for the SVM algorithm, the kernel used is linear. The kernel is a method used to take data as input and transform it into the required form of processing data. It transforms the training set of data so that a non-linear decision surface can be transformed into a linear equation in a higher number of dimension spaces.

To calculate the results of the KNN and SVM algorithms, a confusion matrix table was used to define the performance of a classification algorithm. It looks like the following:

		Prediction	
		0	1
Reality	0	TN	FP
	1	FN	TP

TN: True Negative	FP: False Positive
FN: False Negative	TP: True Positive

Positive or Negative refers to the prediction, if it predicts 1 it will be positive, if it predicts 0 it will be negative.

True or False refers to whether the prediction is right (True) or not (False).

The performance of the algorithms is seen using accuracy, precision, and recall as parameters:

- Accuracy measures the percentage of cases where the model is right.

    The formula to calculate it is: (TP + TN)/(TP+TN+FP+FN)

- Precision measures the quality of the model in classification tasks.

    The formula to calculate it is: TP/(TP+FP)

- Recall informs us about the amount that the model is capable of identifying.

    The formula to calculate it is: TP/(TP+FN)

In Figure 1.2 we can see image representations of the training and test sets of both models (KNN and SVM):

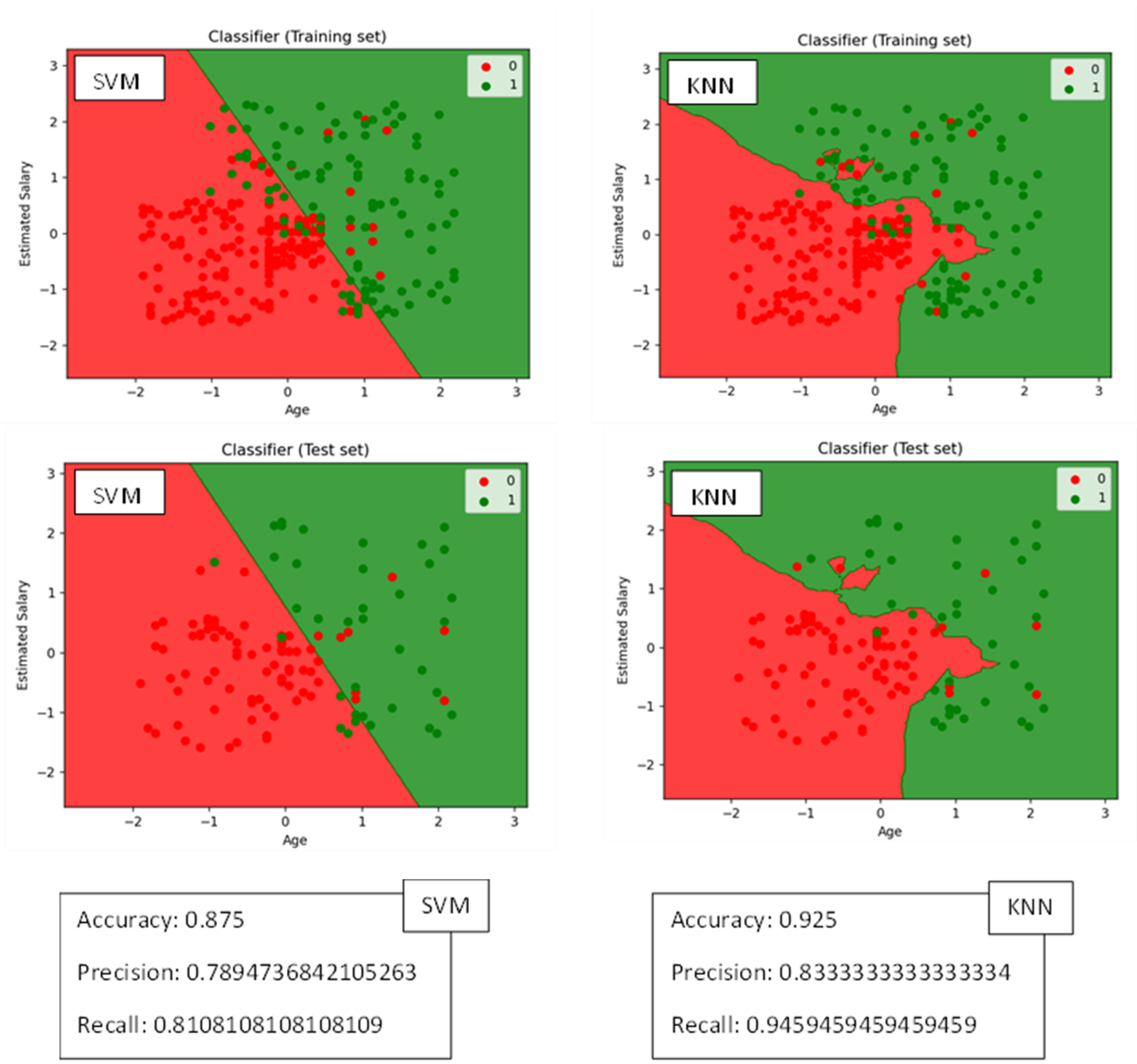

Figure 1.2: KNN and SVM results

After running both models, we can see that the KNN algorithm can predict the results more accurately for this particulate dataset. KNN has done a better job in this classification problem as ’k=6’, although if we had to generalize more and ’k’ was higher, SVM would do a better job finding the hyperplane between the two different classes. KNN is linear to the number of data points there are. Therefore, it will perform poorly when looking at time efficiency in a larger dataset.

1.3.2 Artificial neural networks

Artificial Neural Networks (ANN) are a type of ML algorithms created as a generalisation of mathematical models of biological nervous systems. As they are based on the human brain, the synapsis is represented by connection weights between the nodes or neurons [4]. As displayed in Figure 1.3, the basic ANN architecture is made of three types of layers: an input, a hidden, and an output layer. The input layer contains the data in numerical form that will be passed to the rest of the ANN layers. The hidden layer can contain one or more layers where the calculations are made using different weights between the neurons. The output layer gets the output or result of the model once all the calculations are made. In every node, there is a transfer function responsible for taking the output of the previous node and multiplying it by the weight between those two nodes.

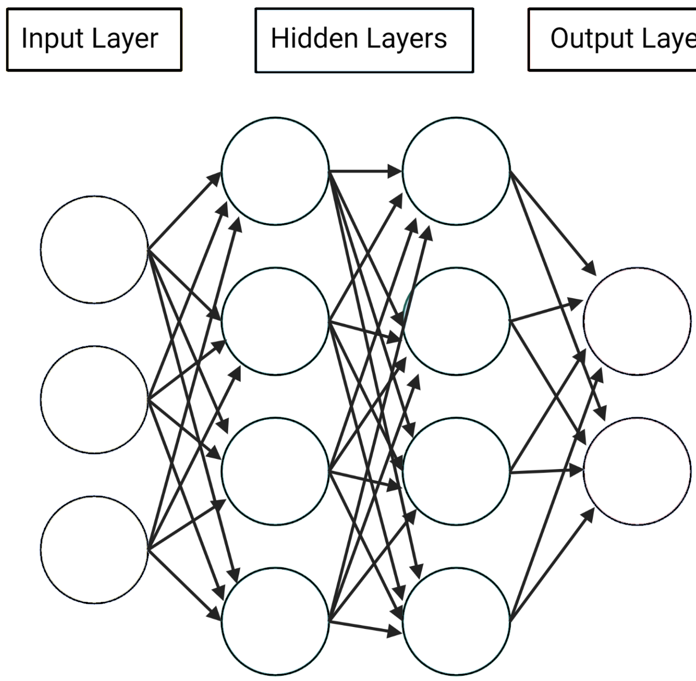

Figure 1.3: Multi-layered ANN graph representation

To find new materials, materials scientists are now looking towards using ANN models to find complex relationships between the machine parameters and the material properties. Xinbo Qi et al. [25] stated that “the combination of

AM and ANNs has demonstrated great potential for realising the attractive concept of "agile manufacturing" in industry". They also found that ANNs rely strongly on the quantity and quality of the data you have, thus making it a difficult and precise process. The use of ANNs will make choosing the materials easier and cheaper, as by changing the machine parameters we can predict what properties the material will have. One of the most famous programming languages for data science is Python. It provides free access to many useful libraries such as NumPy, TensorFlow, Pandas, etc, and makes the development of ANN models very straightforward.

The objective of this research is to find the correlation between input (machine parameters) and output (mechanical properties) in the context of AM process. We will be looking at 3D printer parameters such as wall thickness, nozzle temperature, print speed, laser power, or layer height and try to correlate these with the ultimate tensile strength (UTS), % elongation, and roughness.

1.4 Related work

The study of ANN in the context of AM has been a hot research topic in material and ML science in recent years. ML has already demonstrated its ability to accelerate both fundamental and applied research.

Attoye et al. [12] studied the Young's Modulus, yield strength, and ultimate strength of Poly Lactic Acid (PLA) and Acrylonitrile Butadiene Styrene (ABS) filaments and the effects the nozzle temperature, printing speed, and print orientation have on those mechanical properties. Overall, they found PLA has stronger mechanical properties in comparison with ABS. When using the Y-axis orientation to print, PLA presented the best mechanical properties, but when using the X-axis orientation, ABS performed better. In a study by Sondagar et al. [8] UTS was investigated and showed it is dependent on some of the mechanical properties, including laser power, scan speed, hatch spacing, and layer height. It suggested an ANN model can be effective in predicting processing parameters and can act as a replacement for the traditional analytical and mathematical approach. Another study also looked at the relationship between 3D printer parameters and tensile strength. They found that all the PLA samples printed using a crisscross raster pattern presented the highest tensile strength. Higher melt temperatures also showed a correlation with the increase in tensile strength [10]. Another study by Mahalle et al. [21] concluded that ANNs shows consistency in predicting mechanical properties. They focused on yield strength, UTS, % elongation, and other anisotropic properties.

The literature review showed research studies using datasets containing only 16 to 32 observations and obtaining low error rates [7, 30]. Others had a higher number of observations going from 50 up to 375 [34, 22, 1]. Although some researchers obtained low error values and good results, it is known that ML algorithms like ANNs will be more accurate if a higher number of observations are used.

Chapter 2

Methodology

2.1 Dataset information

The data were obtained from research by the mechanical engineering department from the Selçuk University in Turkey. They were produced in laboratory conditions and are appropriate to the research (this is, following strict guidelines and rules on the correct use of different equipment). The data have been checked to be valid, always using the appropriate instruments and units for each measurement. The tolerance of error will depend on the precision of the machine used to measure each variable.

