20090325

Supervisor: Dr. Daniel Karapetyan
Module Code: COMP3003

2021/22

Prediction of Metal Coating Properties for Greener Air Travel

Submitted 25 April 2022, in partial fulfilment of
the conditions for the award of the degree **BSc Computer Science with Artificial Intelligence**.

20090325

School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature _____

Date 25 / 4 / 2022

Abstract

The application of Thermal Barrier Coatings onto components of modern jets is the mainstream approach used these days to provide modern jets with extra durability, heat insulation, better cooling features and improved fuel efficiency. However, the cost-intensive nature of developing new coatings is one of the main hurdles that forbids the discovery of innovative breakthroughs within the Thermal Barrier Coating industry. An economical solution to this issue has to be done as these thermal coatings indeed play an important role in increasing fuel efficiency of modern jets that can gradually lead to greener air travel by emitting lesser greenhouse gasses into the environment. This dissertation proposed a novel Machine Learning approach to predict the in-flight particle characteristics of the Atmospheric Plasma Spray Process. The proposed Machine Learning approach assembles the usage of Gaussian Kernel Regression and Artificial Neural Network to establish a model evaluation pipeline. In comparison with existing literature, the proposed Machine Learning approach has shown great capability to avoid data leakage that resulted in a closer depiction to the model's true performance. As a result, the proposed Machine Learning approach will facilitate researchers from the Surface & Coating industry to better determine the practicability of the subjected input parameters before conducting physical experiments of Atmospheric Plasma Spray processes that are cost ineffective and time consuming.

Acknowledgements

I would like to express my deepest gratitude towards my supervisor, Dr. Daniel Karapetyan, for his constructive guidance and constant encouragement throughout the development of this project. I would also wish to acknowledge Prof. Tanvir Hussain and his team from the Department of Coatings and Surface Engineering for their continual support in the Chemical Engineering aspect of this project. Additionally, I would like to thank Prof. Ender Ozcan, Dr. Grazziela Figueredo, Dr. Geert De Maere and Dr. Isaac Triguero from the School of Computer Science of University of Nottingham for their expertise in the field of Machine Learning. Finally, I would like to thank my friends and family for their abiding faith in my ability to succeed in this dissertation.

Contents

Acronyms

ANN Artificial Neural Network

APS Atmospheric Plasma Spray

BP Back Propagation

BR Bayesian Regularization

COP26 2021 United Nations Climate Change Conference

ELM Extreme Learning Machine

FoE Faculty of Engineering

GKR Gaussian Kernel Regression

GPU Graphics Processing Unit

HVOF High Velocity Oxygen Fuel

ICAO International Civil Aviation Organization

IDE Integrated Development Environment

LM Levenberg Marquardt

MAE Mean Absolute Error

ML Machine Learning

Mt Metric Tons

RB Resilient Backpropagation

RBF Radial Basis Function

ReLU Rectified Linear Unit

ResNet Residual Neural Network

RMSE Root Mean Square Error

S&C Surface and Coating

SD Standard Deviation

SPS Suspension Plasma Spray

SRM Structural Risk Minimisation

SVM Support Vector Machine

TBC Thermal Barrier Coating

UoN University of Nottingham

List of Tables

List of Figures

Introduction

In 2020 alone, the International Civil Aviation Organization (ICAO) estimated a range between 216 and 239 Metric Tons (Mt) of fuel being consumed by the aviation industry, discharging approximately 682 to 755 Mt of CO_2 emissions into the environment [5]. It can be seen from the recent COP26 summit, the issue of climate change caused by the aviation industry has drawn much attention across the globe. Therefore, it is obvious that radical transformation must be done in the aviation industry to tackle climate change resulting from excessive greenhouse gases emission [6].

These days, modern jets are equipped with turbines of greater thrust that allow aircrafts to propel at an even faster speed. To generate greater thrust, high pressure turbine blades located after the combustion chamber of a jet engine have to constantly operate under an extreme temperature where super alloy used to construct the blades would not sustain the amount of regular exposure to extreme heat [7]. To tackle this issue, application of Thermal Barrier Coatings (TBCs) has been introduced to enhance the durability and performance of components in jet engines [8]. Previous studies suggested the application of TBCs onto jet engine's components such as turbine blades can prevent these blades from melting during operation as TBCs are proven to provide better thermal insulation and cooling features to the blades, allowing them to operate under optimal temperature hence increasing fuel efficiency of the turbine to create a greater amount of thrust [9].

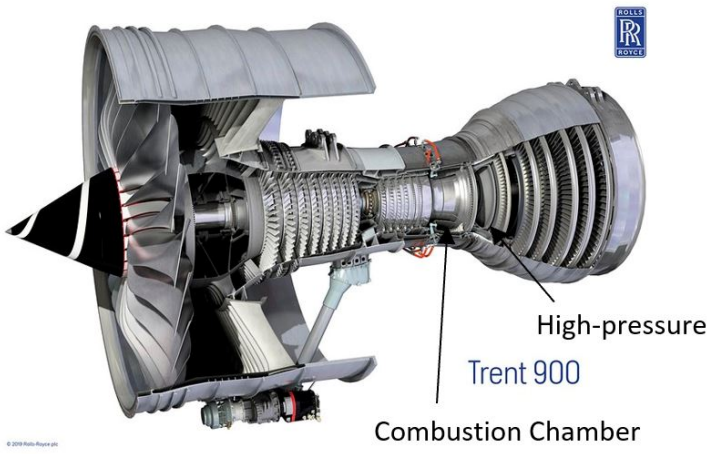

Figure 1.1: Cutaway view of Rolls-Royce Trent 900 jet turbine

There are two methods of applying TBCs, namely Atmospheric Plasma Spray (APS) and Suspension Plasma Spray (SPS). The APS has feedstock material of powder form that has to be transported via carrier gas into the plasma jet. These powder particles are heated and accelerated towards the turbine blades (substrate), forming layers of coating as thin as a strand of hair as it solidifies [1]. In contrast, the SPS has feedstock material in the form of suspense state but shares a similar coating formation mechanism as the APS. However, performances of TBCs produced using both methods are heavily dependent on the behaviours exerted by the projected in-flight coating particles. With that being said, it can be deduced

that input features of the plasma jet play an important role in producing high-performance TBCs and therefore have to be monitored closely.

Within this project, APS technique will be the primary experimental setting throughout the project workflow. This project will focus on developing a model that has been equipped with appropriate Machine Learning (ML) techniques where it will be used to study the relationship between input and the output features of the plasma jet; which is of great interest of the Faculty of Engineering (FoE) of University of Nottingham (UoN). One of the main reasons behind this collaboration with FoE is that they are expecting the resulting ML model from this project to be beneficial for them in the long run. It will be a cost-effective option to have an ML model that predicts in-flight particle characteristics of an APS process as it is uneconomical and cost inefficient for conducting physical experiments to obtain these output particle characteristics. In other words, this "high risk low reward" scenario is indeed one of the biggest challenges being faced by coating researchers in the industry to conduct research that may potentially lead to breakthroughs. With that being said, the motive of this project is to provide researchers with an ML model where the model will aid researchers in decision-making by generating a series of predicted output characteristics of in-flight coating particles, in which researchers will be able to make a better judgement based on the predicted output characteristics by the ML model, only then to decide whether to conduct physical experiments using similar settings to input values that has been fed into the ML model. This approach is a crucial step towards greener air travel because researchers can effortlessly filter settings of input features that contributed to non-promising output characteristics of in-flight coating particles in a cost-effective and efficient manner, hence producing a TBC that has better thermal insulation and cooling feature.

1.1 Aims and Objectives

The aim in this project is to develop a predictive machine learning model based on a small dataset regarding Atmospheric Plasma Spray that helps researchers in identifying output parameters of in-flight particle characteristics that possess higher chances of being effective before being implemented in physical experiments.

The key objectives of the project are:

1. To investigate previous attempts of machine learning models that have been made by researchers to predict the output parameters of these in-flight coating particles.
2. To conduct research study on data pre-processing methodology that expands the current size of dataset by using kernel filters where noise will be added into the dataset, allowing the creation of a larger dataset to be used to train the machine learning technique.
3. To implement a range of machine learning techniques that can perform effectively on a small dataset.
4. To implement hyperparameter optimization method for machine learning techniques implemented, retrieving the best estimates of scoring metrics out of each machine learning technique.
5. To analyse and evaluate trained machine learning techniques that have been implemented; from then on, the project will focus on the best predictive model followed by an in-depth review on the performance of the selected model.

Related Work

Recent years, much attention has been given into the industry of Surface and Coating (S&C) technology. This is due to the imperative demand for newer coating technology by the industry that aims to advance the capabilities of coatings to function in harsh environments posed by challenges such as extreme temperature, friction, and erosion [10]. There exist two contributing factors that resulted in a high cost of APS operation, namely the cost of direct material (coating chemical) and the cost of labour that includes the preparation, inspection, and cleaning of the thermal booth [11]. Being cognizant of this fact, it is understandable that there is hardly much data of APS runs being made available to the public [12].

Bobzin et al. suggested that the complex nonlinear interdependencies relationship that exist between the input and output features of the APS process encourages the quantification of these complex relationships to be reinforced by computer-aided methodologies as this will certainly help in enhancing the reproducibility of the process [13]. Two different ML techniques, namely the Support Vector Machine (SVM) and Residual Neural Network (ResNet) were presented to make predictions on the APS process. In recent years, the SVM approach proposed by Vapnik [14] has gained popularity among researchers in the field of S&C technology [15, 16]. This is due to the unique property present within the SVM approach, having an ability to generalise without being dependent on the complexity of the input space. Meaning, given the case where a high computational complexity input space is being supplied into the SVM, this approach will still generalise well and produce a result that is of a relatively high prediction rate. On the other hand, ResNet that has been suggested by He et al. has gained much reputation recently especially in solving high complexity problems in the field of image processing [17]. A reason for this approach to be widely implemented is because it addresses the notorious problem of vanishing and exploding gradients that were commonly found during the training of an Artificial Neural Network (ANN). However, a comparison between the two techniques has shown that results generated by ResNet have a slight enhancement in performance of predicting in-flight particle properties over the SVM.

One such paper proposed by Liu et al. developed an ML model based on an ANN architecture that targets specifically onto the high velocity oxygen fuel (HVOF) spray process [18]. Unlike the APS process, the HVOF spray process utilises the mixture of fluid fuel and oxygen where this mixture, when being combusted, will act as a carrier that channels the feedstock material onto the substrate's surface. Nevertheless, there exist similarities between HVOF spray process and APS process in terms of dimensionality and computational complexity between the input and output space present within both experimental settings. Besides, variation on the training function of the ANN proposed by Liu et al. has been made to examine how each training function affects the performance of the model [18]. As a result, only the Levenberg Marquardt (LM) optimization and Bayesian Regularization (BR) have managed to exhibit notable performances, with the R-value of 0.99938 and 0.99974 respectively. Although BR has a better R-Value performance, LM optimization has still been selected as the ideal training function because the main intention of this paper was to achieve an accurate ML model that can make predictions within a shorter amount of training time. According to this paper, BR requires approximately 2-3 times more training duration. Therefore, the slight difference in performance can however be neglected as training

	Mean Absolute Error (MAE)	
	Average (SD)	**Best**
ANN trained with unexpanded dataset	0.0639 (0.0082)	0.0472
ANN trained with expanded dataset	0.0072 (0.0017)	0.005

Table 2.1: Performances of ANN models trained on expanded and unexpanded dataset in terms of MAE [1]

time in this case has been prioritised for the selection of ML model with the best setting.

Kanta et al. suggested an implementation of ANN model that predicts the in-flight particle characteristics of an APS process based on their corresponding plasma spraying parameters such as arc current intensity, total plasma gas flow and hydrogen content [19]. This paper focuses exclusively on investigating the underlying relationships between the microstructures of the resulting coating from the APS process and the plasma spraying parameters.

Choudhury et al. in 2011 proposed a data expansion method using a nonparametric kernel regression technique that aims to tackle the issue of limited dataset of APS runs for the training of ML techniques [1]. This proposed technique has claimed to produce an expanded dataset by increasing the number of data instances from 16 to approximately 19 times the unexpanded dataset. This interpolation of the dataset is a result from the application of the Parzen-Rosenblatt estimate where a regression function that best fits the data points is produced. A Gaussian Kernel will be applied onto these interpolated data points, where each data point will be assigned with a weight based on its distance from the original data point. Followed by the computation of Nadaraya-Watson Kernel Regression where an estimated output feature value will be produced. Besides, the paper also suggested the implementation of 2 distinct ANN models with different configurations to predict in-flight particle characteristics, where each model is required to cater either the unexpanded or expanded dataset. According to the paper, the ANN model being trained with unexpanded dataset has its best performance when configuration of the model is of five and four neurons in first and second hidden layers resulting in 0.0472 of Mean Absolute Error (MAE) with a corresponding R-Value of 0.9233. In contrast, ANN model trained with the expanded dataset is best when configuration of the model is of nine and eight neurons in the first and second hidden layers, producing MAE of 0.005 with an R-Value of 0.9923. Table 2.1 shows the average and best performances of both ANN models trained using the unexpanded and expanded dataset in terms of MAE. It is obvious that ANNs trained with the expanded dataset has a better generalisation ability when compared to ANNs trained with unexpanded dataset. The proposed methodology of dataset expansion within the paper is one of the key motivations that inspired this dissertation as it will certainly be beneficial to the training of ML techniques using a small-scale dataset.

