Palestine Polytechnic University
College of Information Technology and Computer Engineering

Machine Learning Based Modeling and Optimization of Drilling Parameters of Developed Bio-Based Composites

Project Team :

Omar Shahin

Anas Doufesh

Bahaa Salamin

Oday Manasra

Supervisors :

Dr. Mohammed Abu Taha

Dr. Mahmoud Nassar

2022

ACKNOWLEDGMENT

First and foremost, all praise be to Allah the Almighty. For without his blessings, we couldn't do anything nor could we have completed this humble piece of work.

We would like to thank our supervisors, **Dr. Mohammed Abu Taha** and **Dr. Mahmoud Nassar**. For they were patient with us when we slacked. Strict when we needed discipline and guiding when we felt lost and confused. We would also like to thank our parents. They raised us and made us the young men that we are today. We wouldn't have reached this point in life without them and of course without the blessings of God.

We would like to thank our friends and colleagues. They were there for us in our lowest and darkest times. They were there when we needed them. Their support and words of encouragement helped us and gave us the strength to continue our work.

We would like to thank everyone that helped us even in the slightest to make us reach this point in our academic life and living. For those, we thank you all from the bottom of our hearts.

ABSTRACT

Numerical and Artificial Neural Network (ANN) approaches are now frequently employed for modeling and optimizing the performance of industrial systems. Optimal machining parameters are of major relevance in production contexts, as machining operations efficiency is critical to market competitiveness. The optimal machining parameters (i.e., spindle speed, drill diameter, and feed rate) for drilling operations will be researched in this project in order to minimize the delamination factor. Regression modeling and Response Surface Methodology (RSM) was previously used to explore the effects of specified parameters on process variables (i.e., delamination factors). The data acquired during the machining operation is utilized to create machine learning (ML)-based surrogate models that test, assess, and optimize different input machining parameters. To predict different output reactions of bio-composites drilling, several ML approaches such as polynomial regression (PR), random forest (RF) regression, gradient boosted (GB) trees, and adaptive boosting (AB) based regression are utilized. The ML results will then be compared to the experimental and RSM results.

Keywords: Artificial Neural Networks; Bio-Composites; Drilling; Machine Learning; Optimization;

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ML	Machine Learning
ROP	Rate of Penetration
ANN	Artificial Neural Networks
RSM	Response Surface Methodology
DF	Desirability Function
CFRP	Carbon Fiber Reinforced Plastic
ANOVA	Analysis of Variance
Av. abs. Error	Average Absolute Error
ε (%)	Absolute Percentage of Error
d	Tool Diameter (mm)
DoE	Design of Experiments
DPF/PP	Date Palm Fronds Reinforced Polypropylene
Ent.	Entry
Exp.	Experimental
Ext.	Exit
f	Feed Rate (mm/min)
F_d	Delamination Factor
MSE	Mean Square Error
HSS	High Speed Steel
PR	Polynomial Regression
GB	Gradient Boosted Trees
RF	Random Forest Regression
AB	AdaBoost

CHAPTER ONE: INTRODUCTION

1.1 Introduction

Modern technologies are supposed to aid the human race in various fields. Manufacturing is one of the most important fields in human life development. To achieve a higher level of development and modernity, humans should rely on new technologies that will help reach this goal. This research will discuss the use of modern technologies in order to achieve the goals set. Machine learning based models will be used to determine the best parameters for manufacturing drilling in bio composite materials.

Determining these parameters will save time, energy and cost. In finding out these parameters, drilling the bio composite materials will be more efficient and be at a better level of manufacturing. In bio-composite materials, the manufacturers face a very serious issue, delamination.

Delamination is when the several layers or components in a material separate. This presents an issue in drilling these materials. Some of these issues are lack of accuracy in holes that were drilled, more time spent in repairing the delamination and others.

In previous research (Nassar et al., 2021) ,2 machine learning based models were used to try and find the lowest delamination using ANN & RSM. This project will act as a continuous work based on the previous research. Machine learning based models will be trained with a given data sheet of the previous experiments to be the guideline of the training process. The results will be crucial in determining the optimal parameters in the drilling process to reduce the delamination in the bio composite material (DPF/PP) drilling. The new addition in this research is that more advanced and several other machine learning models will be used. By using them, it will be able not only to determine the drilling parameters but also to compare results between other models and discuss acquired results.

It will be observed how each model's algorithms works and how will it contribute to finding or determining the drilling parameters needed to decide on them.

The human error in determining the parameters is impractical. Regarding drilling and modern manufacturing, the most optimum results are expected with minimal error. The machine learning models will be able to determine them at a very high accuracy with very low error. After determining these errors, custom request of these parameters can be made to achieve the best results.