The dataset contains 50 observations with 12 parameters in total. 9 of these parameters are the machine parameters, in this case, parameters from the Ultimaker S5 3D printer, and the other 3 parameters are measured mechanical properties. The mechanical properties were measured with a Sincotec GMBH tester capable of pulling up to 20kN. The machine parameters are layer height (mm), wall thickness (mm), infill density (%), infill pattern (grid or honeycomb), nozzle temperature (ºC), bed temperature (ºC), print speed (mm/s), material (PLA or ABS), and fan speed (%). The mechanical properties measured are roughness (μm), UTS (MPa), and elongation (%). We can identify and divide the data into two separate groups: independent variables and dependent variables. The independent variables are the ones that can be changed and in this experiment are the machine parameters (print speed, nozzle temperature, bed temperature, etc). The dependent variables are usually the result or output of the ANN model, the things that you are actively trying to measure, such as the UTS, roughness, and % elongation.

Layer height goes from 0.02 mm to 0.20 mm and the wall thickness ranges between 1 mm and 10 mm. Infill density and fan speed were measured as a percentage going from 10% to 90%, and from 0% to 100% respectively. Infill pattern and material were introduced in the dataset as categorical values, being grid or honeycomb, and PLA or ABS. Nozzle and bed temperatures increase in intervals of 5ºC or 10ºC, ranging between 200ºC and 250ºC for the nozzle temperature, and between 60ºC and 80ºC for the bed temperature. Print speed has 3 different speeds: 40 mm/s, 60 mm/s, and 120 mm/s. Lastly, the measured mechanical properties values that were introduced in the dataset are as follows: roughness varies between 21 (μm) and 368 (μm), elongation is measured at 0.4% for the minimum and 3.3% for the maximum elongation, and UTS goes from 4 MPa to 37 MPa. See Appendix A for graph representations of each variable.

The dataset had to be cleansed before the analysis could start. Changes were made to the infill pattern and material values as they were entered as categorical values but need to be numerical values. For the infill pattern parameter, the 'grid' value was assigned the number '1', and the 'honeycomb' value was assigned the number '0'. For the material parameter, the same approach was taken. The 'ABS' value was replaced for a '1', and the 'PLA' value for a '0'. Each parameter was checked for the presence of null or odd values and adjusted accordingly.

2.2 ANN parameters

For analysing the data, ANNs were used. A model was created using the 9 machine parameters as input to try and find a correlation with the UTS, % elongation, and roughness.

Two different programming languages were used to create ANN models, Python and MATLAB. Python would have been the preferred programming language, but because of time constraints, the final ANN model was built using MATLAB. However, a trial ANN model was tested using Python, as can be seen in Figure 2.1. Future research should aim to use Python for creating the ANN model as it offers more libraries and customisation in comparison with MATLAB.

```python
import tensorflow as tf
from tensorflow import keras
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense

# load the dataset
dataset = loadtxt('data.csv', delimiter=',')

# split into input (X) and output (y) variables
X = dataset[:, 0:8]
y = dataset[:,11]

# define the keras model
model = Sequential()
model.add(Dense(10, input_dim=8, activation='sigmoid'))
model.add(Dense(1))

# compile the keras model
model.compile(loss='mse',
              optimizer='sgd',
              metrics=[tf.keras.metrics.MeanAbsolutePercentageError()])

# evaluate the keras model
_, accuracy = model.evaluate(X, y)
print('MSE: ', accuracy)
```

Figure 2.1: Trial ANN model built using Python programming language

The code written in Python uses the libraries NumPy, Keras, and TensorFlow. The NumPy library was used for loading the dataset into the programme. In this trial, Keras was used for defining and compiling the ANN model. Keras is an ML library which provides tools to build neural network models. It also has tools for compiling models and visualising the model results. Finally, TensorFlow is another ML library optimised for speed. It was used when compiling the model, to select the Mean Absolute Percentage Error metric. This model has two layers, the first layer is made up of 10 nodes, and is responsible for processing the input parameters using the sigmoid activation function. The second layer processes the output and has only 1 node.

The model created in MATLAB (Figure B.1 and B.2) divides the data as follows: 70% is used for training the network, 15% as validation data, and the other 15% for testing. Those percentages mean that 34 observations are used for training, 8 for validation, and 8 for testing. 70% - 15% - 15% is one of the best and most used ways of dividing the dataset in ANN models. The data points are randomly divided into those percentages when running the model. Nguyen et al. [23] studied how the split ratio affects the output of an ANN model. They considered different ratios (i.e., 10/90, 20/80, 30/70, 40/60, 50/50, 60/40, 70/30, 80/20, and 90/10), and concluded that the 70/30 ratio had the best performance.

The network created is a two-layer-feed-forward network. The model was only built with two layers to avoid over-training. This can happen if the model is too powerful for the dataset presented and the ANN model just learns the data by heart. If a model has been overtrained, it will not be able to generalise well [32]. This can be avoided by reducing the number of layers or using the cross-validation method, which consists of splitting the data into sets to then test whether it performs well on data that it has never seen. The first hidden layer takes the 9 machine parameters as input, and the second hidden layer is responsible for transmitting the output. The first layer has a total of 10 neurons which use a sigmoid activation function, and the second layer has 1 neuron with a linear activation function in charge of processing the output. There are different activation functions including sigmoid, tanh or ReLU. The activation function ReLU is widely used by many data scientists when creating ANNs, but they do not work in all cases. ReLU has the problem of sometimes pushing output units close to 0, where they are considered 'dead'. For this particular problem, the model was tested with both the ReLU and sigmoid activation functions, and a better performance of the model was obtained with the sigmoid activation function.

The training algorithm chosen was the Levenberg-Marquardt (LM) algorithm. It is one of the most extensively used optimisation algorithms and surpasses simple gradient descent methods [26]. The LM algorithm was created to provide a solution for the nonlinear least-squares minimisation problem. Other algorithms were also considered. The Bayesian Regularisation algorithm works by converting a non-linear regression into a statistical problem as a ridge regression [11]. This algorithm is slower in larger datasets and its time efficiency is in the order of O(n2). The Scaled Conjugate Gradient algorithm was also considered but it is computationally expensive as it requires n cycles to reach the minimum gradient.

The model performance was measured using the mean square error (MSE), and the Pearson correlation (R-value) [27]. Plots for the model were also created. The first plot generated was the 'best validation performance' which measures the best MSE for the train, validation, and test sets. Next, an error histogram plot with 20 bins was created. The errors for this plot are calculated by subtracting the outputs of the model from the targets. Lastly, the training, validation, target, and overall plots show how the model performed in each phase.

Chapter 3

Results and discussion

To identify potential linear relationships between two variables, a correlation matrix was used. It was created using Python with the Seaborn and Matplotlib libraries. Both are data visualisation libraries for creating informative statistical graphics. The correlation matrix compared all the input and output parameters before running the ANN model. If 2 variables do not appear to be correlated in the matrix created, it does not mean there is no correlation between them, there could exist a nonlinear correlation which has not been studied. To measure the strength of the correlation, the R-value is used. For describing the R-value, a guide created by James D. Evans [14] was used. He suggests the following: for values between .00 and .19 the correlation is very weak, .20 to .39 is a weak correlation, .40 to .59 represents a moderate correlation, .60 to .79 is a strong correlation, and .80 to 1.00 is a very strong correlation. The correlation matrix can be seen on the next page in Figure 3.1. It suggests that roughness is positively strongly correlated with layer height and has a positive weak correlation with nozzle temperature. UTS only presents a positive weak correlation with layer height, wall thickness, and infill density, and has a negative weak correlation with nozzle temperature. Finally, elongation is positively moderately correlated with layer height, and negatively moderately correlated with nozzle temperature. It also has a negative weak correlation with bed temperature, material, and fan speed. All the other machine parameters did not show a linear correlation with the mechanical properties as they all present a very weak R-value.

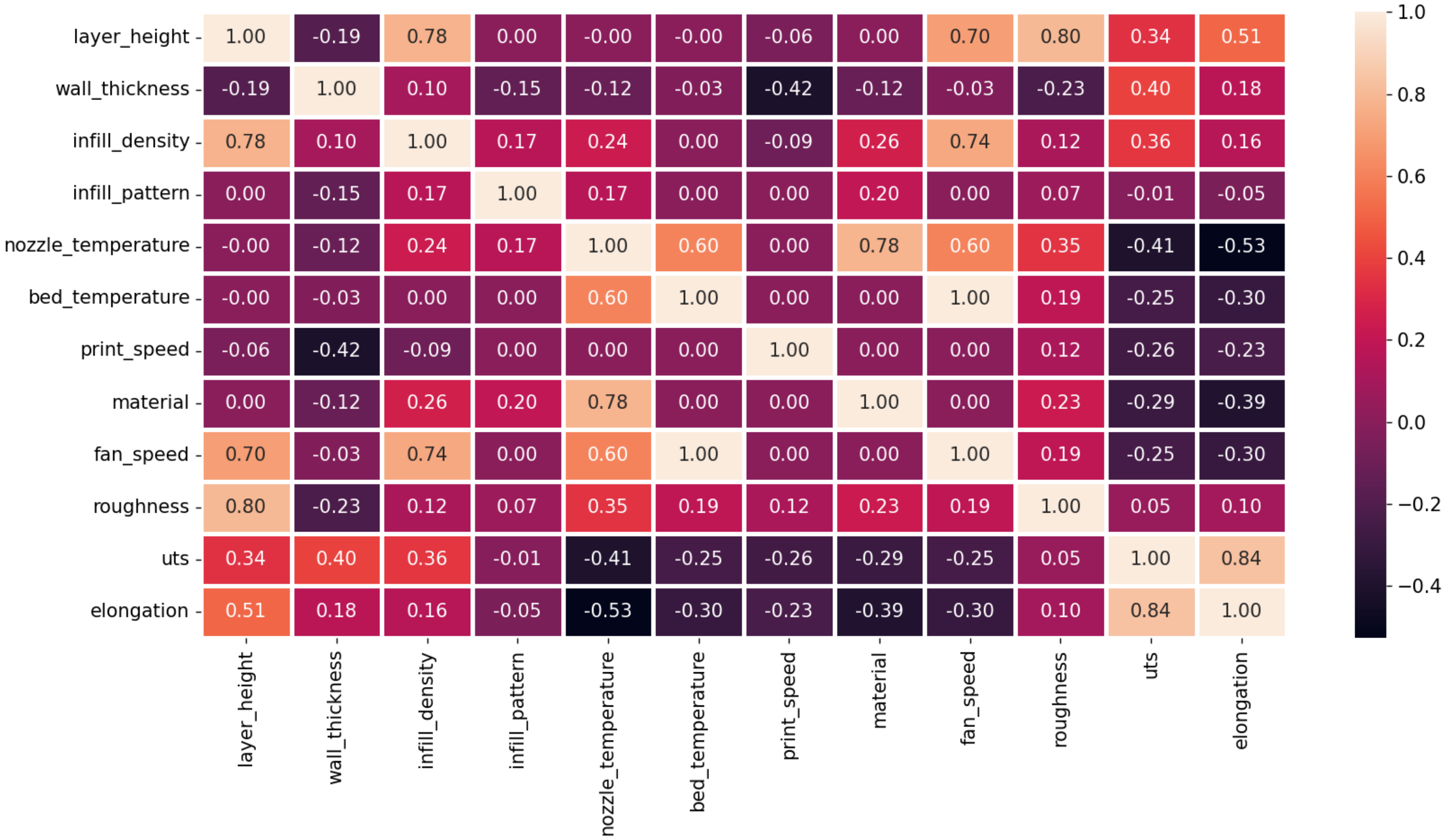

Figure 3.1: Correlation Matrix

A study by Jess Hartcher-O'Brien et al. [17] found that layer height did not present a strong correlation with roughness, although only 16 observations were studied. Another study [29] with a larger sample size discovered that layer height is directly correlated with the roughness, which agrees with the findings of this study. This could be because the layers become less visible as the layer height is reduced, and the surface of the printed component gets smoother. This shows that sample size affects the results obtained about the correlation between these two parameters.