Choudhury et al. (2013) studied the usage of Single Hidden Layer Feed Forward Neural Network in the prediction of in-flight particle characteristics of the APS process [2]. This approach is nonetheless a modification of the study being conducted back in 2011 [1] where the ANN model of two hidden layers has been simplified into a single hidden layer. The expanded and unexpanded dataset generated in the previous study [1] has again been reused within this paper. However, there exists some changes of motivation when compared to the previous paper. This paper mainly focuses on implementation of an online ML model that will be incorporated into a real-time thermal spray control system that prioritises learning speed of the model. The authors also suggested the substitution of standard Back Propagation (BP) training algorithm with Extreme Learning Machine (ELM) algorithm [20]. Moreover, this paper has also included several other training functions into comparison such as LM optimization, Resilient Backpropagation (RB) algorithm, and BR algorithm. Out of the four algorithms, the BR training

	ELM	LM	RB	BR
Correlation coefficient (R-Value)	0.9950	0.9263	0.8441	0.9972
Generalisation error (Mean Absolute Error)	0.0070	0.0191	0.0205	0.0043
Training time (seconds)	1.78	31.95	2.64	2730.97

Table 2.2: Performances of different training function on ANN model with the expanded dataset [2]

function has the best performance with R-Value of 0.9972. However, ELM has been the preferred choice for this real-time thermal spray control system as it offers a shorter training time yet provides a good generalisation performance. Refer Table 2.2 for comparison of ANN performances using different training functions.

Based on papers found in the literature, the most common ML approach to study in-flight particle characteristics of the APS process is the ANN technique [21, 22, 23, 24]. Despite having most papers to concentrate on implementation of ANN techniques onto different variation of feedstock materials such as zirconia coatings [25], WC-12% Co coatings [24], 8 mol% yttria stabilised zirconia electrolyte coatings [21] and grey alumina coatings [26], results generated from these implementation of ANN techniques have still been considered a great success. This is because ANN has great competence in modelling the underlying complex nonlinear interdependencies of input and output space within the APS process. In other words, an ANN with an optimised weight matrix will be capable of generalising intricate relationships of input and output parameters during its training process. As a reflection of the literature review, application of ANN technique will be adopted into the dissertation to investigate the complex problem of predicting in-flight particle characteristics of an APS process.

Being inspired by Choudhury et al., implementation of the ANN technique within this dissertation will be based entirely on the authors' work using the supplied dataset from the papers [1, 27]. Both datasets have been included in Appendix A and Appendix B. Nevertheless, in-depth research into the papers has revealed a major flaw that potentially leads to an overly optimistic ANN model. Due to the limited number of data instances within the dataset, the authors first applied a Kernel Regression Method that aims to interpolate the dataset by adding noise into existing data instances. Later, normalisation of the dataset will be carried out based on the lower and higher physical limits of each processing parameter, followed by splitting of the dataset into train, validation, and test sets. This flow of data pre-processing execution is illicit as it will certainly introduce data leakage into the training of ML technique, scilicet, using data which is not part of the training set within the training phase of the ML technique [28].

In summary, the work presented within this dissertation builds on previous research to explore applications of ANN technique that aims to predict in-flight particle characteristics of APS process. The main contribution of this dissertation is to reveal any underlying flaws within previous research by Choudhury et al. [1], then rectifying any imperfections that have been discovered. The impact of this dissertation that addresses these existing limitations is to aspire more researchers from both Computer Science and S&C discipline to collaborate in research work that focuses on the implementation of ML techniques in the APS process as it is evident that there is a lack of research concerning this topic. With more collaboration between the two disciplines, advancement in the implementation of ML techniques in the APS process is guaranteed.

Description of the Work

The description of the work section has been split into three subsections where each subsection serves the purpose of revealing one of the three main aspects to be achieved in this work.

3.1 Developing ML Model

The first aspect of this work is to develop an ML model that is based on existing state-of-the-art ML techniques to predict in-flight particle characteristics of coating chemicals within the APS process. This developed ML model will be adopted by researchers in FoE where it will facilitate them in decision making by means of computational power of Artificial Intelligence. With this implementation, researchers of FoE will be able to determine the practicability of the subjected input parameters that will be passed into the plasma jet, without needing to carry out physical APS processes that are cost-ineffective and time consuming.

3.2 Establishing Data Expansion Technique

Secondly, it is to establish a data expansion technique that can be applied by researchers from the FoE to expand the size of their dataset that may lead to an ML technique which has a better generalisation ability and performance in making predictions. As mentioned previously, datasets from the S&C technology that have been made available to the public are often in a limited number of data instances due to its high operational cost of conducting physical experiments of the APS process [11]. Also, there exists a diversity of coating chemical compositions that are frequently used in the field of S&C technology that targets specific functionalities. This distinctive nature of each chemical composition caused complications during attempts to combine datasets originated from different sources as these datasets generally have non-identical input and output parameters. Besides, these parameter values of the APS process are heavily coupled with each other; the slightest variation on any of the input parameters will affect output characteristics of these in-flight particles. With that being said, it is obvious that exploration into data expansion techniques will indeed be a crucial step for the implementation of a decent ML model that predicts the in-flight particle characteristics of the APS process as these data expansion techniques will result in a larger dataset that certainly allow ML techniques to better recognize underlying patterns found within the dataset.

3.3 Addressing Potential Issues in Previously Published Work

As aforementioned, the work presented within this dissertation has been built on previous research. The third aspect to be achieved in this dissertation is to rectify potential issues that exist specifically within

Choudhury et al. [1].

3.3.1 Potential Issues

Replication of the work from Choudhury et al. [1] was an arduous effort. This is due to the ambiguity present within the literature where most of the key parameter settings of the equipped framework from the literature are not provided. For instance, parameter settings like bandwidth for the Gaussian Kernel Regression (GKR) technique that has been implemented to expand the dataset of the APS process are not provided within the literature. Therefore, manual testing by means of trial and error must be done on each framework using viable parameter settings that could potentially lead to promising outputs.

In addition, there exist several misconceptions concerning erroneous applications of data pre-processing techniques within this literature. Referring to a direct quote from Choudhury et al. [1]:

> "The input processing parameters and the output in-flight particle characteristics require a linear transformation before being used ... X_{NORM} stands for the normalised parameter value, X_{MAX} and X_{MIN} are the maximum and minimum possible values of the parameters based upon their physical limitations of the process, not from the experimental sets." [1]

It implies that data normalisation has been performed onto the entire dataset based on a global minimum and maximum value. This approach is, however, inappropriate as it introduces the occurrence of data leakage into the training of an ML technique. Data leakage can be defined as a condition where there is an occurrence of information revealing from sources outside the training set that gives an unrealistic advantage to the performance of an ML model [29]. In other words, normalisation using global minimum and maximum value has somewhat given the ML technique an access to insights of the global distribution present within the entire dataset. With that being said, it is obvious that the occurrence of data leakage increases the likelihood of resulting in an overly optimistic model that appears to be performant during training but fails intensely during production due to its poor ability to generalise when encountering unseen data instances [30].

Another violation that has been found within Choudhury et al. [1] is the usage of data expansion technique by means of GKR before splitting of dataset into train and test set. Estimation of output feature value by this nonparametric kernel regression method is made depending on a weight that represents the distance between the interpolated data points and their corresponding original data point. In other words, these weights can be depicted as the significance level of the interpolated data points based on a normal distribution curve where the mean of the curve is represented by the original data point [31]. With that being said, it is evident that these interpolated data points estimated by the GKR carry information associated to their corresponding original data point. Therefore, it will be unfair for such a process to be applied onto the dataset before dataset splitting as this violation will certainly introduce the concept of data leakage into the training of the ML technique [32]. A detailed discussion on the mechanism behind GKR will be made within the Methodologies section of this dissertation.

In essence, data leakage is often termed as the silent killer in most ML solutions these days that resulted in overly optimistic predictive ML models. The process of identifying the presence of data leakage is tedious and usually be discovered only when the ML model has been put into production. However, there exist various schemes and preventive measures that have to be followed closely to avoid the occurrence of data leakage when implementing an ML technique. These schemes will be elaborated in the section below that aims to resolve the issue of data leakage found within the paper by Choudhury et al. [1].

3.3.2 Rectification of Potential Issues

Rectification work has been conducted within this dissertation to attest the suggested methodological flaws that lie within the implementation of Choudhury et al. [1] that provoked the development of an overly optimistic model.

To address the aforementioned issue of data normalisation, minimum and maximum value for the Min-Max normalisation has to be acquired exclusively from the training set instead of using a global minimum and maximum value of physical limitations of each parameter in the APS process. Kuhn & Johnson (2021) emphasised that any statistical approach of data scaling or transformation has to be estimated from the training set before being applied onto the test set [32]. With that being said, it is apparent that splitting of dataset into train and test set has to be done before conducting any normalisation of dataset. The outcome of this practice is to prevent occurrence of data leakage where the training dataset learns information that lies within the test set that eventually allows the ML technique to access the global distribution present within the entire dataset [33].

The second misconception found within Choudhury et al. [1] is the usage of data expansion technique before splitting of dataset into train and test set. As discussed briefly regarding the nature of GKR, isolation of test sets from any sort of data preparation methodologies can avoid the happening of data leakage [33]. To counter this misconception, interpolation of the dataset using GKR has to be executed only after the splitting of the dataset, particularly onto the training set. Results of the rectification work on Choudhury et al. [1] will be closely discussed within the Evaluation section of this dissertation. A comparison between the replicated technique from [1] and the rectified work presented by this project will be made in the Design section of this dissertation.

As a summary of this section, the work presented within this dissertation aims to correctly incorporate an ML pipeline that combines bits and pieces of pipeline elements ranging from data preparation methodologies to ML techniques. Researchers of the FoE will only be required to pass as input the dataset containing APS runs of their designated chemical composition generated from past laboratory experiments into the ML pipeline. An automated data interpolation technique will be applied onto the dataset, followed by the training of ML technique using the interpolated dataset. Performance of the ML model will be generated and evaluated based on error metrics such Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). With these error metrics, researchers will be able to obtain insights to the reliability and effectiveness of the resulting ML model.

Methodologies

4.1 Error Metrics

Performance of the final regression ML model will be evaluated in the form of error metrics by measuring the proximity of predicted values to the target values.

4.1.1 Mean Absolute Error (MAE)

MAE is the calculation of absolute difference of magnitude between predicted value of a data instance and its corresponding true value, then performing an average onto the summation of absolute differences of all data instances. However, this approach of error calculation is not sensitive to outliers present within the dataset as it tends to treat every data instance equally by taking the average error rate of the entire dataset. From Eq. 4.1, it shows the formula for calculating MAE with n representing the total number of data instances within the dataset.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |Predicted\ value - True\ value| \tag{4.1}$$

4.1.2 Root Mean Square Error (RMSE)

In the context of ML, RMSE is one of the most frequently used model evaluation methods. RMSE can be computed by calculating the Euclidean distance between the predicted value and true value for each data instance, taking the norm of Euclidean distance of each data instance, followed by the computation of mean Euclidean distances, and applying a square root onto the mean. However, the presence of outliers within the dataset will heavily affect the computation of RMSE. Therefore, to obtain a better overview to the performance of the ML model with dataset containing outliers, it is best to use MAE as the error metrics calculation. From Eq. 4.2, it expresses the formula of RMSE where n represents the total number of data instances within the dataset.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} ||True\ value - Predicted\ value||^2}{n}} \tag{4.2}$$

4.1.3 Pearson's Correlation Coefficient (R-Value)

The R-Value, often known as the Pearson's Correlation Coefficient represents the measurement of degree of association between two continuous variables of x and y. R-Value returns a value ranging between -1 to 1. The value of -1 signifies a full negative correlation where an increase in either one of the variables will

Injector stand-off distance		
Values (mm)	**Neuron 1**	**Neuron 2**
6	0	1
7	1	0
8	1	1

Injector diameter		
Values (mm)	**Neuron 1**	**Neuron 2**
1.5	0	1
1.8	1	0
2.0	1	1

Table 4.1: Possible neuron values that represent both categorical features

cause a decrease in another variable. However, it is vice versa for the case for R-Value of 1. Whereas an R-Value of 0 represents there exists no correlation between the two variables. Measurement of R-Value will be helpful to understand the degree of effectiveness of the predictions made by the trained ML model in fitting its corresponding value within the dataset. From Eq. 4.3, it shows the computation of R-Value where n represents the total number of data instances within the dataset; x and y representing individual data points from the dataset.

$$R\text{-}Value = \frac{\sum_{i=1}^{n}(x_i - mean(x))(y_i - mean(y))}{\sqrt{\sum_{i=1}^{n}(x_i - mean(x))^2 \sum_{i=1}^{n}(y_i - mean(y))^2}} \tag{4.3}$$

4.2 Data Analysis

The data analysis section has been divided into two subsections that illustrates various techniques of data pre-processing and data utilisation.