1.2 Motivation

In manufacturing, drilling is a very common and important process. Bio composite materials drilling is as well important, but an issue arises, delamination. Delamination is a serious issue when it comes to drilling bio composite materials. Achieving the lowest delamination is the key and the most important objective of this research.

In order to find delamination, drilling the bio composite material with each parameter is an unpractical way to solve the problem. Instead, by using some actual drilling with different drilling parameters to train machine learning models, finding delamination will be easier, faster and much more accurate.

Using machine learning models will provide a solution to the problem. This will happen by training the four models that will be used in the project to find the best parameters. In optimizing the drilling diameters for bio composite materials, more precise and efficient drilling can be used. After finding the best parameters, they can be sent to a manufacturing company to produce a customized drill with the parameters that have been found to obtain the best results in drilling.

As a result, less damage done to the bio composite materials, more precise drilling, lowering the time, costs and efforts exponentially in the pursuit of finding the optimal parameters by trial and error or going through thousands of parameter combinations to find them.

1.3 Problem Statement

Previously, an experimental design of drilling parameters using three distinct levels was accomplished, including drill bit diameter, feed rate and depth of cut. The delamination at the machined hole's entering and exiting is determined using Full Factorial Design. The parametric study and interaction were then statistically examined with ANOVA and Response Surface Methodology (RSM) based on quadratic regression .(Nassar et al., 2021) RSM was primarily used to analyze interactions between input parameters and their impact on machining quality. However, choosing the appropriate models to optimize the machining parameters is a vital role in getting the desired quality of holes and slots. The implementation of machine learning (ML) modeling assists in reducing the cost and time required to carry out the machining process while producing trustworthy outcomes. Hence, this work intends to use machine learning-based modeling to determine the optimal parameters of machined hole quality. Various ML models will be investigated and developed in order to determine the ideal parameters for achieving the desired quality of machined holes. The machine learning algorithms are: Polynomial Regression, Random Forest Tree Regression, Adaptive Boosted Trees & Gradient Boosted Trees. The results will then be compared to the experimental and statistical results.

1.4 Objectives

In simple words the main objective of this research is to reduce the delamination to the utmost minimum. That will be achieved by determining the best and most optimal drilling parameters used in the drilling and manufacturing of bio composite materials. This is the general and main objective, use machine learning models to determine the parameters so that the delamination is at its lowest.

- To develop ML based models for the drilling process of developed bio-composites, in order to find the lowest delamination in drilling and optimize the drilling parameters on the holed quality using ML models.

1.5 Project Activities

Requirements

The requirements needed for this project are as follows. A previous data sheet of results that will act as a guideline is needed for the work.(Nassar et al., 2021) This data will act as the input to the machine learning models to obtain the results needed. The algorithms for the machine learning model that were chosen must be acquired. These can be obtainable from the internet and coded in python programming language. Several research papers related to this topic to act as references and as a resource to work on and benefit from in writing the research paper. A laptop or computer to execute the work on it and run the models several times to obtain the best results.

Expected results

It is expected to gain reasonable results from the machine learning models that will be sufficient enough to satisfy said objectives. It is also expected that the parameter, feed rate will be the most influencing and determining factor in deciding the drilling parameters as it has been read in other research papers (Belaadi et al., 2020).

Contribution

The contribution will be using several machine learning-based models in determining the parameters for drilling. This will be able to give a much better insight on the whole topic. Different machine learning models will obviously lead to more than one result. Several results will make it possible to be able to compare them and study them to find if there are any relations between the results of the different machine learning models. Some models may provide a completely different result than the others, this will be due to the nature of the machine learning model or algorithm itself that will lead to different results than the other machine learning models.

After gaining the different results from the different models, the assumptions made can either turnout to be correct or not. An entire new perspective can appear that wasn't known about. The different models used will prove to be a beneficial point rather than be a liability.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

Several research studies examine how artificial intelligence can be used and exploited to develop methods that facilitate human work in several ways, such as reducing time, increasing accuracy, and decreasing the error rate.

2.2 Bio composite Materials

2.2.1 Definition

A bio composite material is a material made up of a matrix, which is usually resin, and a reinforcement element, which is usually natural fibers derived from plants or cellulose. These materials have a wide range of applications, including biomedicine, construction, and the development of these materials is founded on the idea that each of their components is biocompatible.(R., 2012)

2.2.2 Components of Bio composite Materials:

It is largely made of natural fibers generated primarily from plants or cellulose, with a matrix, usually resin, and a reinforcement element. The development of these materials is predicated on the fact that each of their components is biocompatible. These bio composite materials are defined by the fact that the resin matrix is essentially natural, therefore biodegradable, and we are talking about substances like glass fiber and carbon fiber. Instead, natural fibers such as wood fibers, flax, and others are commonly employed for the fibrous component (R., 2012).