There is a lack of research on the relationship between UTS and the machine parameters studied in this experiment, although a study by Yubo Tao et al. [31] found a correlation between layer height and UTS. Mohammad Shirmohammadi et al. [29] discovered that wall thickness is also correlated to the UTS. Ali Keshavarzkermani et al. [20] identified a correlation between UTS and printing direction, a machine parameter that was not considered in this study.

A study by Shuheng Wang et al. [35] identified a negative correlation between nozzle temperature and elongation when the nozzle temperature is increased from 210 ºC to 230 ºC. This might be because the extruded material's fluidity is too high to ensure the material's surface quality and, as a result, diminish the material's elongation. At the same time, the material's mechanical properties are reduced due to the thermal degradation of the materials.

The ANN model created was run 3 different times to see the correlation of the 9 input parameters for UTS, elongation, and roughness. The model was first run using UTS as output. The R-value for the training set achieved 0.9805, 0.9326 for the validation set, and 0.8583 for the test set, as shown in Figure C.1. The model achieved the best validation performance at epoch 3 with an MSE value of 11.1144 as seen in Figure C.2. Figure C.3 shows the UTS histogram of the errors between target and predicted values.

Using elongation as the output gave the following results. The R-values were 0.96716, 0.42791, and 0.47336 for the training, validation, and test set, as shown in Figure C.4. The best validation performance was achieved at epoch 3 with an MSE value of 0.59169 as seen in Figure C.5. Figure C.6 shows the elongation histogram of the errors between the target values and the predicted values.

Lastly, the roughness output parameter gave the following results. The R-value shows 0.86859 for the training set, 0.73818 for the validation set, and 0.83462 for the test set, as shown in Figure C.7. The model achieved the best validation performance at epoch 2 with an MSE value of 0.78026 as seen in Figure C.8. Figure C.9 shows the roughness histogram of the errors between target and predicted values.

Two different statistical metrics were used to evaluate the models. The average of the squared difference between the original and predicted values in the dataset is the MSE. It calculates the variance of the residuals, while the R-squared coefficient of determination (or R-value) shows the proportion of the variation for a dependent variable that is predictable by an independent variable. A lower value of MSE implies higher accuracy. However, a higher value of the R-value is considered desirable.

When using UTS as the output parameter, the R-value was 0.9187 overall for the training, validation, and test sets, and the MSE was 11.1144. For the elongation, the values were 0.8039 and 0.59169 respectively. Lastly, the roughness model returned a value of 0.7813 overall for the training, validation, and test sets and 0.7803 for the MSE value. Elongation and roughness had the lower MSE value, while UTS had the highest R-value. Looking at all the evaluation statistical values, the model was able to predict with more accuracy the % elongation parameter, followed by the roughness parameter, and then the UTS.

To train an ANN, large amounts of data are needed. The dataset used in this research study only contained 50 observations. A study by Ahmad Alwosheel [6] proposed a 'factor 50' rule of thumb (where the number of observations must be at least 50 times that of the network's configurable parameters) in comparison with the 'factor 10' rule of thumb that is traditionally used. They also found the 'factor 10' rule of thumb to be 'somewhat too optimistic', and as the complexity of the data generating process increases, more observations will be required to maintain validity and optimal performance. Another research [28] proposed a two-fold framework for smaller datasets and obtained an accuracy of 86.5%. To conclude, it is possible to train an ANN with a small dataset although data scientists need to be more careful and use the right tools depending on the size of the sample. Past research has provided new approaches and methods for maintaining and optimising the performance of ANNs.

There are different biases to consider when planning and designing this experiment. The first one is the information bias; not all the machine parameters that can be adjusted are needed to be the input in our artificial neural network model. The focus should only be on the variables to be examined. Predictions will not be better by having many mechanical properties, nor will the experiment be more accurate. What variables are needed or not should be carefully considered before starting the experiment. In this experiment, a correlation matrix was created before training the ANN. The correlation matrix showed the linear relationships that exist between some of the parameters. This helped understand what outputs might be expected from the ANN model and it is always beneficial to have more than one method describing the same information.

Another type of bias to acknowledge which is related to the previous one is the selection bias. This can happen if the focus is only on certain variables from the start and if other variables/machine parameters are not considered. For example, 3 variables might have been selected at the start, and because good results were obtained with these 3 variables, it is deemed unnecessary to look at how other variables affect the experiment and therefore, these variables are not considered. However, efforts should always be made to run the experiment with different variables. Thus, the ANN model was run 3 different times with a different mechanical property each time. UTS, % elongation, and roughness were tested.

Lastly, another type of bias that might come across is the availability bias. The tendency is to use the first information found and not continue researching and looking for more. It can be avoided by researching multiple papers that discuss the same ideas. When researching what 3D printer parameters affect the mechanical properties of an object, the aim should be to read many scientific papers before making any decisions, in order to have a wider spectrum of opinions/conclusions and have a wider viewpoint. Finding datasets containing more than 10 observations related to this research topic was very difficult, and a lower number of observations would have made the ANN model almost impossible to learn.

Chapter 4

Conclusions

To summarise, a dataset containing 50 observations and 12 parameters was used to find the correlation between machine parameters and mechanical properties. The ANN model proposed was a two-layer-feed-forward network. One layer with 10 nodes for processing the output with a sigmoid activation function, and another layer containing 1 node with a linear activation function before the output layer. The ANN model was capable of predicting the mechanical properties with an accuracy of 83% overall. This result, alongside the correlation matrix and past research papers, indicates that there is a correlation between the machine parameters and the mechanical properties proposed in this study. The ANN model was able to predict with great accuracy the roughness and % elongation, which have been previously studied in other papers, but failed to predict the UTS with high accuracy. This could be due to the UTS not being highly correlated to the machine parameters studied, but to other parameters such as the printing direction. It is known that datasets with many observations will perform better when using ANNs as they rely on the training data to learn and make accurate predictions. In future studies, the focus should also be on other machine parameters to find accurate correlations and make the use of ANNs possible in the field of AM. This study shows that ANNs are an effective way of finding the correlation between machine parameters and mechanical properties and are able to predict those mechanical properties with accuracy.

References

[1] ‘A neural network approach to the modelling and analysis of stereolithography processes:,’ vol. 215, pp. 1719–1733, 12 Aug. 2016 (p. 10).

[2] ‘A novel knn algorithm with data-driven k parameter computation,’ *Pattern Recognition Letters*, vol. 109, pp. 44–54, Jul. 2018 (p. 4).

[3] ‘A review of additive manufacturing,’ *ISRN Mechanical Engineering*, vol. 2012, pp. 1–10, Aug. 2012 (pp. 1, 2).

[4] A. Abraham, ‘129: Artificial neural networks,’ (p. 8).

[5] ‘Additive manufacturing of metals: A brief review of the characteristic microstructures and properties of steels, ti-6al-4v and high-entropy alloys,’ vol. 18, pp. 584–610, 1 Dec. 2017 (p. 2).

[6] A. Alwosheel, S. van Cranenburgh and C. G. Chorus, ‘Is your dataset big enough? sample size requirements when using artificial neural networks for discrete choice analysis,’ *Journal of Choice Modelling*, vol. 28, pp. 167–182, Sep. 2018 (p. 19).

[7] ‘An investigation on sliding wear of fdm built parts,’ *CIRP Journal of Manufacturing Science and Technology*, vol. 1, pp. 48–54, 5 2012 (p. 10).

[8] ‘Artificial neural network (ann) based prediction of process parameters in additive manufacturing,’ *IOP Conference Series: Materials Science and Engineering*, vol. 1136, p. 012 026, 1 Jun. 2021 (p. 9).

[9] ‘Basics, applications and future of additive manufacturing technologies: A review numerical modelling of temperature rise generation at tool , chip and workpiece in metal cutting view project optimization for the density of laser sintered parts view project basics, applications and future of additive manufacturing technologies: A review,’ *Article in*

Journal of Manufacturing Technology Research · March, vol. 5, 1 2013 (p. 2).

[10] Ö. Bayraktar, G. Uzun, R. Çakiroğlu and A. Guldas, 'Experimental study on the 3d-printed plastic parts and predicting the mechanical properties using artificial neural networks,' *Polymers for Advanced Technologies*, vol. 28, pp. 1044–1051, 8 Aug. 2017 (p. 10).

[11] F. Burden and D. Winkler, 'Bayesian regularization of neural networks,' *Methods in molecular biology (Clifton, N.J.)*, vol. 458, 2008 (p. 14).

[12] 'Correlation between process parameters and mechanical properties in parts printed by the fused deposition modeling process,' Springer Science and Business Media, LLC, 2019, pp. 35–41 (p. 9).

[13] P. Delfs, M. Tows and H. J. Schmid, 'Optimized build orientation of additive manufactured parts for improved surface quality and build time,' *Additive Manufacturing*, vol. 12, pp. 314–320, Oct. 2016 (p. 2).

[14] J. D. Evans, *Straightforward statistics for the behavioral sciences*. Brooks/Cole Pub. Co., 1996, p. 600 (p. 16).

[15] J. Gardan, 'Additive manufacturing technologies: State of the art and trends,' *International Journal of Production Research*, vol. 54, pp. 3118–3132, 10 2016 (p. 1).