4.2.1 Data Pre-processing

There are two distinct sources for obtaining datasets that have been used within this project. The unexpanded dataset, tabulated in Appendix A, was retrieved from a paper by Choudhury et al. [1], whereas the expanded dataset, tabulated in Appendix B has been obtained from a thesis by Choudhury (2013) [27]. Both datasets have the same dimensionality of a total of six feature variables and three target variables. Moreover, both papers share the same author that focuses on an identical research motive, that is to apply ANN technique into the prediction of in-flight particle characteristics of APS process. Therefore, it is reasonable for concepts and findings from both research papers to function bilaterally.

A good extent of work has been made on analysing both datasets obtained. Choudhury (2013) suggested that two out of the six input features of the APS process, namely the injector stand-off distance and injector diameter can be represented categorically [27]. This is because the equipped plasma jet offers only a certain pre-set options for both input features stated above. In this case, the pre-set options (all represented in millimetres) for injector stand-off distance are 6, 7, and 8 whereas for injector diameter, they are 1.5, 1.8 and 2.0. It can be seen that both features share the same characteristics, having a nominal nature where there exist no particular orders between values [34] and they can be represented by three discrete categories. With that being said, Choudhury et al. proposed an approach to split each feature into two distinct neurons where each neuron will hold either a value of 0 or 1 [1]. This proposed approach has resulted in a dataset with a total of 8 input features. Table 4.1 depicts the conversion of both features into neurons that have been previously discussed.

In regression problems, it is usual to have different features to be measured on a different scale of range values. This different scale of range values often induces ML technique the effect of over-reliance onto features which have higher values. In other words, it is not an ideal case to have an ML technique to heavily rely on a single feature that drives the performance of the model. To counter this issue, Witten

et al. (2017) suggested the normalisation of features into a range value within 0 and 1, promoting an equal importance over all features [35]. With that being said, the issue of features with higher values that dominate the learning process of the ML technique can be eradicated with the application of data normalisation. In this project, Min-Max Normalisation has been implemented to rescale feature values based on the minimum and maximum value found within the feature. This Min-Max Normalisation guarantees every feature value to be mapped onto an equivalent ratio between 0 to 1 that corresponds to their original value [29]. The formula for Min-Max normalisation can be expressed as follows:

$$x_{\text{Norm}} = \frac{x - x_{\text{Min}}}{x_{\text{Max}} - x_{\text{Min}}} \tag{4.4}$$

4.2.2 Data Utilisation

There are a few key aspects to consider when evaluating a trained ML model, that is its stability in carrying out prediction when being introduced with unseen data, and its ability to generalise without picking up excessive noise introduced by the dataset. Both these aspects are rather challenging to be resolved within this project, given that both datasets provided are of a very small size [1, 27].

In response to this challenge, cross validation has been introduced to enhance the evaluation process of the trained model. This robust scheme randomly divides the dataset into k folds of approximately equal sizes, then taking the first fold as the validation set where fitting of ML technique will be carried out on the remaining *k-1* folds [36]. The trained model will then be evaluated based on the hold-out fold. This algorithm repeats the procedure for a total of k times until every remaining fold has once been treated as the validation set, resulting in a total of k number of model evaluations. These evaluations will be averaged to obtain the mean performance of the model, at the same time utilising every bit of the dataset that aims to result in a less biased and less optimistic model evaluation when compared to the traditional train test split of dataset [37].

4.3 Data Interpolation Technique

4.3.1 Gaussian Kernel Regression (GKR)

Gaussian Kernel Regression (GKR) is one of the most widely used kernel regression techniques present these days. The underlying mechanism that caused GKR to gain its popularity in the field of ML is that it incorporates prior knowledge gathered from Gaussian Kernels together with a series of probabilistic measures to produce its prediction [38]. Within this project, this GKR algorithm plays a prominent role in dataset expansion that aims to facilitate the training process of the ML technique.

In mathematical terms, a Gaussian Kernel can be expressed as below [1]:

$$K_h(x_j, Xi) = \exp^{\frac{-(x_j - Xi)^2}{2h^2}} \tag{4.5}$$

where h denotes the bandwidth value that acts as the smoothing parameter of the kernel; x_j represents an element from a series of existing domain values; Xi denotes the input data to be predicted by the GKR; and $K_h(x_j,Xi)$ that represents the resulting kernel function. A point that is worth mentioning is that bandwidth value, h plays a significant part in determining the quality of predictions made by the GKR algorithm. This is because the slightest variation on h will cause changes to the width of the Gaussian Kernel that will indirectly affect the kernel weights that are used to compute the prediction. Illustrated

in Fig. 4.1 are the effects of bandwidth on the Gaussian Kernel.

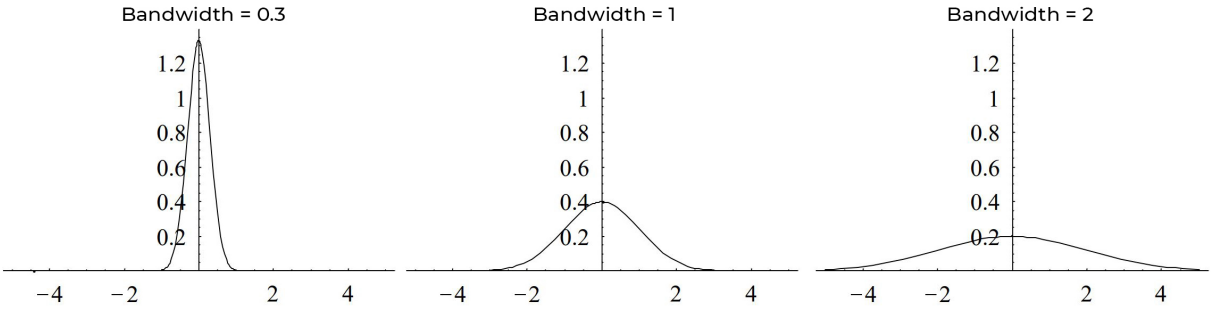

Figure 4.1: Effects of bandwidth on Gaussian Kernel

The Gaussian Kernel, $K_h(x_j,Xi)$ will be applied onto every element of the domain value, where each of these kernels will be adapted into the Nadaraya-Watson Kernel weighted average that eventually computes the prediction resulting from the GKR algorithm. Nadaraya-Watson Kernel weighted average can be expressed using Eq. 4.6:

$$y_j = \frac{\sum_{i=1}^{n} w_i K_h(x_j, Xi)}{\sum_{i=1}^{n} K_h(x_j, Xi)} \tag{4.6}$$

where n represents the number of elements present within the domain; w_i represents the weight that corresponds to each of the kernels; and y_j represents the resulting prediction from the weighted average.

Taking an example from the dataset, it can be observed that the distribution of values in the feature Current Intensity, x_1 and the distribution of values in target Average Particle Velocity, y_1 is as below:

$$x_1 = [350, 530, 750, 530, 530, 530, 530, 530, 530, 530, 530, 530, 530, 530, 530, 530]$$
$$y_1 = [242, 270, 278, 205, 241, 260, 264, 176, 179, 263, 252, 277, 270, 278, 265, 278]$$

By assigning the first element from list x_1, that is 350 to x_j in Eq. 4.5, a Gaussian Kernel can be represented as a normal distribution curve illustrated in Fig. 4.2, where the x-axis corresponds to a range of possible values with x_j being positioned at the centre of the curve having the left portion of the x-axis being values which are smaller than x_j and values on the right portion of x-axis being greater than x_j; and the y-axis represents the likelihood of the occurrence of values from x-axis [39]. In other words, the Gaussian Kernel serves as a kernel function that maps values from the x-axis to their corresponding probability of occurrence with x_j that sits at the centre of the curve being the value with maximum likelihood of occurrence. This process of formulating the normal distribution graph will be iterated for n times, where n represents the length of list x_1. This will result in a collection of normal distribution graphs illustrated in Fig. 4.3.

It is noticeable that most kernels overlap specifically at $x_j = 530$, resulting in only three obvious kernels being illustrated within Fig. 4.3 despite having a total of 16 feature values present within the list of x_1. This scenario can simply be explained by referring to x_1, as the list comprises only three unique values that is 350, 530, and 750 where most values happen to be 530. As for the Nadaraya-Watson Kernel weighted average, it can be depicted by the illustration shown in Fig. 4.4.

With the assumption that the value of Xi is 400 in Fig. 4.4, it can be seen that there are a few intersection points between the straight line of $Xi = 400$ and the Gaussian Kernels. Computation of Nadaraya-Watson Kernel weighted average can be done by performing a dot product on all the Gaussian

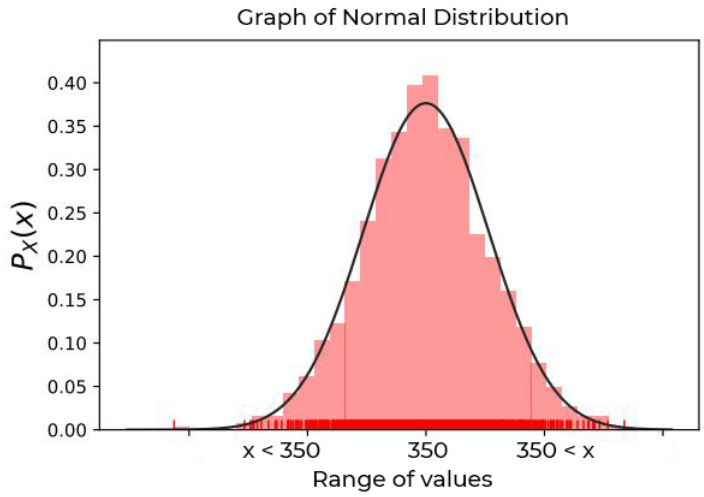

Figure 4.2: Illustration of Gaussian Kernel

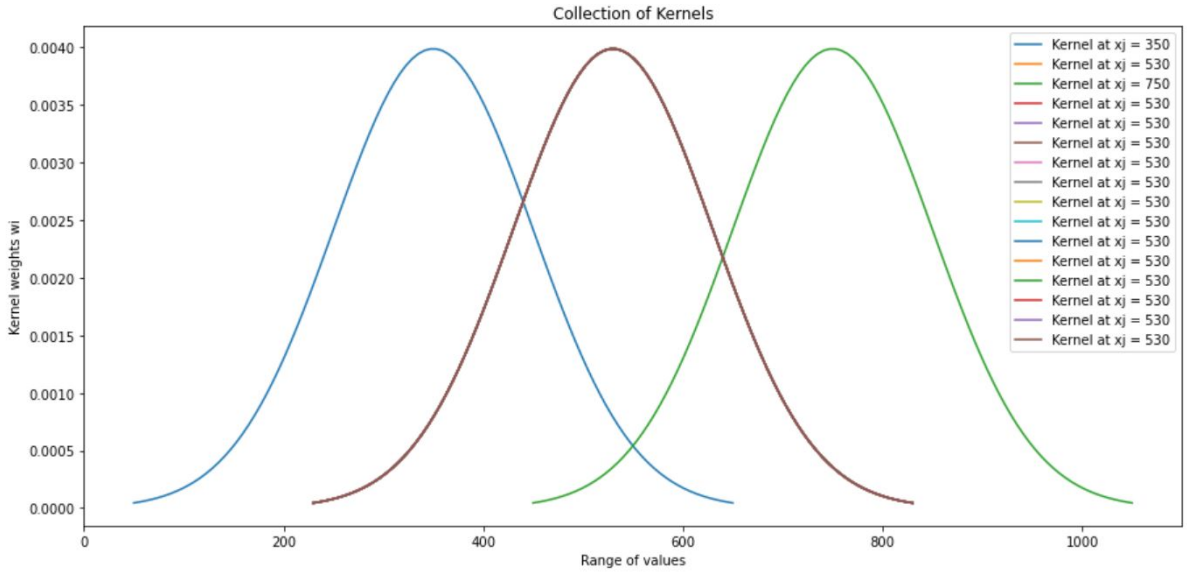

Figure 4.3: Collection of Gaussian Kernels

Kernels of each feature value within x_1 with their corresponding y value that originated from the list of target values, y_1, then averaging the dot product based the length of list x_1 being represented by n [40].

4.4 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is a biologically inspired ML technique that models the human-brain processes by means of computations and mathematics [41]. It has been proven that application of ANN in various research disciplines has led to great advancements [42]. A distinctive characteristic found in ANN is that establishment of complex relationships between the independent and dependent variables are made empirically [43]. Meaning, no assumptions regarding the mathematical representation have to be made for the establishment of these complex relationships. This ability of modelling complex relationships present within ANN is often comparable to the performance of human cognition, therefore causing a broad adoption of ANN technique into various research disciplines.