In their structure, composite materials combine two or more starting materials while keeping them separate, resulting in a third type of material created by combining the two, which has chemical and physical properties that are significantly better than the characteristics of the starting materials individually, and Reinforced concrete is one of the most prevalent composite materials used in building. In fact, the steel and concrete elements of reinforced concrete interact with one another without losing their original material identity (R., 2012).

2.2.3 DPF/PP

Due to many advantages of using natural resources, natural fibers have been used recently as a method of providing added strength and ductility to reinforced polymer composites. This is mainly due to their availability, renewability, low density, cost effectiveness as well as satisfactory mechanical properties.

In this research, machine learning algorithms will be used to produce the best results for the drilling process on a class of bio-composites in which polypropylene (PP) is reinforced with palm fronds fibers.(Nassar et al., 2021)

2.2.4 Drilling of Bio composite Material

Damage caused by drilling bio composites differs greatly from damage caused by composites with synthetic reinforcement such as carbon or glass fibers. Induced damage increases with feed rate in composites reinforced with carbon or glass fibers, but damage decreases with feed rate in bio composites reinforced with natural fibers. (Díaz-Álvarez et al., 2021)

2.3 Drilling Process

2.3.1 Definition

Drilling is the most prevalent machining procedure used in the production of composite components and structures. For this, traditional drilling with a twist drill is commonly used. Due to their unique properties, however, ensuring good hole quality on composites is challenging. To increase the quality of drilled holes in composites, special drill bits with different geometries and materials, as well as unconventional hole-making techniques, have been created. Delamination is the most common drilling flaw that leads to composite construction failure. Various metrics or delamination factors are used to determine the severity of delamination damage. For many drilling applications, delamination suppression methods have been devised. (Zarif Karimi et al., 2013)

Composites have recently become key materials in a variety of engineering applications; as a result, various final industrial procedures, such as cutting, trimming, and drilling, are required to use these materials. Because composites are heterogeneous, as opposed to homogenous materials like metals and polymers, various problems have occurred while processing composites using typical cutting processes, such as significant surface roughness and material degradation at the cutting zone. Unconventional cutting procedures were investigated in order to overcome these obstacles. Cutting forces, which are the principal source of cutting flaws in typical cutting techniques, were not taken into consideration in unconventional cutting methods.(Díaz-Álvarez et al., 2021)

The most economical general-purpose drill material is high-speed steel (HSS). It is a cost-effective solution for many drilling applications. It is a popular metal drill bit material that can drill safely through several types of metals, plastic, and hardwood.(Nassar et al., 2021)

2.3.2 Drilling Scheme

In this research, drilling was performed using 2-flutes high speed steel (HSS) drill. HSS drill bits are hardwearing and heat resistant. The large concentrations of chrome and nickel in stainless steel ensure that HSS bits are super strong and durable. HSS drill bits are capable of safely drilling through hardwood, some metals and plastic. They're safe to operate at high speeds and provide long-lasting performance – as long as they are maintained. (Patil, 2020)

2.3.3 Delamination

Among all the manufacturing processes, the drilling process is one of the most used processes in the treatment of vital compounds, and one of the most serious problems that we face in the drilling process is Delamination, which is a major failure model in the drilling process. In addition to reducing the structural integrity of the material, also results in poor assembly tolerance and has the potential for long-term performance deterioration, **Figure 2-1** shows the Schematic mechanisms for entry (peel-up) and exit (push-out) delamination.