[16] M. Ghobakhloo, 'Industry 4.0, digitization, and opportunities for sustainability,' *Journal of Cleaner Production*, vol. 252, Apr. 2020 (p. 1).

[17] J. Hartcher-O'Brien, J. Evers and E. Tempelman, 'Surface roughness of 3d printed materials: Comparing physical measurements and human perception,' *Materials Today Communications*, vol. 19, pp. 300–305, Jun. 2019 (p. 17).

[18] M. I. Jordan and T. M. Mitchell, 'Machine learning: Trends, perspectives, and prospects,' *Science*, vol. 349, pp. 255–260, 6245 Jul. 2015 (p. 4).

[19] S. L. Kakani, *Material Science*. 2004 (p. 1).

[20] A. Keshavarzkermani *et al.*, 'Controlling mechanical properties of additively manufactured hastelloy x by altering solidification pattern during laser powder-bed fusion,' *Materials Science and Engineering: A*, vol. 762, p. 138 081, Aug. 2019 (p. 17).

[21] G. Mahalle, O. Salunke, N. Kotkunde, A. K. Gupta and S. K. Singh, 'Neural network modeling for anisotropic mechanical properties and work

hardening behavior of inconel 718 alloy at elevated temperatures,' *Journal of Materials Research and Technology*, vol. 8, pp. 2130–2140, 2 Apr. 2019 (p. 10).

[22] J. Munguía, J. Ciurana and C. Riba, 'Neural-network-based model for build-time estimation in selective laser sintering,' *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 223, pp. 995–1003, 8 Aug. 2009 (p. 10).

[23] Q. H. Nguyen *et al.*, 'Influence of data splitting on performance of machine learning models in prediction of shear strength of soil,' *Mathematical Problems in Engineering*, vol. 2021, 2021 (p. 14).

[24] F. Provost and T. Fawcett, 'Data science and its relationship to big data and data-driven decision making,' *Big Data*, vol. 1, pp. 51–59, 1 Mar. 2013 (p. 3).

[25] X. Qi, G. Chen, Y. Li, X. Cheng and C. Li, 'Applying neural-network-based machine learning to additive manufacturing: Current applications, challenges, and future perspectives,' *Engineering*, vol. 5, pp. 721–729, 4 Aug. 2019 (p. 8).

[26] A. Ranganathan, 'The levenberg-marquardt algorithm,' 2004 (p. 14).

[27] J. L. Rodgers and ; W. A. Nicewander, 'Thirteen ways to look at the correlation coefficient,' *The American Statistician*, vol. 42, pp. 59–66, 1 1988 (p. 15).

[28] T. Shaikhina and N. A. Khovanova, 'Handling limited datasets with neural networks in medical applications: A small-data approach,' *Artificial Intelligence in Medicine*, vol. 75, pp. 51–63, Jan. 2017 (p. 19).

[29] M. Shirmohammadi, S. J. Goushchi and P. M. Keshtiban, 'Optimization of 3d printing process parameters to minimize surface roughness with hybrid artificial neural network model and particle swarm algorithm,' *Progress in Additive Manufacturing*, vol. 6, pp. 199–215, 2 May 2021 (p. 17).

[30] A. K. Sood, R. K. Ohdar and S. S. Mahapatra, 'Experimental investigation and empirical modelling of fdm process for compressive strength improvement,' *Journal of Advanced Research*, vol. 3, pp. 81–90, 1 Jan. 2012 (p. 10).

[31] Y. Tao, P. Li and L. Pan, 'Improving tensile properties of polylactic acid parts by adjusting printing parameters of open source 3d printers,' *Materials Science*, vol. 26, pp. 83–87, 1 Jan. 2020 (p. 17).

[32] I. V. Tetko, D. J. Livingstone and A. I. Luik, 'Neural network studies. 1. comparison of overfitting and overtraining,' *Journal of Chemical Information and Computer Sciences*, vol. 35, pp. 826–833, 5 1995 (p. 14).

[33] S. V. Vishwanathan and M. N. Murty, 'Ssvm: A simple svm algorithm,' *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 2393–2398, 2002 (p. 5).

[34] G. C. Vosniakos, T. Maroulis and D. Pantelis, 'A method for optimizing process parameters in layer-based rapid prototyping,' *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 221, pp. 1329–1340, 8 2007 (p. 10).

[35] S. Wang, Y. Ma, Z. Deng, S. Zhang and J. Cai, 'Effects of fused deposition modeling process parameters on tensile, dynamic mechanical properties of 3d printed polylactic acid materials,' *Polymer Testing*, vol. 86, Jun. 2020 (p. 18).