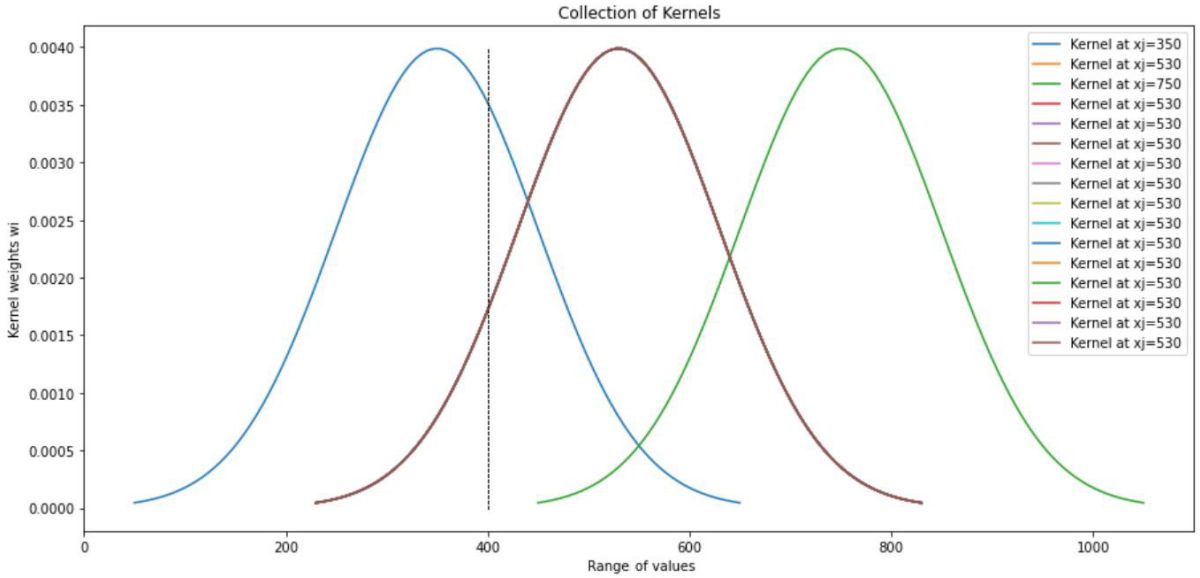

Figure 4.4: Collection of Gaussian Kernels with $Xi = 400$

The structure of an ANN comprises a collection of interconnected neurons that has been arranged into distinct layers. Each of these neurons acts as the fundamental information-processing unit that drives the ANN [44]. Fig. 4.5 illustrates the model of a neuron that forms an ANN.

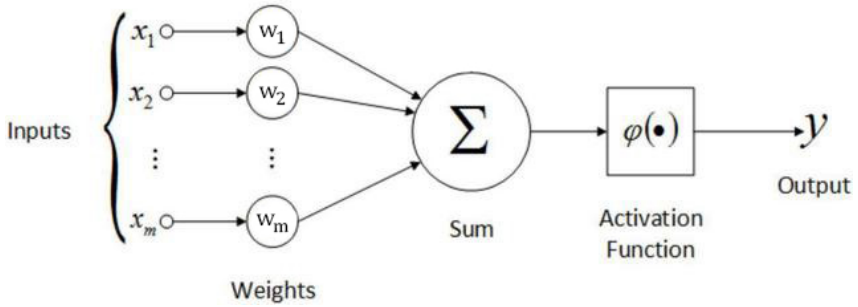

Figure 4.5: Model of a neuron [3]

With reference to Fig. 4.5, each neuron has a series of inputs ranging from x_1 to x_m where m represents the number of inputs each neuron is being fed with. Each of these inputs will be multiplied with their corresponding weights that signifies the degree of importance of the particular input denoted by w_i where i represents the index of each input, followed by the linear combination that sums the multiplication computed between each input-weight pair. The result from the summation will be passed into an activation function that serves as an amplitude delimiter that only allows a certain range of finite values to be the output of the neuron.

As mentioned above, neurons will be arranged into distinct layers that comprise the input layer, hidden layers and the output layer. Input features will traverse across each layers in a forward propagation manner that eventually leads to the resulting prediction in the output layer. Illustration in Fig. 4.6 is an ANN architecture that consists of various layers.

During the training of an ANN technique, individual weights within the network are updated based

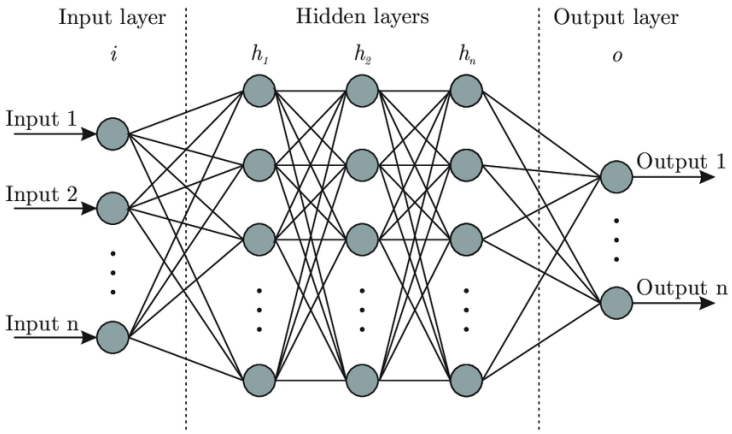

Figure 4.6: Architecture of ANN [4]

on the backpropagation mechanism. This mechanism will take into account the gradient of the error function with respect to weights of each individual neuron, then perform a gradient descent optimization that minimises the error function [45]. Besides, training of an ANN technique is a sophisticated process that requires careful consideration especially in determining hyperparameter values such as the network's learning rate, epoch size and batch size. Learning rate can be defined as the parameter that governs the pace of weight update within an ANN training. As for the case of epoch size, it represents the number of training cycles that will be made by the ANN over the entire training dataset. Whereas batch size defines the number of data samples that will be used to train the network before updating its weight. With that being said, it can be seen that hyperparameters are crucial components of an ANN as any usage of inappropriate values will cause direct impact on the performance of the ANN model. Therefore, it is best to perform a hyperparameter tuning that finetunes these hyperparameters to obtain a reliable ANN model.

Design and Implementation

5.1 Languages and Specifications

Development and training of ML techniques in this work are performed using Python 3.7.11 on a workstation that has specification of Intel® Xeon® CPU E5-2630 v2 @ 2.60GHz with 32GB RAM. Considering the size and nature of the supplied dataset, training with GPU has not been enabled as there will be no visible improvements in terms of training time of the ML techniques. Besides, the reason for using Python throughout this development is that it offers an extensive collection of open-sourced ML libraries and frameworks that allow a faster yet simplified methodology of developing ML models. Several libraries that have been adopted within this project are Keras, Tensorflow, scikit-learn and Matplotlib. The adoption of these libraries allows a faster pace of prototyping and development of ML models as they are user-friendly and well-equipped for research usages. As for the IDE, Jupyter Notebook was the preferred choice as the execution of code has been arranged in a step-by-step manner where output of code segments is stored and can be visualised in terms of cells. This criterion in Jupyter Notebook has really benefited the data science nature of this project as it permits the reproducibility of results in cases where code segments and their corresponding outputs are kept in an independent reusable component.

5.2 Data Interpolation

5.2.1 Gaussian Kernel Regression

Within this project, there exist two distinct implementations of the GKR algorithm. The first approach is to develop based on the formulation of kernel equations presented in Eq. 4.5 and Eq. 4.6 [1, 27]. The second approach is by utilising the prebuilt GKR function within the scikit-learn library. The introduction of a second GKR is to act as the baseline value for the performance estimation of the first GKR. This performance estimation has been conducted by comparing both implementations in a setting where bandwidth values were kept constant at the value of 100. As aforementioned, the bandwidth parameter of the GKR algorithm is one of the most crucial parameters that determines the effectiveness of the resulting GKR model. Therefore, it is clear that an inconsistent bandwidth value for both implementations will result in an unfair comparison. By comparing, it means analysing the performances of ML models that have been trained using the resulting interpolated dataset from each GKR. The outcome of this comparison is to be capable of identifying the suitability of each GKR implementation in performing data interpolation on the dataset being presented in this project.

Variable	Lower Limit	Higher Limit	Reference Value
Arc Current Intensity $[A]$	303	840	530
Argon Gas Flow Rate $[V_{AR}]$	20	44	40
Hydrogen Flow Rate $[V_{H_2}]$	0	17	14
Argon Carrier Gas Flow Rate $[V_{CG}]$	2	5	3.2
Injector Diameter $[ID]$	1.5	2	1.8
Injector Stand-off Distance $[D_{inj}]$	6	8	6
Average Particle Velocity $[V]$	122	408	-
Average Particle Temperature $[T]$	1236	3240	-
Average Particle Diameter $[D]$	14	101	-

Table 5.1: Physical limits of input and output parameters of the APS process [1]

5.2.2 Data Interpolation Algorithm

The principal intention of implementing a data interpolation technique using GKR approach is to tackle the issue of limited dataset made publicly available within the field of S&C technology. The technique presented within this work has achieved its motive by expanding the original dataset of size 16 into an expanded dataset with the size of 590 data instances. Compared to the proposed technique by Choudhury et al. (2011) [1], the authors' data expansion technique has only managed to expand the dataset of size 16 into an expanded dataset of size 310.

The mechanism behind the GKR proposed within this work is simply by utilising the range of values extracted from the lower and higher physical limits of each APS input processing parameters presented in Table 5.1. Each value from the range of lower and higher physical limits will be processed by the Gaussian Kernel in a univariate manner. The implementation of data interpolation technique using GKR has been demonstrated in the algorithmic outline given in Algorithm 1.

Algorithm 1 Data interpolation using Gaussian Kernel Regression

$Num_f \leftarrow$ number of features
$Num_t \leftarrow$ number of targets
for $i = 1, 2, \ldots, Num_f$ **do**
    $l_{limit} \leftarrow$ Load lower limit of the $feature[i]$
    $h_{limit} \leftarrow$ Load higher limit of the $feature[i]$
    $range \leftarrow$ Initialise a list of values containing range(l_{limit}, h_{limit})
    for $j = 1, 2, \ldots, Num_t$ **do**
        $gaussianKernel \leftarrow$ Initialisation of Gaussian Kernel using $feature[i]$ and $target[j]$
        $predList \leftarrow$ Initialisation of a list that stores predictions
        for $k = 1, 2, \ldots, len(range)$ **do**
            $predList.append(gaussianKernel.predict(range[k]))$
        end for
        Combining results in $predList$ into the interpolated dataset
    end for
end for

5.3 Artificial Neural Network (ANN)

Two different configurations of ANN have been developed within Choudhury et al. to cater both unexpanded and expanded dataset retrieved from previous research [1, 27]. According to Choudhury et al.

(2011), both configurations of ANN were proven to be effective in predicting in-flight particle characteristics of the APS process [1]. This has led to the adoption of both ANN configurations into this project. However, both configurations of ANN proposed by Choudhury et al. share similar characteristics, where they have input layer that accepts eight input features accompanied by two hidden layers that channel input features into output layer that predicts three target values. The only difference between them is the number of neurons present within each existing hidden layers, in this case the configuration of the ANN targeting unexpanded dataset has been made up of a combination of five and four neurons in the first and second hidden layer, whereas the configuration of ANN focusing on the expanded dataset has its first and second hidden layer being made up of nine and eight neurons. Illustrated in Fig. 5.1 and Fig. 5.2 are the ANN configurations that have been implemented within this project, with ANN in Fig. 5.1 targeting the unexpanded dataset and ANN of Fig. 5.2 targeting the expanded dataset.

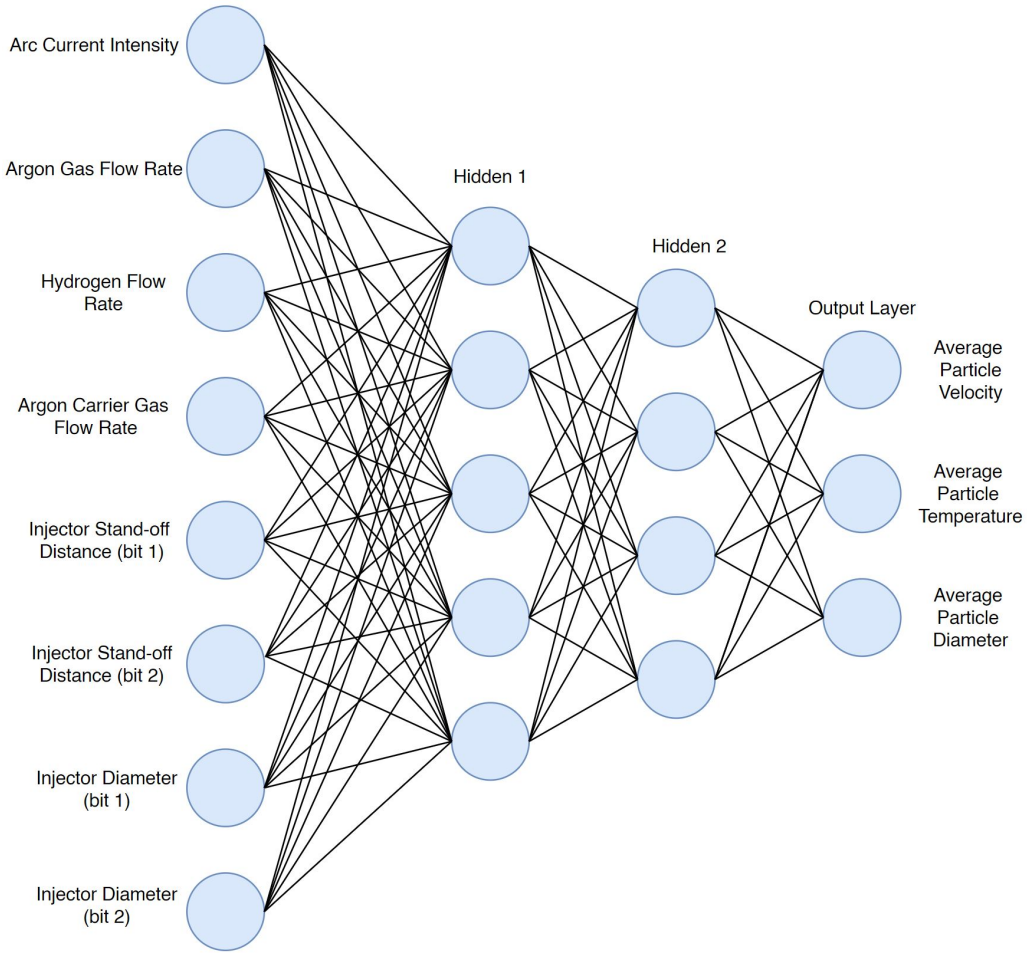

Figure 5.1: Configuration of ANN targeting unexpanded dataset

Within the literature, Choudhury et al. suggested a set of hyperparameters that aims to enhance the performance of the proposed ANN model [1]. However, not all hyperparameter values that are considered essential to the development of an ANN technique have been explicitly stated within Choudhury et al [1]. This ambiguity present within the paper has led to the act of making assumptions based on ML principles. Hyperparameter values such as the early stopping criteria and optimisation algorithm that play a significant role in preventing the ML technique from overfitting were not provided by the authors [1]. To solve this issue, assumptions have been made where early stopping criteria has been set to 1800 and optimisation algorithm was assumed to be Stochastic Gradient Descent. The value 1800 has been chosen for the early stopping criteria as analysis on the training of the ANN technique suggested that the