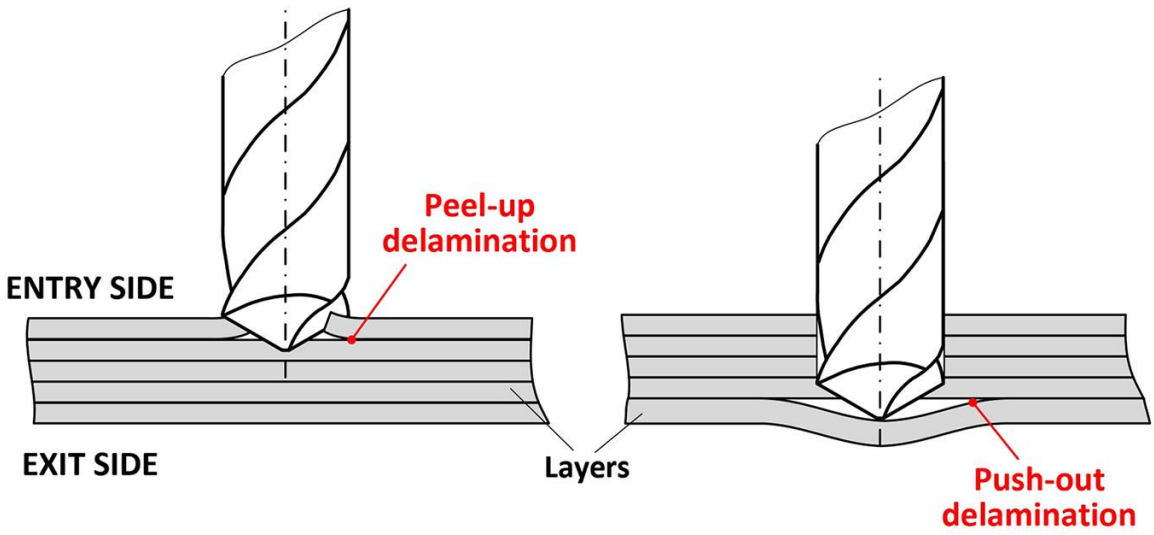


Figure 2-1 : Schematic mechanisms for entry (peel-up) and exit (push-out) delamination. (Barbosa et al., 2019).

Unloading is one of the most important problems in the drilling process that will lead to a decrease in the load-carrying capacity of the vehicles. Previous studies indicate that thrust is the main cause of the unloading process, which can be affected by tool geometry and drilling parameters. A high thrust force may cause a large-scale discharge of the workpiece. Both drilling type and feed rate have significant impacts on tool life, thrust, and, therefore, unloading. Besides, by changing the cutting speeds and feed rates, the average thrust force can be changed. In general, the final discharge factor is a synergistic result of cutting speed, drilling size, and feed rate (Zhang & Xu, 2021).

Delamination of the bio-composites at the hole entrance and exit during drilling has a negative impact on the hole surface quality and engineering qualities of the material. The feed rate was found to be the most beneficial parameter on hole entrance and exit delamination of the composite material using ANOVA. Using regression analysis, first-degree mathematical models for each cutting tool's entrance and exit delamination components were established. The acquired data revealed that optimization, mathematical models, and experimental test findings are all quite consistent (Bayraktar & Turgut, 2020).

- Factor of Delamination

The hole quality must be maintained properly when drilling composite materials, as this is the most important aspect of production. Surface finish, roundness error, hole diameter with tolerance, and other factors are used to assess hole quality. Delamination is a type of damage that occurs as a result of the anisotropy and brittleness of composite materials (Sasikumar, 2015).

The experimental results demonstrate that though the critical thrust force is higher with increasing wear, the delamination becomes more liable to occur because the actual thrust force increases with the wear to larger extent (Tsao & Hocheng, 2007).

2.4 Machine Learning

2.4.1 Definition

Machine learning, which is a subset of artificial intelligence, can make a machine learn automatically from past information without having to explicitly program (Paturi & Cheruku, 2020). Many machine learning algorithms have been developed for this purpose, and these algorithms have been exploited in many applications that facilitate human work, and among these applications is the use of machine learning models to determine the best parameters for manufacturing pits in bio composites (Belaadi et al., 2020) (Belaadi et al., 2020).

Machine learning (ML)-based approaches have exploded in popularity over the last decade, affecting a wide range of industries, including autonomous driving, health care, banking, manufacturing, energy harvesting, and more. ML, like computers in the 1980s and 1990s, is often regarded as one of the most disruptive technologies of our ages (Carleo et al., 2019).

A learning problem is defined as the inability to improve some measure of performance while completing a task through training. For example, learning to detect credit card fraud entails categorizing each credit card transaction as "fraud" or "non fraud." This fraud classifier's accuracy might be improved, and the training experience could include a collection of past credit-card transactions, each of which could be classified as fraudulent or not in hindsight. Alternatively, when "fraud" is improperly labeled "not fraud" rather than "fraud" is incorrectly labeled "not fraud," a separate performance indicator could be defined. It's also possible to define a new type of training experience (Jordan & Mitchell, 2015).

Machine learning aims to answer the question of how to create computers that learn on their own. It is one of the fastest-growing technical topics today, straddling the lines between computer science and statistics, as well as artificial intelligence and data science. The development of novel learning algorithms and theory, as well as the continual explosion in the availability of online data and low-cost computation, have fueled recent advances in machine learning. Data-intensive machine-learning methods are being adopted throughout science, technology, and commerce, resulting in greater evidence-based decision-making in a variety of fields, including health care, manufacturing, education, financial modeling, law enforcement, and marketing (Jordan & Mitchell, 2015).