Appendices

Appendix A

Graph representation of the dataset

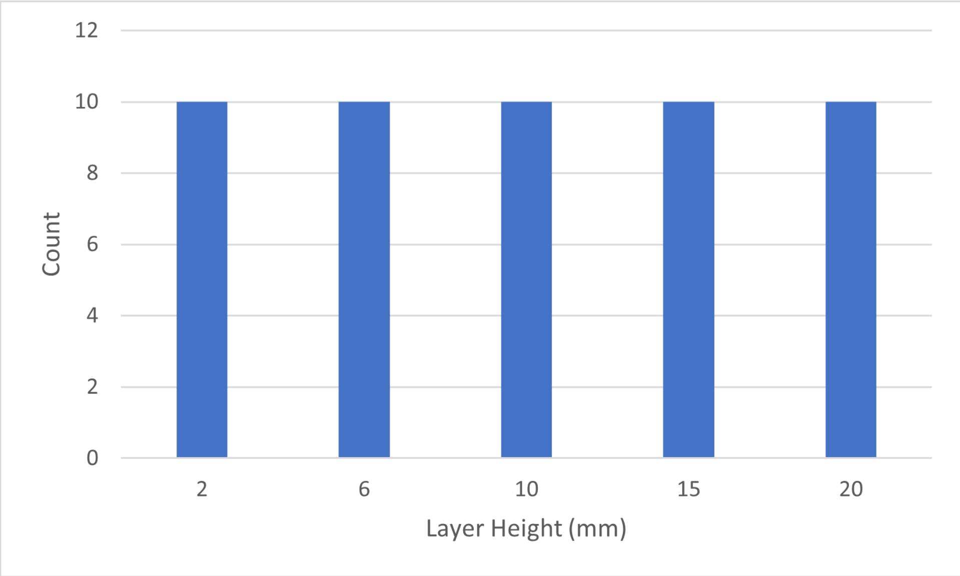

Figure A.1: Layer Height

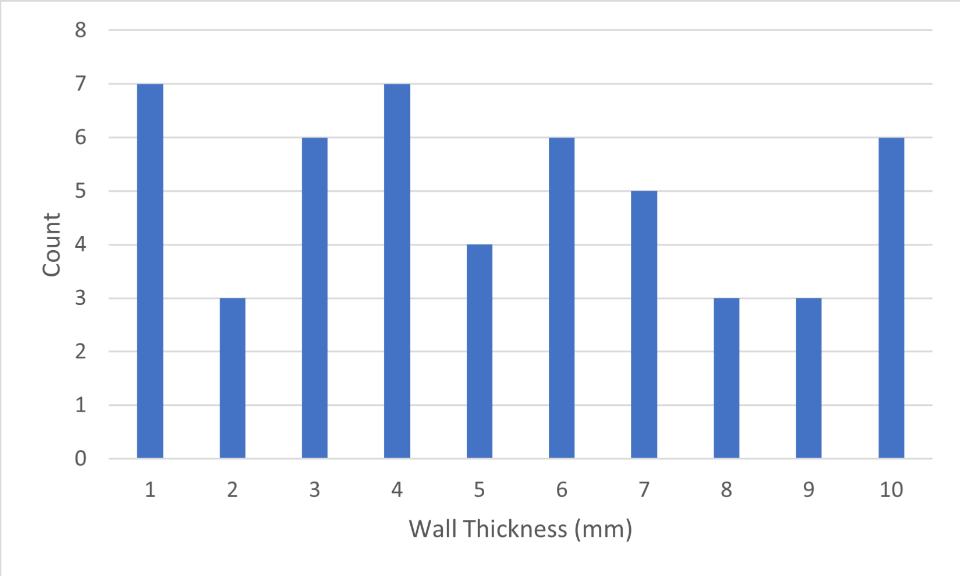

Figure A.2: Wall Thickness

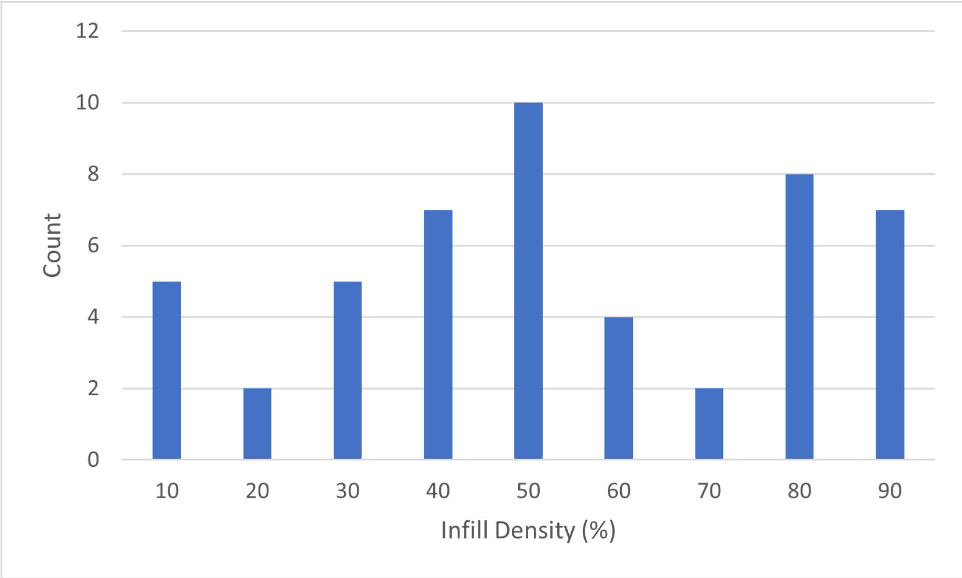

Figure A.3: Infill Density

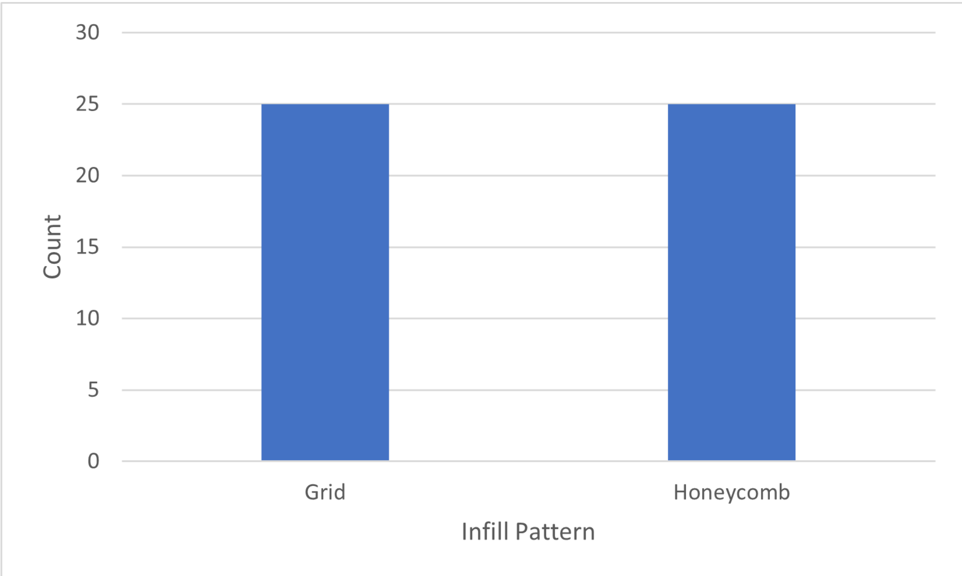

Figure A.4: Infill Pattern

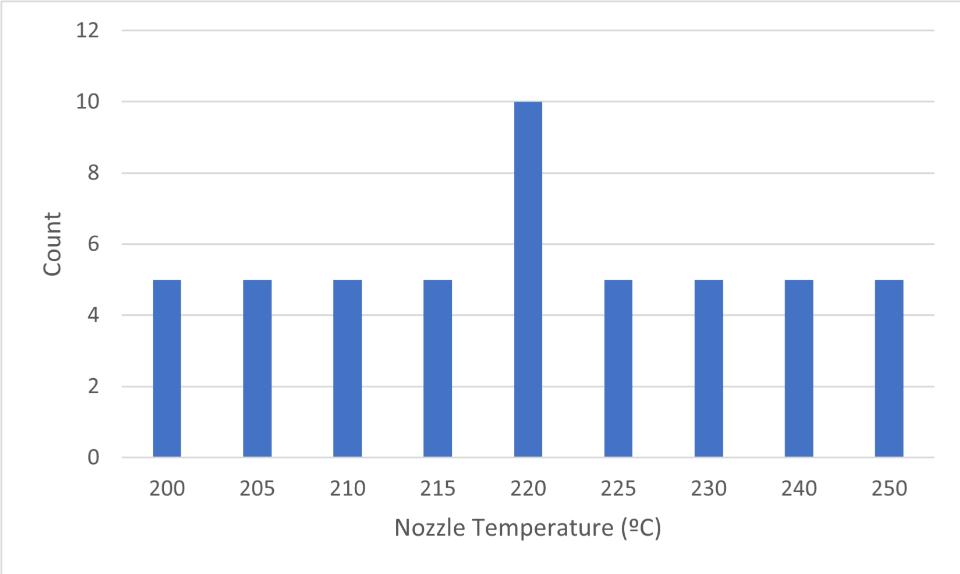

Figure A.5: Nozzle Temperature

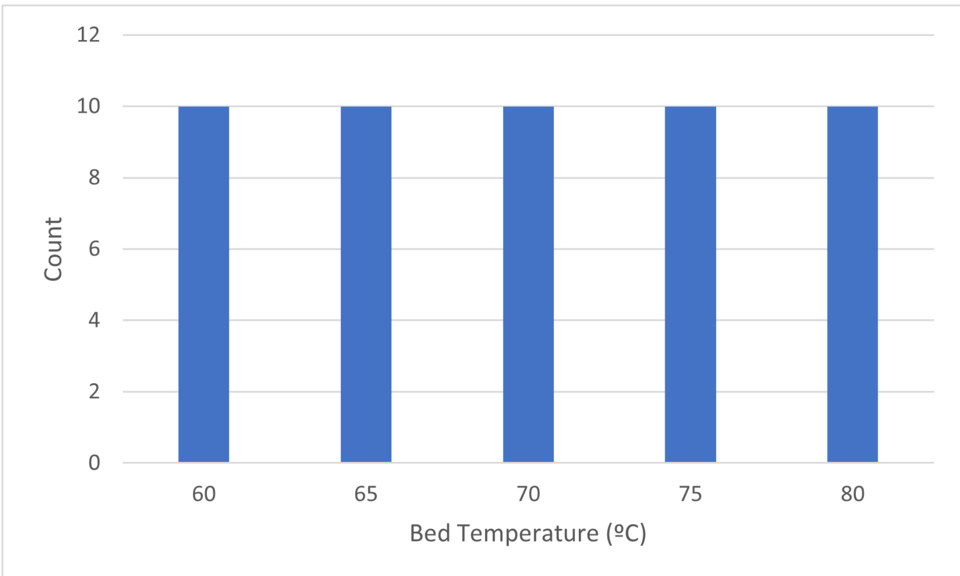

Figure A.6: Bed Temperature

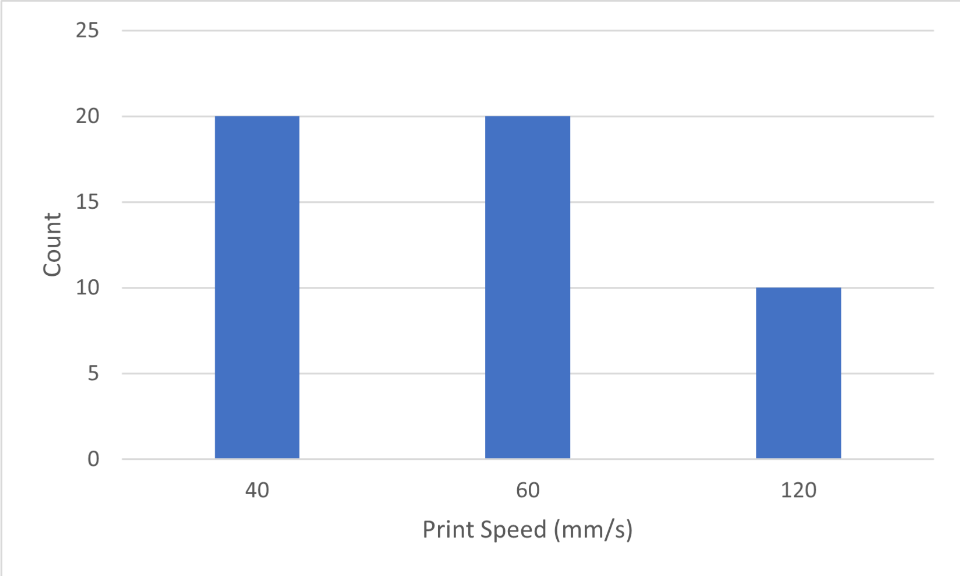

Figure A.7: Print Speed

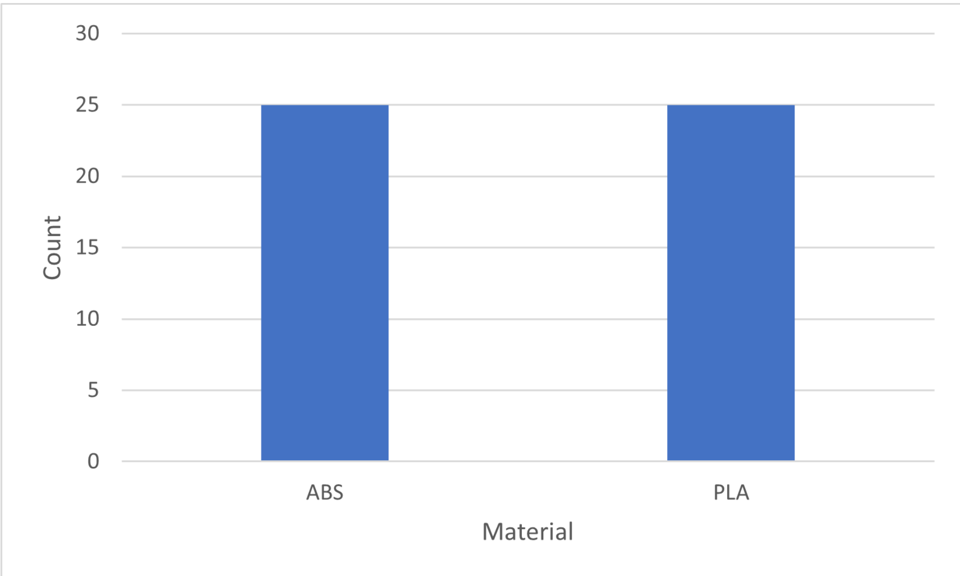

Figure A.8: Material

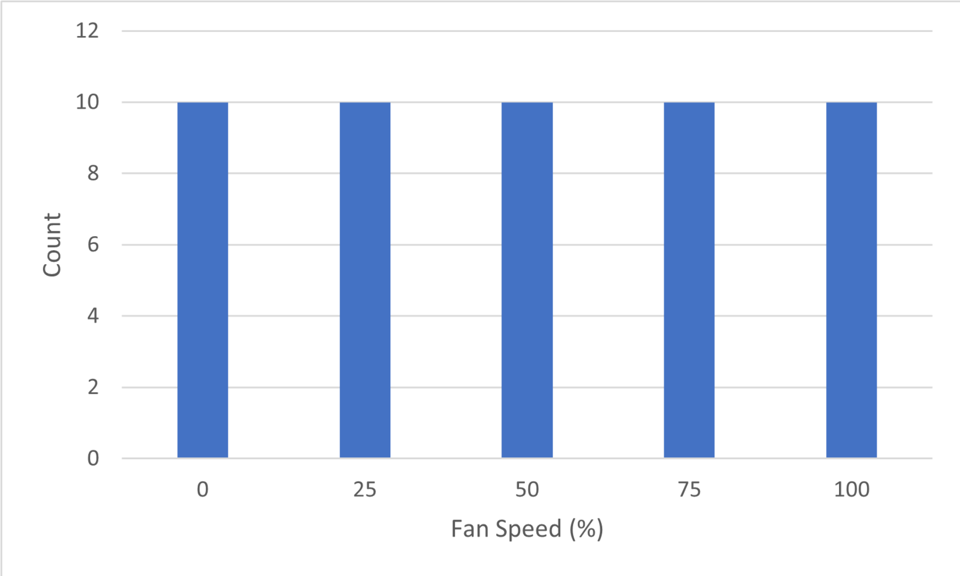

Figure A.9: Fan Speed

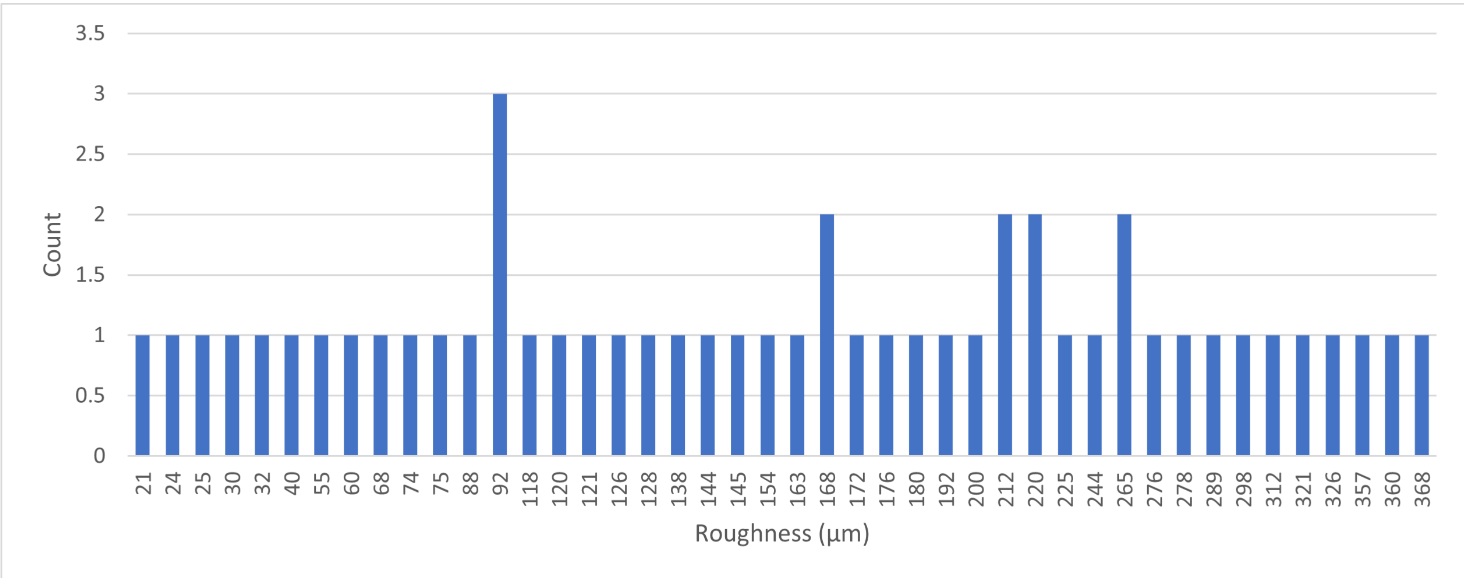

Figure A.10: Roughness

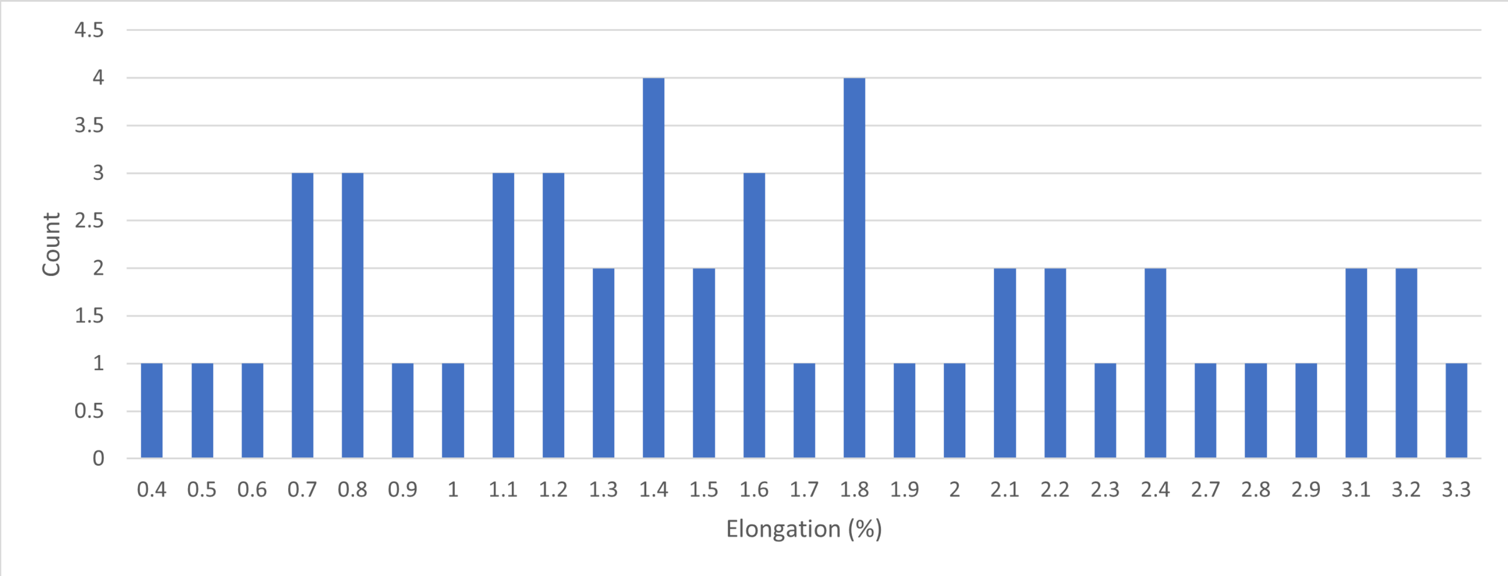

Figure A.11: Elongation

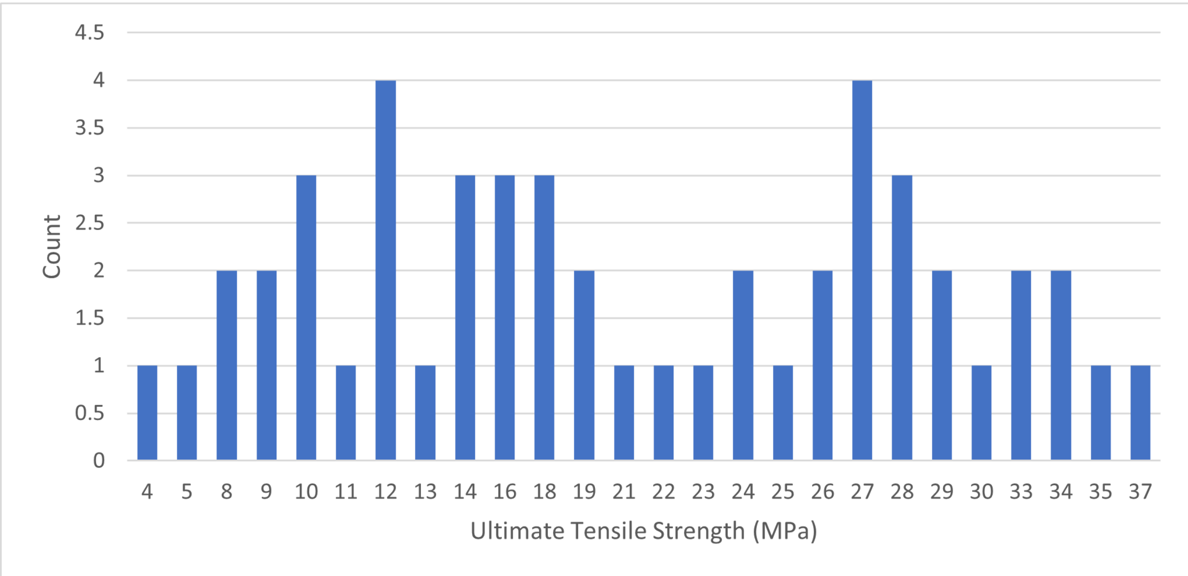

Figure A.12: Ultimate Tensile Strength

Appendix B

MATLAB code

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
%
% This script assumes these variables are defined
%
%   kaggle_input1 - input data.
%   kaggle_output - target data.

x = kaggle_input1;
t = kaggle_output;

% Choose a Training Function
% For a list of all training functions type help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output PrePost-Processing Functions
% For a list of all processing functions type help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type help nndivision
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70100;
net.divideParam.valRatio = 15100;
net.divideParam.testRatio = 15100;

% Choose a Performance Function
% For a list of all performance functions type help nnperformance
net.performFcn = 'mse';  % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);
```

Figure B.1: ANN model built using MATLAB part 1

```
% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t . tr.trainMask{1};
valTargets = t . tr.valMask{1};
testTargets = t . tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots

% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

Figure B.2: ANN model built using MATLAB part 2

Appendix C

ANN results plots

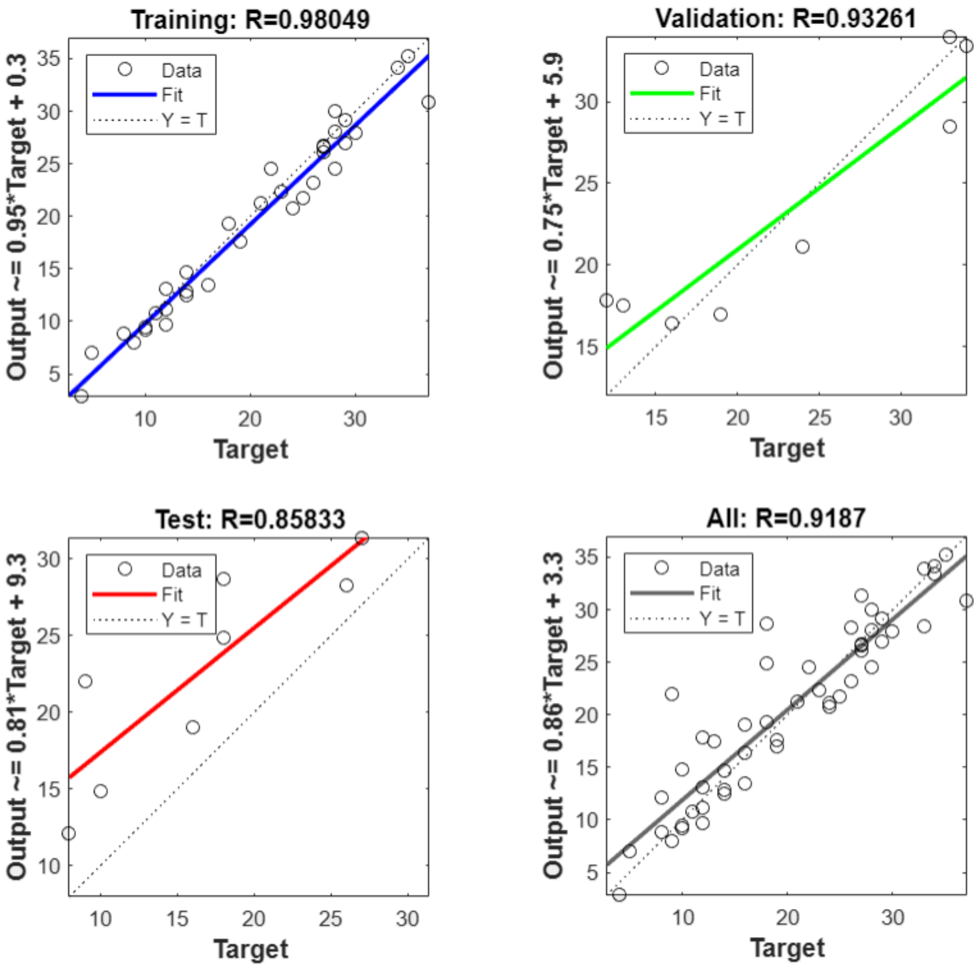

Figure C.1: UTS Training, Validation, Test, and All Plots

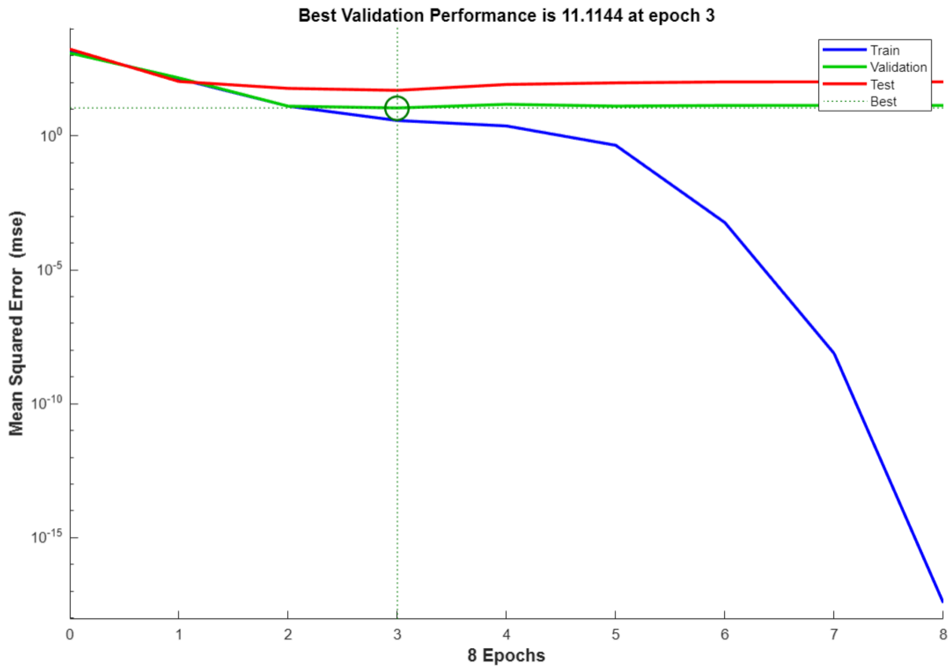

Figure C.2: UTS Performance

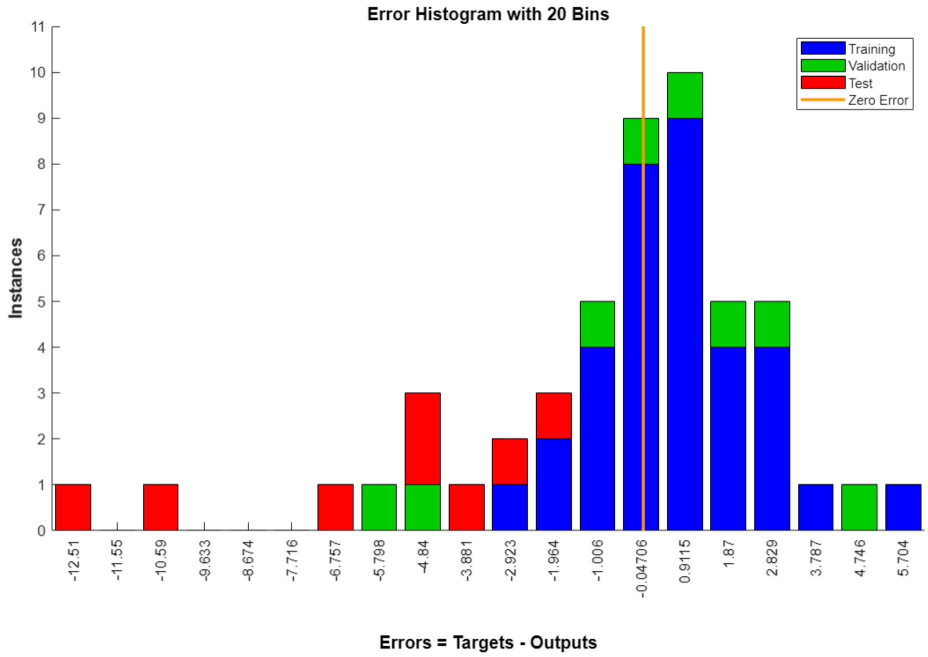

Figure C.3: UTS Histogram

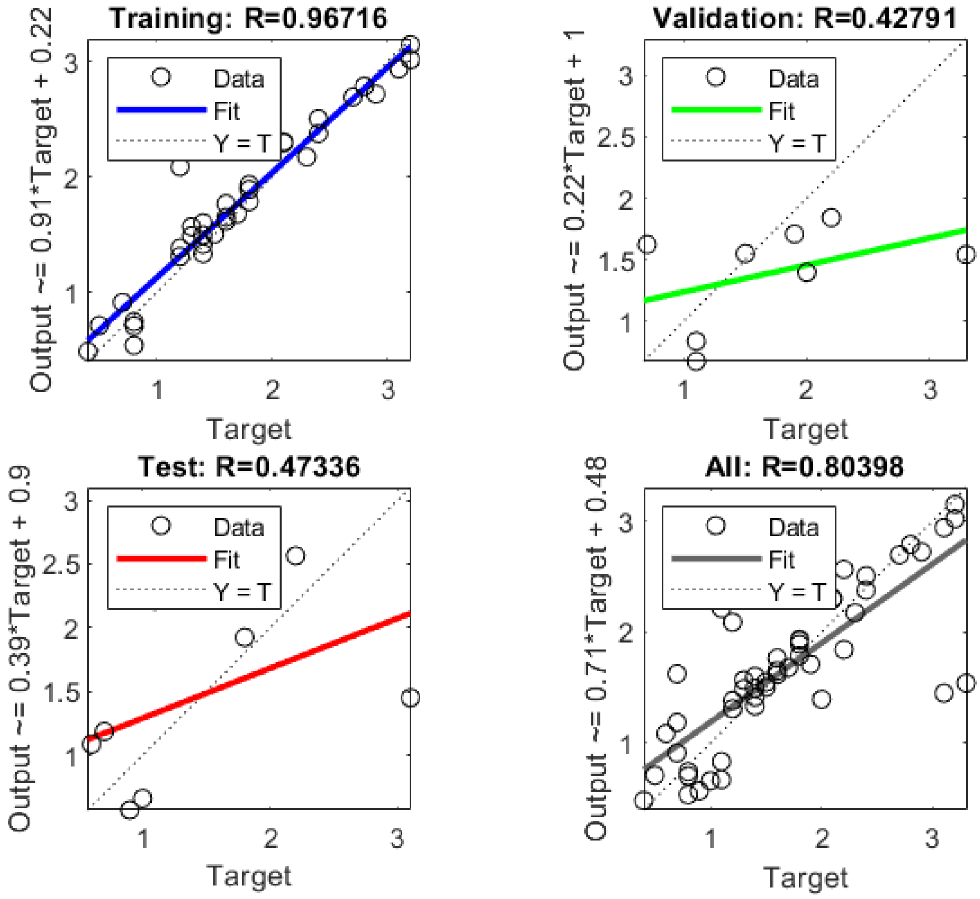