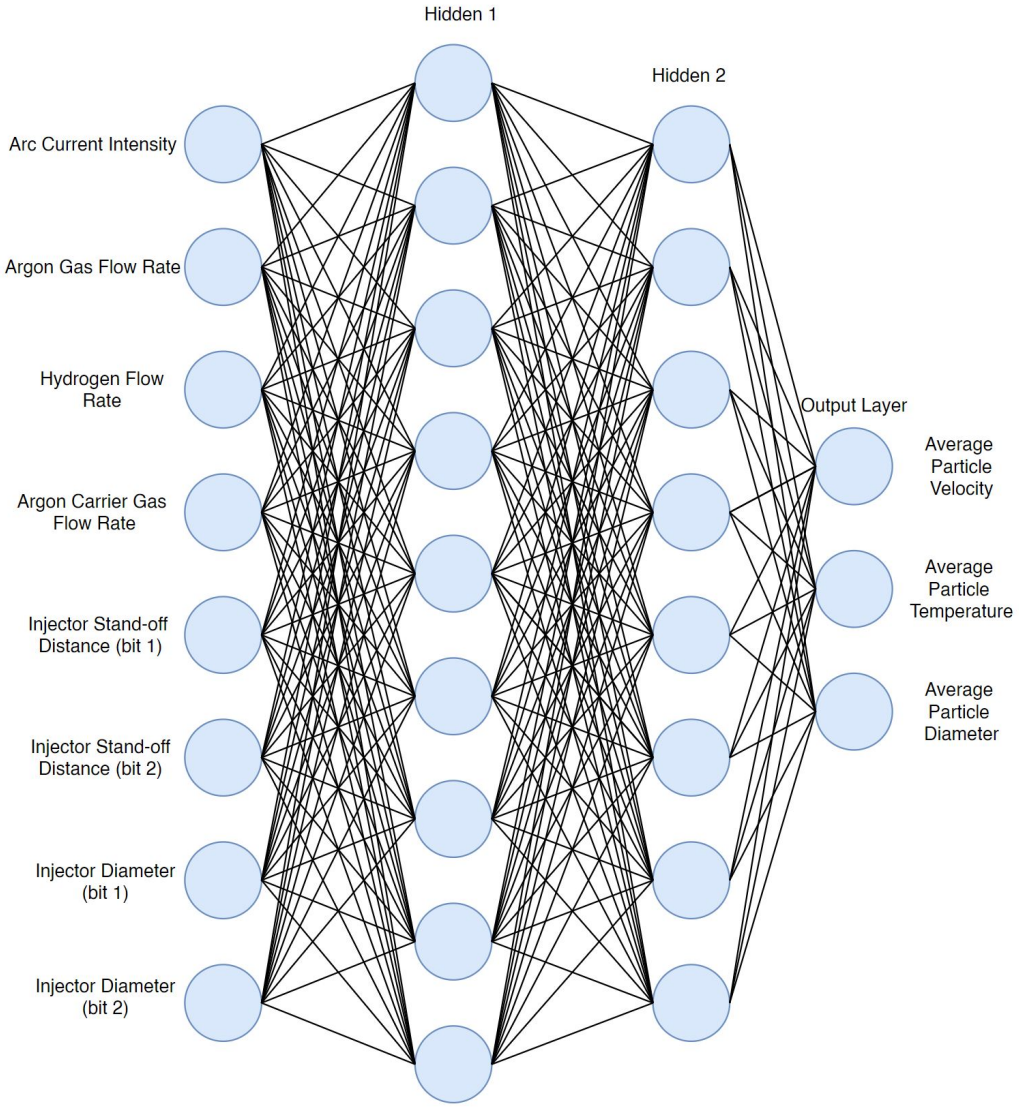

Figure 5.2: Configuration of ANN targeting expanded dataset

resulting model is more likely to overfit given that there exists no sign of improvement in the loss value after 1800 consecutive epochs during the training of the ANN. For the case of optimisation algorithm, Stochastic Gradient Descent has been the selected choice as this algorithm introduces the concept of randomness into the minimization of the loss function, allowing the algorithm to better escape local minima present within the search space.

A summary of comparison between the hyperparameters that have been suggested within this work and the one being found in Choudhury et al. [1] has been made in Table 5.2. The only notable difference between them is the activation function. Rectified Linear Unit (ReLU) has been suggested as this piecewise activation function will overcome the issue of vanishing gradient problems that are commonly found in implementation of ANN using logistic sigmoid activation functions. This suggested activation will ensure that weight updates will constantly happen throughout the training of the network.

Hyperparameter	Found in Choudhury et al.	Suggested within this Work
Epoch Size	10000	10000
Batch Size	10	10
Early Stopping Criteria (Patience)	1800	1800
Optimization Algorithm	Stochastic Gradient Descent	Stochastic Gradient Descent
Activation Function	Logistic Sigmoid	ReLU

Table 5.2: Summary of comparison between selection of hyperparameters

5.4 ML Pipeline

5.4.1 Replication of Work Proposed by Choudhury et al.

Within the work presented by Choudhury et al., it can be seen that a series of ML tasks have been integrated into an ML pipeline that automates the generation of an ML model. Illustrated in Fig. 5.3 is the sequence of ML pipeline suggested by the authors [1].

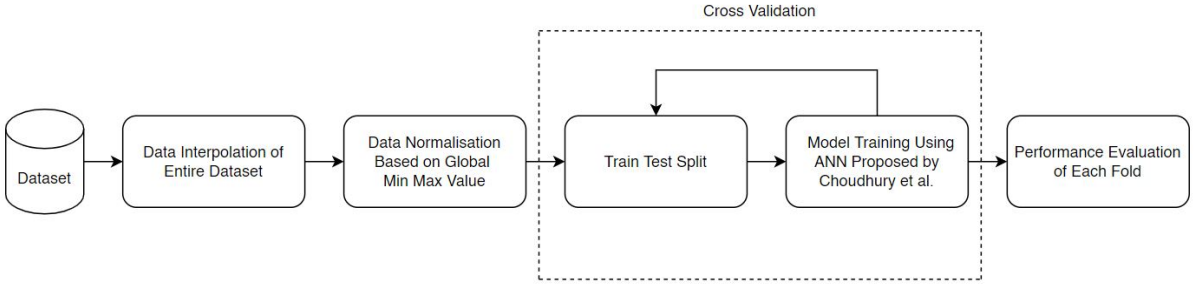

Figure 5.3: Flow of ML pipeline presented by Choudhury et al. [1]

As observed from Fig. 5.3, there lies a few aspects that require scrutiny. The first aspect is the execution of data interpolation onto the entire dataset. As discussed previously, execution of data interpolation onto the entire dataset will cause an unfair advantage to the ML technique in recognising underlying patterns within the dataset. This is due to the nature of GKR that estimates the new datapoint by considering the weights (importance) of its neighbouring points. The second aspect is the normalisation of dataset using a global minimum and maximum value, that is by referring to the physical limitations of each input and output parameters presented in Table 5.1. Whereas the third aspect is the execution of data normalisation before the dataset is being split into training and testing set. There exists a correlation between the second and third aspect as both aspects were linked to the issue of data leakage. In the second aspect, data normalisation should always be carried out only using the minimum and maximum values extracted from the training set, as this can prevent the ML technique from recognising the global distribution of the entire dataset. With that being said, it should be fairly obvious for the third aspect that splitting of dataset into training and testing set should always come before the execution of data normalisation on the dataset.

5.4.2 Rectification of Work Proposed by Choudhury et al.

Within the rectified work presented by this dissertation, there exist several notable modifications being made to address the potential issues found within Choudhury et al. [1]. Illustrated in Fig. 5.4 is the rectified pipeline that has been presented within this work.

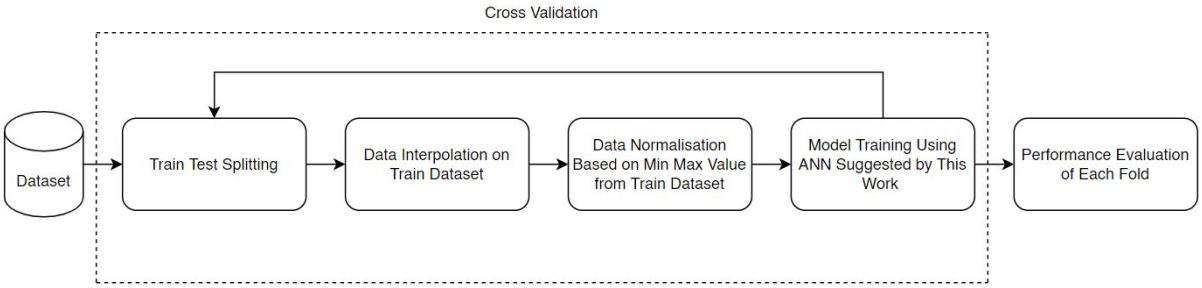

Figure 5.4: Rectified flow of ML pipeline presented within this work

As depicted in Fig. 5.4, it can be seen that all the ML pipeline elements have been reorganised into the execution of cross validation with priorities given to the process of train test splitting. This particular modification will ensure that data pre-processing techniques such as data interpolation and data normalisation will strictly be performed based on values retrieved from the training dataset of each fold within the cross validation. Meaning, occurrence of data leakage can be omitted as test set from each fold of cross validation has been isolated from any data pre-processing techniques. Hence, it can be concluded that fairness in model evaluations during cross validation is guaranteed as isolation of test sets have significantly reduced the likelihood of data leakage occurrences that leads to an overly optimistic ML model evaluation.

5.5 Hyperparameter Tuning

One of the objectives of this project is to develop a hyperparameter optimization method that finetunes the ANN that has been proposed by this project. The mechanism behind this optimization method is simply by altering the configuration of the ANN model by means of trial and error. In other words, ANN of different configurations will be trained iteratively and evaluated by the optimization method, where only the best ANN model configuration will be saved.

There exist a few vital settings within the optimization method. First, it is the usage of a Bayesian Optimization tuner that tunes the ANN technique using Gaussian Process. This approach of ANN tuning requires a higher computational power compared to the Hyperband tuner as search within the Bayesian Optimization has been performed on a much larger solution space. However, this high computational cost present will not be an obstacle for this work as the dataset presented within this work is of such a small size, therefore the process of search is still manageable. For the mechanism behind this tuner, it builds a surrogate model that corresponds to the estimation of the true objective function, where the tuner will be able to pinpoint promising regions consisting of potential minima. Based on these potential regions, the tuner will perform random sampling and update the surrogate model accordingly. After the completion of a series of iteration ($max\_trials = 10$), the tuner will conclude the optimization by selecting the best output as the global minima.

Secondly, it is the settings implemented for the possible configurations of the ANN. The input layer of the ANN has been defined to hold a collection of neurons ranging from a minimum of one neuron to

a maximum of fifteen neurons. As for the hidden layers, the possible number of hidden layers has been bounded within the range between the minimum of one layer to a maximum of twenty layers where each layer will be able to hold a range from two to fifteen neurons. The Bayesian Optimization will search for the minima present in possible configurations of ANN defined as above, resulting in a tuned model that has the best ANN configuration that solves the targeted problem within this project.

Evaluation

The evaluation section has been split into subsections that led the project towards the development of ML models that can effectively conduct predictions on a small dataset concerning the in-flight particle characteristics of the APS process. Within this section, various comparisons between ML models have been made in terms of model performance using MAE. However, other error metrics such as RMSE and R-value will be included wherever relevant to provide a better insight into the performance of the ML model.