A machine learning algorithm is a computing process that uses input data to accomplish a goal without being literally programmed (i.e., "hard coded") to do so. These algorithms are "soft programmed" in the sense that they automatically adjust or adapt their design as a result of repetition (i.e., experience) to get better and better at doing the target objective. Training is the adaptation process, which involves providing samples of input data together with intended consequences. The algorithm then optimizes itself so that it cannot only provide the desired result when given the training inputs, but also generalize to create the desired result when given new, previously unknown data. The "learning" aspect of machine learning is this training. The training does not have to be limited to a single adaption over a set period of time. A good algorithm, like people, may learn "lifelong" as it analyses fresh data and learns from its failures (Naqa & Murphy, 2015).

The recent success of machine learning has been underlined at initially by major improvements on previous technologies, such as image recognition. These advancements were largely the first indications of the impact that machine learning approaches can have on specialized activities. More recently, deep learning technology has effectively enabled applications that were previously unreachable to automated software. The presentation of reinforcement learning techniques in game play, for example, has had a significant impact on the idea that the entire field is moving closer to what was expected from a general artificial intelligence system (AI) (Carleo et al., 2019).

2.4.2 Mechanism of Work

The way in which machine-learning algorithms represent candidate programs (e.g., decision trees, mathematical functions, and general programming languages) and the way in which they search through this space of programs differ greatly (e.g., optimization algorithms with well-understood convergence guarantees and evolutionary search methods that evaluate successive generations of randomly mutated programs) (Jordan & Mitchell, 2015).

A computer algorithm can evolve in a variety of ways in response to training. The input data can be chosen and weighted to produce the best results. Iterative optimization can be used to alter the algorithm's variable numerical parameters. It can arrange a network of possible computational pathways for the best outcomes. It can take the supplied data to generate probability distributions and use them to forecast outcomes (Naqa & Murphy, 2015).

A key scientific and practical goal, regardless of the learning algorithm, is to conceptually characterize the capabilities of various learning algorithms as well as the intrinsic complexity of each given learning problem: How well can an algorithm learn from a specific type and quantity of training data? How resistant is the algorithm to errors in the training data or in its modeling assumptions? Is it possible to construct a successful solution for a learning issue with a given volume of training data, or is this learning problem

essentially intractable? Statistical decision theory and computational complexity theory are commonly used in such theoretical characterizations of machine-learning algorithms and issues. In fact, attempts to theoretically characterize machine-learning algorithms have resulted in a mix of statistical and computational theory with the goal of simultaneously characterizing the sample complexity (how much data is required to learn accurately) and the computational complexity (how much computation is required) and specifying how these depend on features of the learning algorithm like the representation it uses for what it learns. Optimization theory, with upper and lower bounds on rates of convergence of optimization processes combining well with the presentation of machine-learning issues as the optimization of a performance metric, has proven particularly valuable in recent years (Jordan & Mitchell, 2015).

2.5 Summary

Like some data-based machine learning algorithms that were used in previous studies such as RSM and ANN (Belaadi et al., 2020). In this research some other data-driven machine learning algorithms will be used to determine the best basic parameters of the drilling process in bio-composites (spindle speed, drilling diameter and feed rate), and compare the resulting results with practical data and conclude which of these algorithms yields the best results in terms of the lowest delamination and lowest error.

CHAPTER THREE: DESIGN/METHOD

3.1 Introduction

This chapter outlines the detailed procedures of the work. It consists of programming the four machine learning models, training them & obtaining results from said models. Then, the results are analyzed, graphed and discussed in detail later in chapter 4.

3.2 Project activities.

This section aims to provide a simple frame of the activates. The project is divided into two phases. The first phase is programming the models using Python programming language on Google Colab and training the models to predict the parameters. The second phase is acquiring the results from phase 1, analyzing it and sorting it into tables & graphs. **Figure3-1** showcases the work steps in order.

3.3 ML Algorithms

3.3.1 Polynomial Regression Algorithm (PR)

Polynomial Regression is a type of regression analysis in which the relationship between the independent and dependent variables is represented by an nth degree polynomial (Agrawal, 2021).

Polynomial Regression models are more convenient to be used with the method of least squares. The least square method minimizes the variance of the coefficients, under the Gauss Markov Theorem. The data in Polynomial Regression is fitted with a polynomial equation that has a curvilinear relationship between the dependent and independent variables.

3.3.1.1 When is Polynomial Regression needed?

When linear regression is applied and the linear line in the graph is nowhere near to cut the mean of the points. On the other hand, when the Polynomial Regression is applied, there is no need for a linear relationship between the independent and dependent variables in the dataset.

The figure(3-1) below show the difference between the Linear Regression and Polynomial Regression (PR).