Figure C.4: Elongation Training, Validation, Test, and All Plots

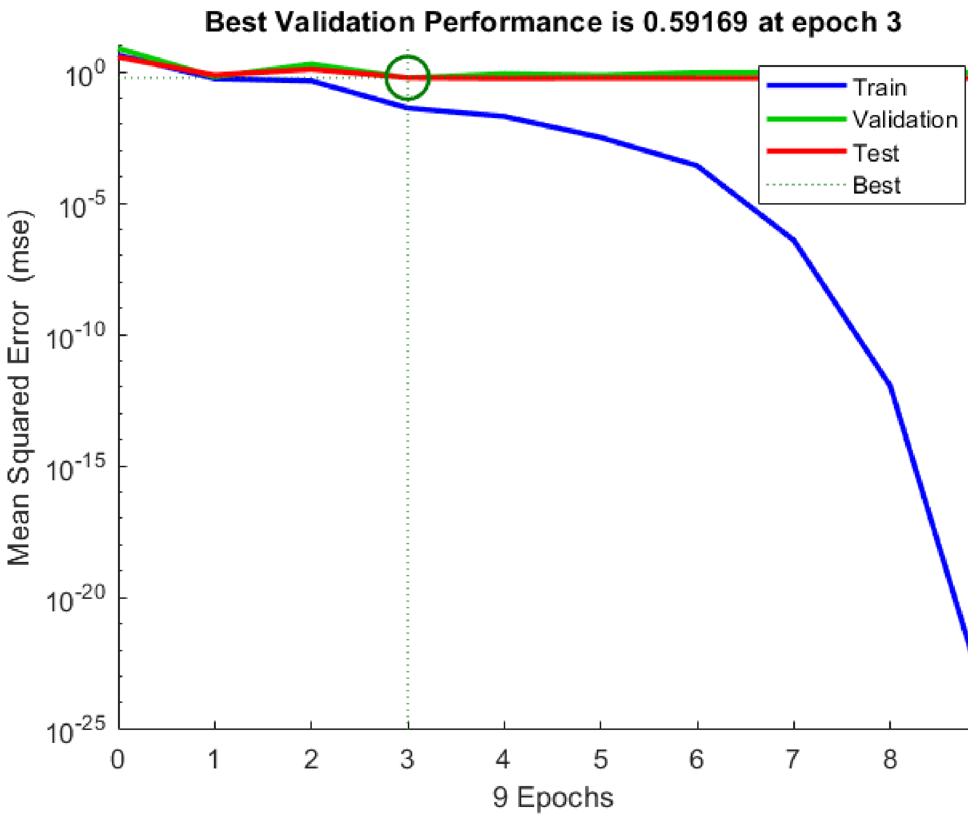

Figure C.5: Elongation Performance

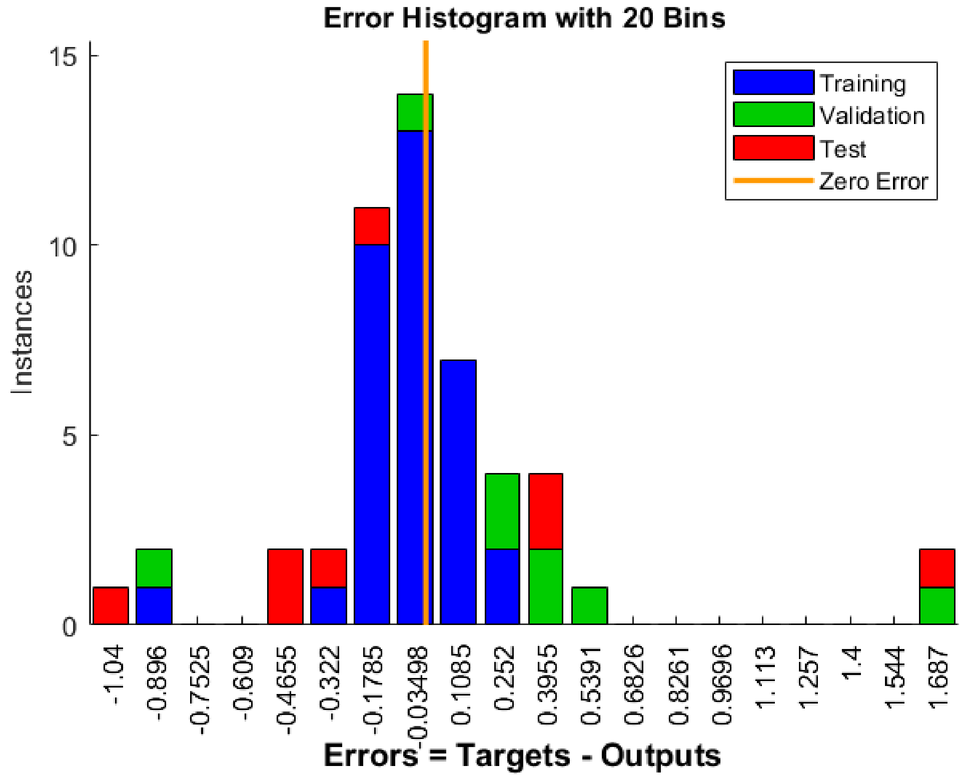

Figure C.6: Elongation Histogram

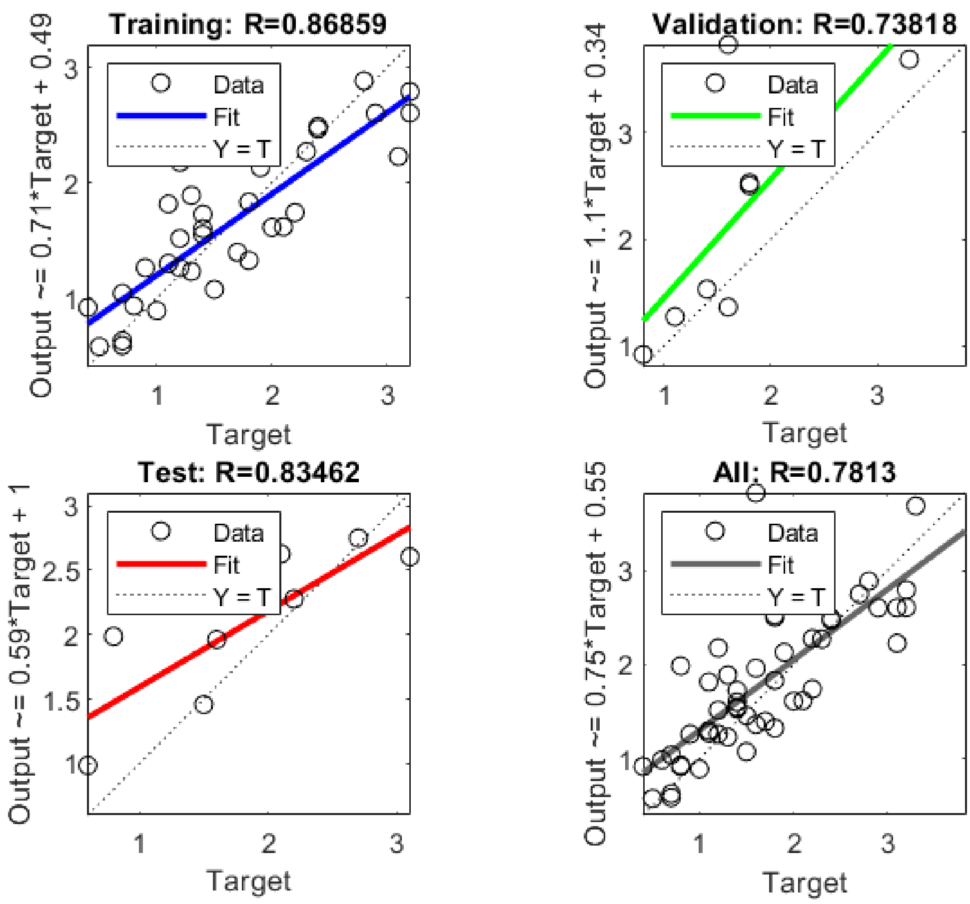

Figure C.7: Roughness Training, Validation, Test, and All Plots

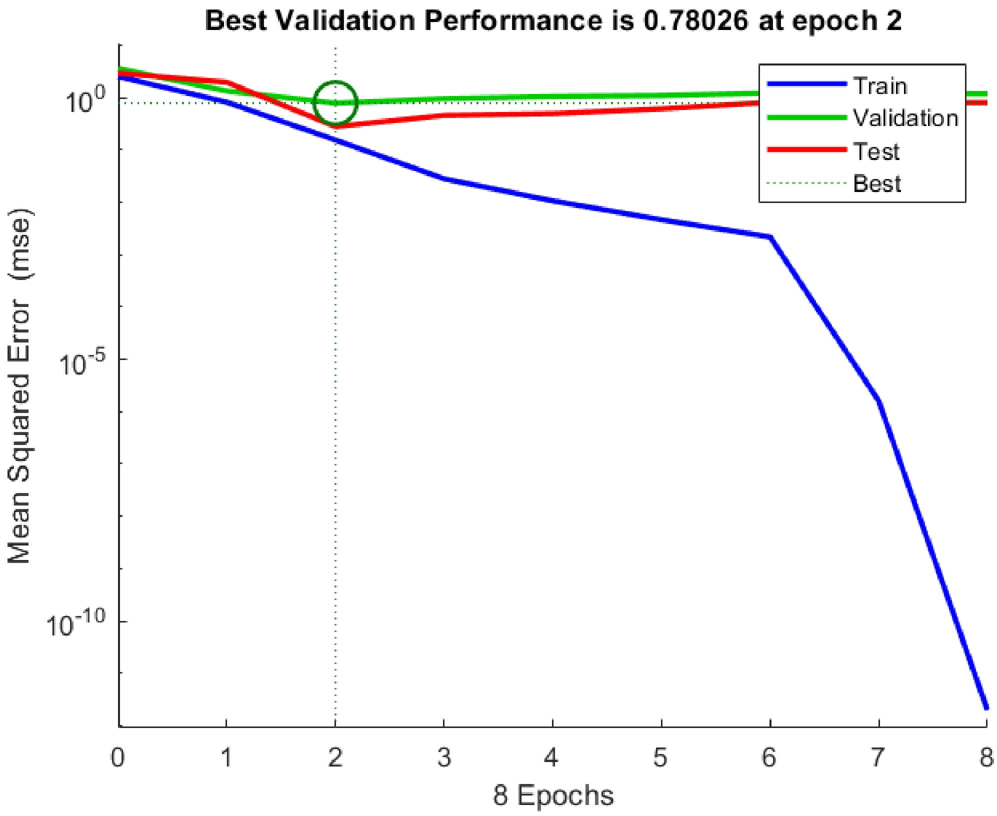

Figure C.8: Roughness Performance

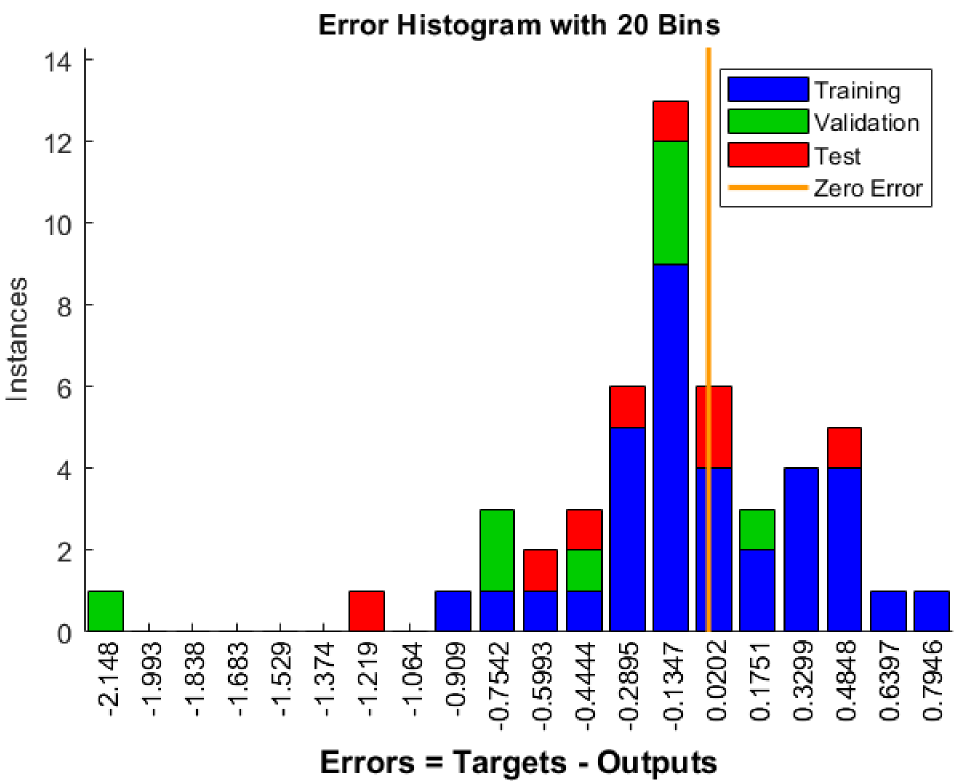

Figure C.9: Roughness Histogram