6.1 Experiments with Standard ML Techniques

The aim of implementing standard ML techniques such as linear regression and support vector machines is to examine the suitability of these ML techniques in engaging with the APS process. Suitability of these ML techniques will be determined by comparing the performances of these standard techniques with the performance of the ANN that has been proposed by Choudhury et al. [6].

6.1.1 Linear Regression

Within this project, implementation of the linear regression model has been conducted using the scikit-learn library. This library offers an extensive number of prebuilt functions that can be used to measure scoring metrics of the trained model. Table 6.1 shows average scoring metrics of linear regression models resulting from k-fold cross validation with unexpanded and expanded datasets.

Despite having a low MAE value across both models that signify a low prediction error rate, linear regression model is still not being chosen as one of the appropriate ML techniques to tackle this APS process. Consideration has been given into the nonlinear interdependencies relationship that exist between the input and output parameters of the APS process where the linear regression model is believed to be over simplistic to generalise within such a complex problem space.

	Unexpanded Dataset (k = 4)	Expanded Dataset (k = 10)
MAE (SD)	0.1137 (0.0454)	0.0340 (0.0252)
MSE	0.0558	0.0038
RMSE	0.1502	0.0427

Table 6.1: Average scoring metrics of Linear Regression in different settings

		Average Particle Velocity	Average Particle Temperature	Average Particle Diameter
RBF	**MAE (SD)**	0.0859 (0.0284)	0.0503 (0.0432)	0.0641 (0.0297)
	MSE	0.0092	0.0055	0.0056
	RMSE	0.0916	0.0562	0.0677
Linear	**MAE (SD)**	0.0716 (0.0271)	0.00486 (0.0445)	0.0642 (0.0296)
	MSE	0.0063	0.0054	0.0056
	RMSE	0.0760	0.0541	0.0681
Polynomial	**MAE (SD)**	0.0718 (0.0269)	0.0047 (0.0456)	0.0645 (0.0292)
	MSE	0.0064	0.0054	0.0056
	RMSE	0.0765	0.0528	0.0681

Table 6.2: Average scoring metrics using RBF, Linear, and Polynomial Kernels

6.1.2 Support Vector Machine (SVM)

Although there is not much implementation of SVM in in-flight particle characteristics prediction of APS process, the SVM approach using Radial Basis Function (RBF) Kernel that has been adopted by Fang et al. has still been proven to be viable [46]. Within this project, three distinct kernel functions have been used, namely RBF, Linear and Polynomial Kernel. Due to the nature of SVM provided by scikit-learn, only one-dimensional prediction can be made at a time. This has resulted in the tabulation shown in Table 6.2 where average scoring metrics of the SVM models on the expanded dataset [27] has been grouped based on each output parameters within the APS process.

The results within Table 6.2 were generated based on a 10-fold cross validation where errors generated from each fold has been averaged. It can be seen that all kernel functions resulted in a low MAE with low standard deviation which denotes those residual errors are centred around the mean. This signifies that implementation of SVM with proper hyperparameter tuning is indeed a good ML technique to be used to predict the APS process.

6.2 Replication of Work Proposed by Choudhury et al.

6.2.1 Artificial Neural Network (ANN)

There is a noticeable increment in existing work from the field of S&C technology that has adapted the ANN technique to perform predictions of the in-flight particle characteristics. However, the implementation of ANN within this project was an inspiration from Choudhury et al. (2011) that revealed the exceptional performance presented by ANN in predicting the APS process. Architecture of the ANNs presented within this work were developed based on the exact configurations provided by Choudhury et al. [1]. The only difference that exists between both work is the choice of development platform. In this project, implementation of ANN has been made using the Python environment accompanied by Tensorflow and Keras that simplifies the development of a Multilayer Perceptron Model. In contrast, a MATLAB approach has been taken by Choudhury et al. in developing the ANN [1].

In Table 6.3, it shows the side-by-side comparison between the model evaluation made on the replicated ANN model within this work (*repl_ann*) and the ANN model proposed by Choudhury et al. (*proposed_ann*) [1]. As shown in Table 6.3, evaluation of *repl_ann* trained using the unexpanded or expanded dataset exhibit a relatively low MAE with low SD that indicates the errors made by the model are clustered more closely at the mean. Meaning, it implies that errors made by this model are less

	Average MAE (SD)	
	repl_ann	***proposed_ann***
Unexpanded dataset	0.0645 (0.0380)	0.0639 (0.0082)
Expanded dataset	0.0171 (0.0048)	0.0072 (0.0017)

Table 6.3: Comparison between model evaluation of the replicated ANN model and the ANN model proposed by Choudhury et al. [1]

	Range of ANN Models		
	repl_ann* with *approach_1	***repl_ann* with *approach_2***	***proposed_ann***
MAE (SD)	0.0044 (0.0008)	0.0014 (0.0005)	0.0072 (0.0017)

Table 6.4: Comparison among model evaluations using various data interpolation techniques

dispersed hence claiming the *repl_ann* to be considerably reliable in making predictions of in-flight particle characteristics of the APS process. To further assure the performance of *repl_ann*, its scoring metrics have been matched closely to the evaluation of *proposed_ann*. This comparison suggested that scoring metrics generated by *repl_ann* and the *proposed_ann* are relatively close to each other. Theoretically, it can be deduced that the *repl_ann* trained using the exact datasets provided in [1, 27] is comparatively similar to the implementation presented within Choudhury et al. [1].

6.2.2 Artificial Neural Network with Data Interpolation

An ML pipeline that contains a series of techniques ranging from data preprocessing to training of ML technique has been presented within Choudhury et al. [1]. The success of replicating the base configuration of *proposed_ann* has led to a tighter scrutiny into the ML pipeline presented by Choudhury et al.[1], as illustrated in Fig. 5.3 . This scrutiny suggested that data interpolation technique presented by the authors plays a crucial role in enhancing the performance of the resulting ANN model. With reference to Table 6.3, a significant decrease in error rate can be observed within *proposed_ann* trained using the expanded dataset. It is evident that *proposed_ann* trained using the dataset that has been expanded by means of data interpolation technique has shown vast improvement compared to *proposed_ann* trained using the unexpanded dataset. This particular finding has driven the project in attempting to replicate the data interpolation technique being presented in Choudhury et al. (2011).

As discussed previously, replication of data interpolation techniques has been constructed in two distinct approaches, namely *approach_1* that is based on the formulation provided in Eq. 4.5 and Eq. 4.6; and *approach_2* that utilises the prebuilt GKR function within the scikit-learn library. To evaluate the replicated data interpolation techniques, the interpolated datasets generated from both approaches will be adopted into the training of *repl_ann*, resulting in two distinct sets of model evaluation. Besides, a comparison among the model evaluations of both *repl_anns* practising distinct data interpolation approaches and *proposed_ann* has been made in Table 6.4.

With reference to Table 6.4, model evaluations of both *repl_anns* utilising different data interpolation approaches demonstrated remarkable performances in terms of MAE and SD. Illustrated in Fig. 6.1 is the error distribution obtained by performing a *10*-fold cross validation onto each respective approach. It is clear that both *repl_ann* models have outperformed the performance presented by *proposed_ann*. Tacitly, an inference can be drawn based on this analogy where both implementation of data interpolation

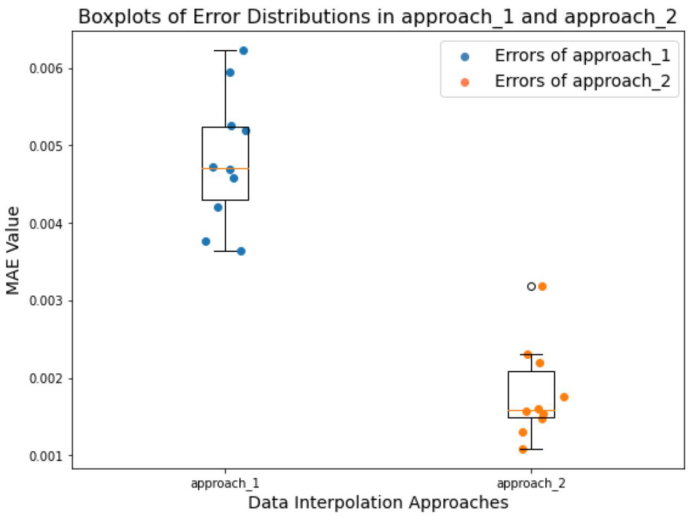

Figure 6.1: Illustration of MAE distribution from 10-fold cross validation

techniques presented to the *repl_ann* shows close proximity to the implementation of data interpolation technique presented by Choudhury et al. [1].

6.2.3 Problems Encountered

As stated previously, this project was ambitioned to rectify potential problems that exist within Choudhury et al. (2011) [1]. Therefore, it is clear that replication of the authors' work was one of the initial procedures to be taken to progress into the project. Due to the degree of ambiguity present within the paper [1], rigorous effort has been made to replicate the authors' work. For instance, the authors proposed a data expansion technique using GKR that aims to produce a dataset of a larger size that facilitates the training and optimization of *proposed_ann*. However, it is impractical to replicate the exact GKR being proposed as crucial parameters such as bandwidth value that represents the smoothness of the kernel density plots has not been defined within the literature. Thus, manual testing by means of trial and error has to be conducted to obtain the nearest identical output of the dataset that has been provided by the literature.

As described by Choudhury et al., training of *proposed_ann* is heavily dependent on hyperparameters such as batch size and the early stopping criteria [1]. These hyperparameters are said to be crucial components of *proposed_ann* as any modifications on these hyperparameters will cause direct impact on the performance of the resulting ANN model. Nevertheless, both hyperparameters are again not being provided by the paper, hence requiring extra effort in monitoring the training process of *repl_ann* to search for possible values that will eventually match the results oriented within the paper.

Knowing that a dataset with a limited number of data instances has been used throughout the research by Choudhury et al. [1], it is understandable that the evaluation on the performance of the trained ANN model would be challenging. To address this issue, the authors adopted the usage of k-fold cross validation that aims to provide an estimation on the overall performance of *proposed_ann*. In this case, the k value of the cross validation holds an important role in regulating the amount of bias induced within the model evaluation. A low value of k will result in an estimate of model evaluation that has higher sensitivity to noise present within the dataset, and it is the vice versa for the case of high value of k. However, the value of k has once again been neglected and not being presented within the work [1]. To solve this issue, assumptions of the k value have been made based on principles of ML, that is to

	ori_pipeline		*rect_pipeline*		*proposed_ann*
	approach_1	***approach_2***	***approach_1***	***approach_2***	
MAE (SD)	0.0044 (0.0008)	0.0014 (0.0005)	1.8551 (1.6988)	0.4177 (0.1920)	0.0072 (0.0017)

Table 6.5: Comparison of model evaluations based on various ML pipelines

consider the trade-off between the biases present within the trained model and the computational cost to conduct the cross validation.

6.3 Rectification of Work Proposed by Choudhury et al.

The inference that has been made within section 6.2.2 suggested that the replicated data interpolation techniques presented within this project is similar to the technique suggested by Choudhury et al. [1]. This inference has been made by performing analysis on model evaluations of *repl_ann* being trained using the dataset that has been interpolated using *approach_1* and *approach_2* along with the model evaluation being produced by Choudhury et al. [1]. However, this success in replicating the data interpolation technique has led to several key findings within this project. During the replication process, several underlying flaws that introduce the concept of data leakage have been discovered within the pre-processing stage of the authors' suggested ML pipeline (*ori_pipeline*) illustrated in Fig. 5.3. These flaws, as discussed in section 3.3 were identified as the main contributor to result in an overly optimistic model evaluation.

Rectification work conducted on *ori_pipeline* has been made simply by reorganising the flow of most existing elements. This rectified ML pipeline (*rect_pipeline*) illustrated in Fig. 5.4, is the solution that has been presented by this project to eradicate the occurrence of data leakage found in *ori_pipeline*. Knowing the fact that both replicated works presented within this project showed close proximity to their corresponding original work proposed by Choudhury et al., it is therefore sensible to apply these replicated methods, the *repl_ann*, *approach_1* and *approach_2* into *rect_pipeline* to demonstrate the occurrence of data leakage within *ori_pipeline*.

A comparison has been made among the performances of *repl_ann* models trained on dataset that has been interpolated by *approach_1* and *approach_2* using *ori_pipeline* and *rect_pipeline*. These comparisons has been tabulated in Table 6.5 alongside with the performance of the original work provided by Choudhury et al.[1].

According to Table 6.5, it can be seen that both *repl_anns* that have been trained in the *rect_pipeline* using datasets expanded by *approach_1* and *approach_2* exhibit unsatisfactory performances. The comparison has deliberately portrayed the impact caused by data leakage, showing the contrast between a model that has prior information regarding the test set and a model that does not. Referring to *ori_pipeline*, data interpolation method was the first to be applied onto the entire input dataset. Due to the nature of GKR found within the data interpolation technique, information from the test set has been leaked into the training set in such a way that estimation of new data points has been made by considering the "importance" of its neighbouring points. These "importance" of neighbouring points are the culprits that cultivated the issue of data leakage. Besides, normalisation within *ori_pipeline* has been made based on the global minimum and maximum value of the dataset. In theory, this will surely introduce the knowledge of the global distribution present within the dataset, hence causing an unfair advantage to the ML technique in recognising the underlying pattern within the dataset. With that being said, it is certain that these unrealistically high performances presented by models trained using the *ori_pipeline* was in fact caused by data leakage as these models have already 'seen', either directly or indirectly, the

Layers	Number of Neurons
Input	1
Hidden 1	7
Hidden 2	7
Hidden 3	11
Hidden 4	2
Hidden 5	12
Hidden 6	11
Hidden 7	6
Hidden 8	6
Hidden 9	2
Hidden 10	6
Hidden 11	6
Output	3

Table 6.6: Configuration of *tuned_ann* after hyperparameter tuning

	Untuned		Tuned	
	approach_1	***approach_2***	***approach_1***	***approach_2***
MAE (SD)	1.8551 (1.6988)	0.4177 (0.1920)	29.0682 (0.0134)	0.3648 (0.0277)

Table 6.7: Comparison of performances of the tuned and untuned models

information within the test set. Therefore, isolation of test sets should always be emphasised throughout the training of an ML model to avoid the generation of overly optimistic evaluations that is most likely to fail during production.