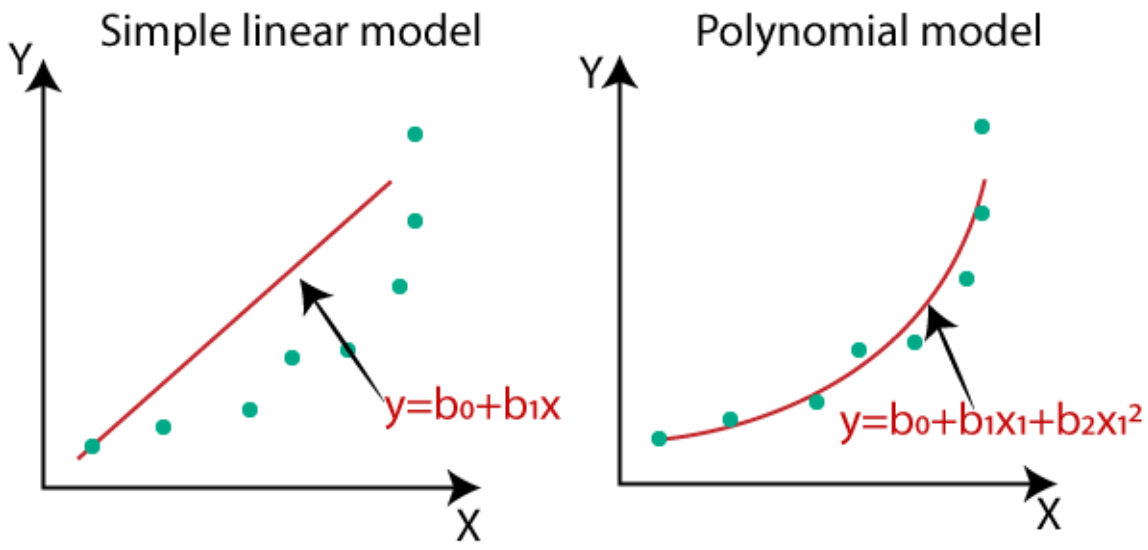


Figure 3-1 : Difference between the Linear Regression and Polynomial Regression(Agrawal, 2021)

Equations of the Polynomial Regression Model:

Simple Linear Regression equation: $y = b_0+b_1x$ **.....(1)**

Multiple Linear Regression equation: $y= b_0+b_1x_1+ b_2x_2+ b_3x_3+....+ b_nx_n$ **.....(2)**

Polynomial Regression equation: $y= b_0+b_1x + b_2x^2+ b_3x^3+....+ b_nx^n$ **....(3)**

When the three equations (1,2 & 3) are compared, it will be clear that the three equations are Polynomial equations, but differ by the degree of variables. The Simple and Multiple Linear equations are also Polynomial equations with a single degree. While the Polynomial regression equation is a Linear equation with the nth degree. So when a degree is added to the linear equations, then it will be converted into Polynomial Linear equations.

Polynomial regression's benefit is that it can be able to work with any data set size. It also works very well with non-linear problems. The major issue or disadvantage of this machine learning algorithm is that the user must apply the exact right polynomial degree for good bias variance tradeoff (Reddy, 2020) .

3.3.2 Random Forest Algorithm

Random forest regression algorithm is a supervised machine learning algorithm that is popularly and widely used in classification and regression problems. This algorithm is combined with a series of tree classifiers. Each tree casts a unit vote for the most popular class, then combining these results, the final sort result is obtainable. When the number of trees in the forest is greater, it leads to gaining a higher accuracy and prevents the problem of overfitting (Javapoint, 2021; Sruthi, n.d.).

3.3.2.1 Assumptions for Random Forest

The random forest combines many trees to predict the class of the dataset. Some trees' decisions may predict the correct output, while others may not. But together, all the trees predict the correct output. So, for a better random forest classifier there are two assumptions

The first assumption is that the feature variable of the dataset should include some actual values so that the classifier can predict accurate results rather than a guessed result. The second one is that the predictions from each tree must have very low correlations.

3.3.2.2 How does the Random Forest Algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision trees, and second is to make predictions for each tree created in the first phase.

- The Working process can be explained with the following steps:

    1. The first step is to select random K data points from the training set.
    2. The second step is to build the decision trees associated with the selected data points (subsets).
    3. The third step is choosing the number (N) for decision trees that need to be built.
    4. Then repeat steps number 1 and 2.
    5. The final step is to find the predictions tree for the new data points and assign it to the category that wins the majority votes (Javapoint, 2021) .

The deciding function is:

$$\mathbf{H(x)= \arg\max} \sum_{i=1}^{k} \boldsymbol{I(hi(x) = Y)} \quad \textbf{…….(4)}$$

H (x):combination of classification model.

hi: single decision tree model.