6.4 Hyperparameter Tuning

From the discussion above, it is undeniable that models trained using the *rect_pipeline* exhibit a poor performance in predicting the in-flight particle characteristics of the APS process. Nevertheless, ameliorative measures have been taken by performing hyperparameter tuning on these *repl_anns*. Table 6.6 shows the summary of *repl_ann* configuration with the best performance (*tuned_ann*) resulting from the hyperparameter tuning algorithm.

The resulting configuration presented within Table 6.6 has a total of 579 learnable parameters. A series of model evaluations have been performed onto *tuned_ann* using the *rect_pipeline* manner. The purpose of conducting these model evaluations is to inspect the significance of performing hyperparameter tuning onto *tuned_anns*. In other words, evaluations of *tuned_anns* will be compared with the performance of *repl_anns* that have been trained using *rect_pipeline* to examine the amount of influence caused by the execution of hyperparameter tuning. Table 6.7 shows the contrast between the performances of *tuned_anns* and *repl_anns* where both have been trained using the *rect_pipeline* fashion.

With reference to Table 6.7, there exist no drastic improvement in performance present within the tuned model. In fact, the performance of the untuned model trained with the dataset interpolated using *approach_1* shows a better performance compared to its corresponding tuned model. This has revealed that it may be inappropriate to use *approach_1* to interpolate the dataset concerning the APS process. To summarise the process of hyperparameter tuning, the application of this tuning algorithm did not cause much improvement onto the performance of the resulting ANN models to accurately predict in-flight

	ori_pipeline		*rect_pipeline*	
	approach_1	***approach_2***	***approach_1***	***approach_2***
MAE (SD)	0.0059 (0.0010)	0.0103 (0.0016)	1.3055 (0.4998)	0.2631 (0.1700)

Table 6.8: Comparison of model evaluations trained using the alternative dataset based on various ML pipelines

particle characteristics of the APS process, as the complication that caused the exceptional performances presented by Choudhury et al. [1] still lies within *ori_pipeline* that introduces the concept of data leakage into the evaluation of the trained ML models.

6.5 Introduction of Alternative Dataset

An alternative set of data concerning the APS process has been provided by the researchers of FoE in UoN. The alternative dataset consists of a total of 26 data instances and has been made available in Appendix C. The purpose of adopting the alternative dataset is to test and further strengthen the claims made regarding the occurrence of data leakage within the suggested *ori_pipeline*. To result in a fair comparison, configuration of the ANN to tackle this alternative dataset has been standardised to the usage of *repl_ann*. Each of these ANN will be trained using a dataset that has been interpolated by either *approach_1* or *approach_2* before it is being evaluated using ML pipelines such as *ori_pipeline* or *rect_pipeline*. Table 6.8 presents the results generated from each model evaluations made using the alternative dataset based on various ML pipelines.

Based on Table 6.8, although the model in *rect_pipeline* that has been trained using the interpolated dataset resulted from *approach_2* showed an acceptable performance, attention has to be focused onto the noticeable spike in terms of error rate between the ML models that has been trained using the *ori_pipeline* and the *rect_pipeline*. Despite using an alternative dataset, this trend is nonetheless mutual to the results illustrated in Table 6.5 and Table 6.7. Therefore, a conclusion can be drawn based on this analysis that there is a guaranteed occurrence of data leakage within the *ori_pipeline* suggested by Choudhury et al. (2011), that led to an overly optimistic model evaluation [1]. This issue can simply be mitigated using the ML pipeline presented within this work, the *rect_pipeline*, where a series of ML pipeline elements has been arranged in a conventional manner.

6.6 Research Limitations

Despite having successfully accomplished most of the aims and objectives that have been established at the beginning of the project, there exist several key limitations within this project that is worth mentioning. The first limitation lies within the proposed data interpolation technique where interpolation of data has been made by the introduction of noise elements into the domain values. There exist no solid physics and chemistry principles that guide this interpolation technique in expanding the dataset. Thus, the relevance of the resulting interpolated dataset is purely dependent onto the quality of the input dataset. The second limitation concerns the shortage of data provided, resulting in not being able to verify the true performance of the ML model when being fed with unseen data.

Summary and Reflections

7.1 Conclusion

The high cost of conducting physical Atmospheric Plasma Spray (APS) process is one of the dilemmas that has been faced by researchers from the Faculty of Engineering (FoE) in producing a Thermal Barrier Coating (TBC) that exhibits reliable and performant characteristics. The aim of this project has been to investigate the application of Machine Learning (ML) techniques in predicting in-flight particle characteristics of the APS process. These ML techniques serve as an alternative approach for researchers to gain insights into the resulting TBC characteristics using only a set of input features. Comparison amongst the ML techniques made within this project showed that Artificial Neural Network (ANN) has the best predictive capability. This has resulted in the selection of ANN as the main ML technique used to model the complex relationship found between the input and output parameters of the APS process. Application of Gaussian Kernel Regression (GKR) in this project has successfully caused expansion to the input dataset, which improved the generalisation capability of the ANN. Besides, implementation of the ML pipeline that has been proposed by this project has effectively eliminated the occurrence of data leakage during the process of model evaluation. These less optimistic model evaluations resulting from the proposed ML pipeline showed a closer depiction to its true performance in events where it has been fed with unseen data. Therefore, the proposed ML pipeline with the suggested ANN model is appropriate to be incorporated into industrial usage to facilitate researchers in gaining a better insight to the in-flight particle characteristics of the APS process.

7.2 Future Directions

It is apparent that the application of ML techniques within the field of S&C technology are relatively immature. There exist plenty of possible directions that could be taken in the future.

The most obvious direction is to gather more data for the training of the ML technique. One possible step to achieve this is by combining a variety of datasets from different sources, for instance, scraping datasets that have been presented within published research work. Although the scraped datasets may be derived from varied chemical compositions, the underlying patterns within these datasets are nevertheless beneficial to the ML technique in understanding the physics and chemistry principles that guide the characteristics of in-flight particles of the APS process.

Secondly, it is the implementation of tacit knowledge from the researchers of FoE into the ML technique. This can be established by setting constraints that have been proposed by the researchers that limit the predicting scope of the ML technique. For example, past experiences accumulated by the researchers will allow them in recognising a particular trend that might exist within the in-flight particle characteristics of the APS process when given a series of input parameter settings. The introduction of intelligence from the researchers into ML techniques will certainly promote a better background under-

standing to the ML technique, thus leading to an ML technique with a better performance in conducting predictions on the APS process.

Considering the size of dataset presented within the APS process, future work could be done in researching for a better ML technique that best fits ML problems that have a limited dataset. The implementation of ANN within this project may not be the best option as fitting of an ANN using a small dataset can easily lead to the issue of overfitting. A potential step to be taken in the future is to explore ML techniques such as few-shot learning algorithm that can adapt itself effectively onto a new task based on training using only a few data samples [47].

Lastly, it is to implement an online ML technique that facilitates a real time training using data that has been fed sequentially into the machine. There hardly exist any implementations of online ML techniques present within the field of S&C technology. However, this particular directive will be advantageous to researchers from the FoE as they will be able to utilise results generated from their day-to-day laboratory experiments as the training data for the machine. This constant supply of training data will ensure an improved reliability of the resulting ML model that tackles the problem of predicting in-flight particle characteristics of the APS process.

7.3 Project Management

Throughout this project, an agile methodology has been practised ensuring that requirements are constantly being met and evaluated along the process of project development. Besides, work requiring more effort has been split into smaller tasks where it allows a better administration over the entire project workflow especially in meeting tight deadlines contributed from other modules.

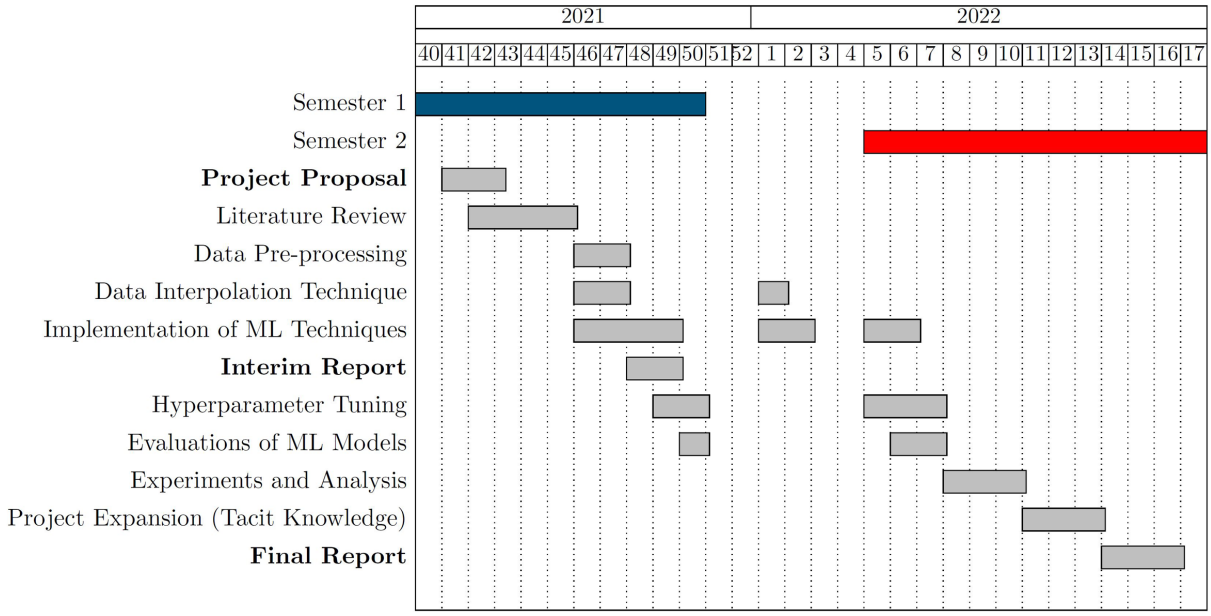

Figure 7.1: Old time plan

Presented in Fig. 7.1 is the old time plan that has been suggested in the Interim Report. Illustrated in Fig. 7.2 is the new time plan where slight alteration has been made onto the old time plan. As seen from Fig. 7.1, the initial plan of the project was to have a heavier workload being assigned into the first semester as I have made a 50/70 split in modules selection that indicated second semester will be more occupied compared to the first. However, this was certainly not the case as I was being faced with a few

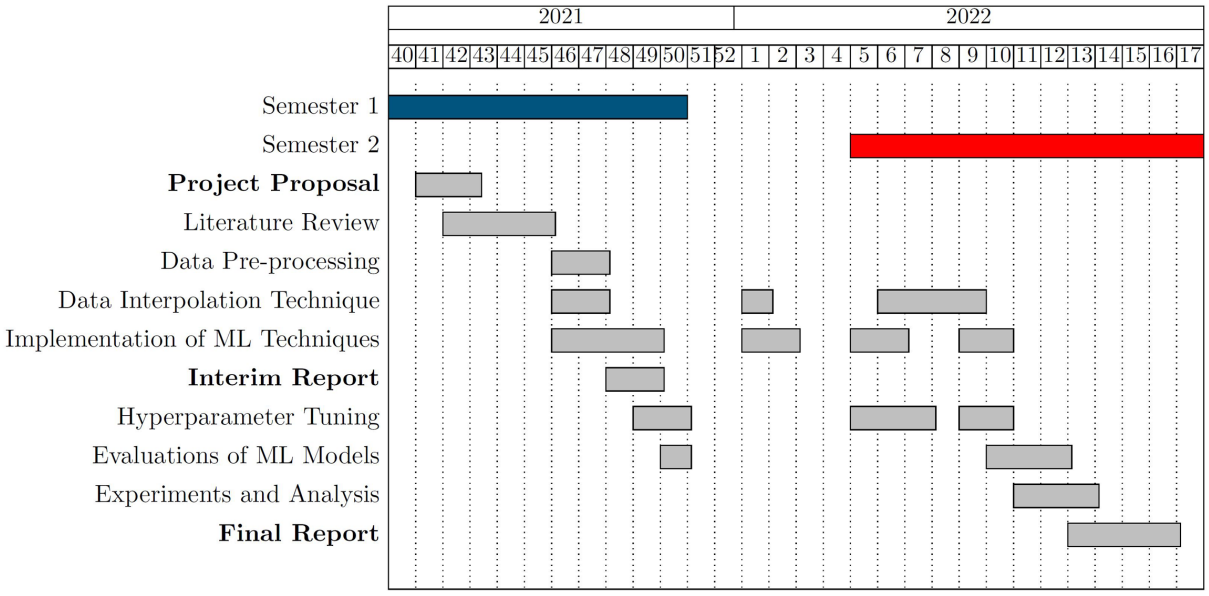

Figure 7.2: New time plan

major setbacks during the course of the project development. Referring to Fig. 7.2, it is observable that a prolonged period of time has been used to develop the data interpolation technique along with the ML techniques. This has resulted in a delay in execution of most remaining tasks presented in Fig. 7.2 that eventually contributed to the elimination of project expansion from the work plan.