Y : the output variable.

I(·) : the indicator function

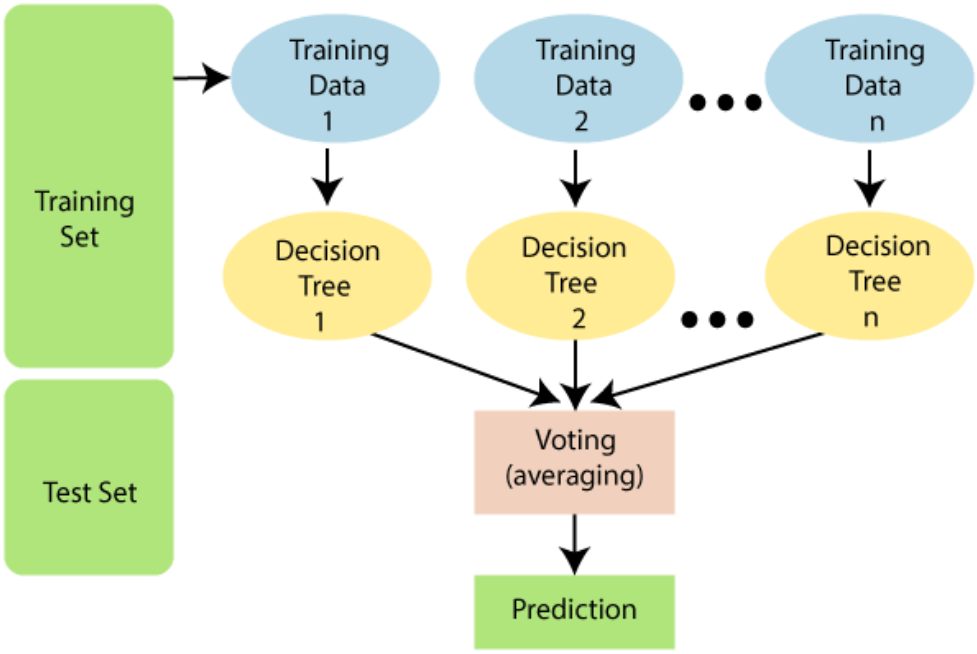


Figure 3-2 : Method of Random Forest Algorithm(Javapoint, 2021)

3.3.2.3 Advantages

There are many benefits to using random forest regression. It reduces the overfitting in decision trees and helps to improve the accuracy. It can be used for both classification and regression problems. It works well with both categorical and continuous values. It also can automatically handle missing values.

3.3.2.4 Disadvantages

The main limitation of random forest is that due to a large number of trees the algorithm takes a long time to train which makes it slow and ineffective for real-time predictions.

3.3.3 AdaBoost Algorithm

AdaBoost algorithm, short for Adaptive Boosting, This algorithm improves the prediction power by converting a number of weak learners into strong learners.

Boosting algorithm combines multiple models (weak learners) to reach the final output (strong learners).

3.3.3.1 Types of Boosting Algorithms

- Adaboost, gradient descent and xtreme gradient descent
- Adaboost or adaptive boosting is a technique used in machine learning used as an ensemble **method.**
- The most common algorithm used with adaboost is decision trees with one level It means that they are decision trees with only 1 split. These trees are also called decision stumps

This algorithm builds a model and gives equal weights to all the data points It then assigns higher weights to points that are wrongly classified. All the points with higher weights are given more importance to be focused on in the next model .

It will keep training models until and unless a lower error is received.

The whole point of this algorithm is assigning different weights to classify Wrong classifications get higher weight to focus on in the next model, we keep repeating this step until we reach a lower error and alpha will be a large number, in a perfect scenario the

alpha will be a very large number and the error is equal to or almost nearly 0 in a bad scenario alpha will be a negative integer and error is high Error is always between 1 and 0

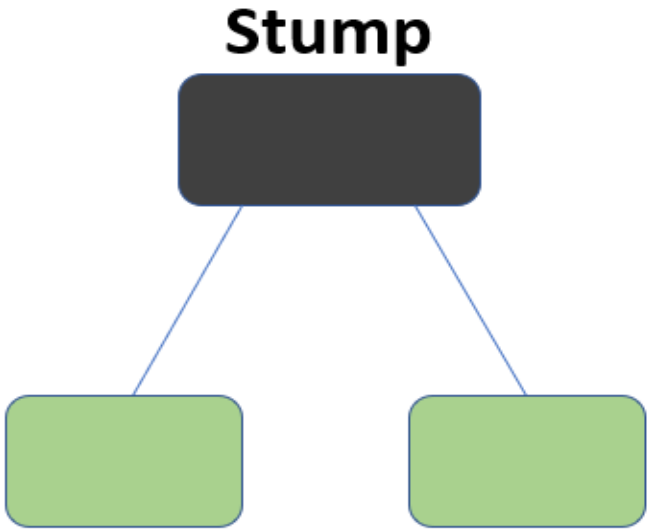


Figure 3-3 : Boosting Algorithm(Saini, 2021)