Being a collaborative project with the FoE, there were two distinct meetings, namely the sprint meetings and sessions with researchers from the FoE. In the first semester that consisted mostly of information gathering work, sprint meetings involving my supervisor would be held at the end of each week where constructive evaluations based on my progress would be provided. Nonetheless, sprints on a weekly basis were found to be inapplicable in the second semester. This has caused the remaining sprint meetings to be conducted on a bi-weekly basis, since most work that has been scheduled into the second semester were development intensive work with concepts that I have little experience with. This change has significantly resulted in fruitful bi-weekly meetings where I was able to better express my development work accumulated throughout the two-week time rather than having to describe bits and pieces of incomplete code components. As for the sessions with researchers from the FoE, it has been held on every Thursday where discussions regarding technical aspects of the project will be made. Both meetings have motivated me to always strive for the best as credible suggestions and evaluations made during the meetings have acknowledged myself regarding the potential flaws found within my project development.

Another point that is worth mentioning is that more time has been allocated into the writing of the final dissertation. This adjustment has been made based on past experiences where the write up of the interim report took longer than I had expected. To spare more time, a portion of work has been allocated into public holidays, while being able to maintain a healthy work-life balance. Thanks to this adjustment, I have had more time to do cross-checks on the content of the dissertation.

7.4 Contributions and Reflections

7.4.1 Research Novelty

At present, general research in the application of ML Technique for predicting in-flight particle characteristics of an APS process is still in its infancy. Although it is accepted that Choudhury et al.'s research with the implementation of ANN within an ML pipeline consisting of various data pre-processing methodologies is highly effective, the effect of data pre-processing techniques on preventing data leakage has not been extensively studied [1]. Therefore, it is sensible to modify the ML pipeline described by Choudhury et al. in order to realise the genuine performance of the ANN in predicting the in-flight particle characteristics of the APS process in an environment without data leakage.

7.4.2 Personal Reflections

This project is by far the most complex project that I have ever worked on. Despite the fact that the project has a few limitations, I am certain that all the core objectives that have been initially presented were successfully achieved. In predicting the in-flight particle characteristics of the APS process, I performed a novel work by proposing an ML model that is free from the issue of data leakage. The proposed ML model includes sufficient details concerning the real-world problem being faced by researchers within the S&C industry. Therefore, I am confident that this proposed ML implementation will be a cost-effective and efficient approach to assist researchers from the FoE in producing a competent TBC for jet turbines that offers a better thermal and cooling feature.

As a summary, this project has been challenging but ultimately rewarding as it promotes the development of a strong research skill where no other project has got to offer. With a keen interest in Machine Learning and aviation, this project has given me the opportunity to work closely with my supervisor and researchers from the FoE, allowing me to gain a better understanding of both fields while being able to apply my knowledge into real world problems. As a result, it is undeniable that values gained through this project have better prepared me to work on industrial problems in the near future.

Bibliography

[1] T. A. Choudhury, N. Hosseinzadeh, and C. C. Berndt, "Artificial Neural Network application for predicting in-flight particle characteristics of an atmospheric plasma spray process," *Surface and Coatings Technology*, vol. 205, no. 21-22, pp. 4886–4895, 8 2011.

[2] T. A. Choudhury, C. C. Berndt, and Z. Man, "An extreme learning machine algorithm to predict the in-flight particle characteristics of an atmospheric plasma spray process," *Plasma Chemistry and Plasma Processing*, vol. 33, no. 5, pp. 993–1023, 10 2013.

[3] A. Borundiya, "Activation Function for Multi-layer Neural Networks." [Online]. Available: https://medium.com/@aborundiya/activation-function-for-multi-layer-neural-networks-a07ac473f69e

[4] F. Bre, "Artificial Neural Network Architecture," 11 2017. [Online]. Available: https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051

[5] G. G. Fleming and U. Ziegler, "ENVIRONMENTAL TRENDS IN AVIATION TO 2050," Tech. Rep., 2016. [Online]. Available: https://www.icao.int/environmental-protection/Documents/EnvironmentalReports/2016/ENVReport2016_pg16-22.pdf

[6] S. Sumsurooah, "COP26: innovating electric aircraft for greener global transport." [Online]. Available: https://www.nottingham.ac.uk/vision/cop26-innovating-electric aircraft-for-freener-global-transport

[7] D. Parsons, "Carbon Brainprint Case Study: Ceramic Coatings for Jet Engine Turbine Blades," 7 2011. [Online]. Available: https://dspace.lib.cranfield.ac.uk/bitstream/handle/1826/6804/CBrainprint-CS01-JetTurbines.pdf?sequence=1

[8] R. A. Miller, "Thermal Barrier Coatings for Aircraft Engines: History and Directions," *Journal of Thermal Spray Technology*, vol. 6, no. 1, pp. 35–42, 3 1997.

[9] H. Xu, H. Guo, and S. Gong, "Thermal barrier coatings," in *Developments in High Temperature Corrosion and Protection of Materials*. Woodhead Publishing Series in Metals and Surface Engineering, 2008, pp. 476–491.

[10] A. S. H. Makhlouf, "Current and advanced coating technologies for industrial applications," *Nanocoatings and Ultra-Thin Films*, pp. 3–23, 2011.

[11] A. Feuerstein, J. Knapp, T. Taylor, A. Ashary, A. Bolcavage, and N. Hitchman, "Technical and economical aspects of current thermal barrier coating systems for gas turbine engines by thermal spray and EBPVD: A review," pp. 199–213, 2008.

[12] M. Viswanathan and R. Kotagiri, "Comparing the Performance of Support Vector Machines to Regression with Structural Risk Minimisation," in *Proceedings of International Conference on Intelligent Sensing and Information Processing, ICISIP 2004*, 2004, pp. 445–449.

[13] K. Bobzin, W. Wietheger, H. Heinemann, S. R. Dokhanchi, M. Rom, and G. Visconti, "Prediction of Particle Properties in Plasma Spraying Based on Machine Learning," pp. 1751–1764, 10 2021.

[14] C. Cortes, V. Vapnik, and L. Saitta, "Support-Vector Networks Editor," Tech. Rep., 1995.

[15] T. Gurgenc, O. Altay, M. Ulas, and C. Ozel, "Extreme learning machine and support vector regression wear loss predictions for magnesium alloys coated using various spray coating methods," *Journal of Applied Physics*, vol. 127, no. 18, 5 2020.

[16] J. Xue and M. Huang, "Optimization of Plasma Spray Process VIA Orthogonal Test Design Method, SVM, and Improved PSO," *International Journal of Materials, Mechanics and Manufacturing*, vol. 5, no. 3, pp. 153–158, 8 2017. [Online]. Available: http://www.ijmmm.org/index.php?m=contentc=indexa=showcatid=51id=378

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 12 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[18] M. Liu, Z. Yu, H. Wu, H. Liao, Q. Zhu, and S. Deng, "Implementation of Artificial Neural Networks for Forecasting the HVOF Spray Process and HVOF Sprayed Coatings," *Journal of Thermal Spray Technology*, 2021.

[19] A. F. Kanta, G. Montavon, M. P. Planche, and C. Coddet, "Artificial neural networks implementation in plasma spray process: Prediction of power parameters and in-flight particle characteristics vs. desired coating structural attributes," *Surface and Coatings Technology*, vol. 203, no. 22, pp. 3361–3369, 8 2009.

[20] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 12 2006.

[21] C. Zhang, A. F. Kanta, C. X. Li, C. J. Li, M. P. Planche, H. Liao, and C. Coddet, "Effect of in-flight particle characteristics on the coating properties of atmospheric plasma-sprayed 8 mol% Y2O3-ZrO2 electrolyte coating studying by artificial neural networks," *Surface and Coatings Technology*, vol. 204, no. 4, pp. 463–469, 11 2009.

[22] P. Fauchais and M. Vardelle, "Sensors in spray processes," in *Journal of Thermal Spray Technology*, vol. 19, no. 4, 6 2010, pp. 668–694.

[23] T. Liu, M. P. Planche, A. F. Kanta, S. Deng, G. Montavon, K. Deng, and Z. M. Ren, "Plasma spray process operating parameters optimization based on artificial intelligence," *Plasma Chemistry and Plasma Processing*, vol. 33, no. 5, pp. 1025–1041, 10 2013.

[24] L. Wang, J. C. Fang, Z. Y. Zhao, and H. P. Zeng, "Application of backward propagation network for forecasting hardness and porosity of coatings by plasma spraying," *Surface and Coatings Technology*, vol. 201, no. 9-11 SPEC. ISS., pp. 5085–5089, 2 2007.

[25] M. D. Jean, B. T. Lin, and J. H. Chou, "Application of an artificial neural network for simulating robust plasma-sprayed zirconia coatings," *Journal of the American Ceramic Society*, vol. 91, no. 5, pp. 1539–1547, 5 2008.

[26] A. F. Kanta, G. Montavon, and C. Coddet, "Predicting spray processing parameters from required coating structural attributes by artificial intelligence," *Advanced Engineering Materials*, vol. 8, no. 7, pp. 628–635, 7 2006.

[27] T. A. Choudhury, "Artificial Neural Networks Applied To Plasma Spray Manufacturing," Ph.D. dissertation, Swinburne University of Technology, Victoria, 2013.

[28] S. Mohammad, B. J. M, A. Professor, and R. Scholar, "DATA WRANGLING AND DATA LEAKAGE IN MACHINE LEARNING FOR HEALTHCARE," Tech. Rep., 2018. [Online]. Available: https://www.researchgate.net/publication/354555695

[29] A. Zheng and A. Casari, *Feature Engineering for Machine Learning PRINCIPLES AND TECHNIQUES FOR DATA SCIENTISTS*. Sebastopol: O' Reilly Media, Inc., 8 2018.

[30] S. Kaufman, S. Rosset, and C. Perlich, "Leakage in data mining: Formulation, detection, and avoidance," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 556–563.

[31] J. Görtler, R. Kehlbeck, and O. Deussen, "A Visual Exploration of Gaussian Processes," *Distill*, vol. 4, no. 4, 4 2019.

[32] M. Kuhn and K. Johnson, "Feature Engineering and Selection; A Practical Approach for Predictive Models; Edition 1." Boca Raton: CRC Press, 2020.

[33] J. Brownlee, "How to Avoid Data Leakage When Performing Data Preparation," 6 2020. [Online]. Available: https://machinelearningmastery.com/data-preparation-without-data-leakage/

[34] H. Reynolds and G. Reynolds, *Analysis of Nominal Data*. SAGE Publications, 1977.

[35] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining Practical Machine Learning Tools and Techniques Fourth Edition*, 4th ed. Cambridge: Morgan Kaufmann, 2017. [Online]. Available: https://www.elsevier.com

[36] G. James, D. Witten, T. Hastie, and R. Tibshirani, *Springer Texts in Statistics An Introduction to Statistical Learning*. New York: Springer, 2013. [Online]. Available: http://www.springer.com/series/417

[37] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," 8 2020. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/

[38] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge: MIT Press, 2006.

[39] J. Wang, "An Intuitive Tutorial to Gaussian Processes Regression," 9 2020. [Online]. Available: http://arxiv.org/abs/2009.10862

[40] K. Mehta, "(Gaussian) Kernel Regression from Scratch," 2020. [Online]. Available: https://www.kaggle.com/code/kunjmehta/gaussian-kernel-regression-from-scratch/notebook

[41] A. Malekian and N. Chitsaz, "Concepts, procedures, and applications of artificial neural network models in streamflow forecasting," *Advances in Streamflow Forecasting*, pp. 115–147, 1 2021.

[42] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 11 2018.

[43] R. Sadiq, M. J. Rodriguez, and H. R. Mian, "Empirical Models to Predict Disinfection By-Products (DBPs) in Drinking Water: An Updated Review," *Encyclopedia of Environmental Health*, pp. 324–338, 1 2019.

[44] S. Haykin, *Neural Networks - A Comprehensive Foundation*, 9th ed. Singapore: Peason Education, 2005.

[45] "Backpropagation Step by Step." [Online]. Available: https://hmkcode.com/ai/backpropagation-step-by-step/

[46] J. C. Fang, H. P. Zeng, W. J. Xu, Z. Y. Zhao, and L. Wang, "Prediction of in-flight particle behaviors in plasma spraying Analysis and modelling," *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 18, no. 1-2, pp. 283–286, 3 2006.

[47] Y. Loo, S. K. Lim, G. Roig, and N.-M. Cheung, "FEW-SHOT REGRESSION VIA LEARNED BASIS FUNCTIONS," in *ICLR 2019*, Singapore, 2019.