3.3.4 Gradient-Boosted Trees

Machine learning algorithms require more than just fitting models and making predictions to improve accuracy. Most of the industry or competition winning models use batch techniques or feature engineering to achieve better performance

Cluster technologies in particular have gained popularity due to their ease of use compared to feature engineering. There are multiple assembly methods that have been proven to increase accuracy when used with advanced machine learning algorithms. One such way is gradient enhancement. While gradation is often discussed as if it were a black box, so we will explain it.

3.3.4.1 What are Gradient-Boosted Trees?

Gradient-boosted trees is a machine learning technique for optimizing the predictive value of a model through successive steps in the learning process. Each iteration of the decision tree involves adjusting the values of the coefficients, weights, or biases applied to each of the input variables being used to predict the target value, with the goal of minimizing the loss function (the measure of difference between the predicted and actual target values). The gradient is the incremental adjustment made in each step of the process. boosting is a method of accelerating the improvement in predictive accuracy to a sufficiently optimum value.

Gradient boosting machines is a class of sophisticated machine-learning techniques that have had a lot of success in a variety of applications. it is very adaptable to the application's specific requirements. introduction to gradient boosting methods' methodology, with a significant emphasis on machine learning components of modeling. Will is supplemented by extensive examples and images that cover all steps of the gradient boosting model design.

3.3.4.2 Why are Gradient-Boosted Trees Important?

Gradient-boosted decision trees is a popular method for solving prediction problems in both classification and regression domains. The approach improves the learning process by simplifying the objective and reducing the number of iterations to get to a sufficiently optimal solution. Gradient-boosted models have proven themselves time and again in various competitions grading on both accuracy and efficiency, making them a fundamental component in the data scientist’s tool kit.

3.3.4.3 Methodology

One can arbitrarily specify both the loss function and the base-learner models on demand. In practice, given some specific loss function **Ψ (*y, f*)** and/or a custom base-learner ***h*(*x*, θ)**, the solution to the parameter estimates can be difficult to obtain. To deal with this, it was proposed to choose a new function *h* (*x*, θ) to be the most parallel to the negative gradient $\{g_t(x_i)\}_{i=1}^{N}$ along the observed data:

$$gt(x) = Ey[x]f(x) = \hat{f}^{t-1}(x) \qquad \text{.....(5)}$$

Instead of looking for the general solution for the boost increment in the function space, one can simply choose the new function increment to be the most correlated with $-g_t(x)$. This permits the replacement of a potentially very hard optimization task with the classic least-squares minimization one:

$$(\rho t, \theta t) = armin_{\rho,\theta} \sum_{i=1}^{N} [gt(xi) + \rho h(xi, \theta)]^2 \qquad \text{.....(6)}$$

To summarize, we can formulate the complete form of the gradient boosting algorithm, as originally proposed by Friedman (2001). The exact form of the derived algorithm with all the corresponding formulas will heavily depend on the design choices of ***Ψ(y, f)*** and ***h*(*x*, θ)**. One can find some common examples of these algorithms in Friedman (2001).

If we consider connections to earlier developments, it will turn out that the well-known cascade correlation neural networks by Friedman (2001) can be considered a special type of a gradient boosted model, as defined in Algorithm 5. Since the input-side weights of each neuron become fixed right after it was added to the network, this whole model can be considered a GBT, where the base-learner model is just one neuron and the loss function is the standard squared error. This algorithm also maximizes the correlation between the error of the whole network and the newly created neuron, which makes the comparison more evident.

3.3.4.4 Advantages and Disadvantages of Gradient Boost

Advantages of Gradient Boosting are

- it provides predictive accuracy that cannot be trumped.
- Flexible: can optimize on different loss functions and provides several hyper parameter tuning options that make the function fit very flexible.
- No data pre-processing required - often works great with categorical and numerical values as is Handles missing data - imputation not required

Now look at some disadvantages

- Gradient Boosting Models will continue improving to minimize all errors. This can overemphasize outliers and cause overfitting
- Computationally expensive - often require many trees (>1000) which can be time and memory exhaustive
- The high flexibility results in many parameters that interact and influence heavily the behavior of the approach (number of iterations, tree depth, regularization parameters, etc.). This requires a large grid search during tuning

Less interpretative in nature, although this is easily addressed with various tools

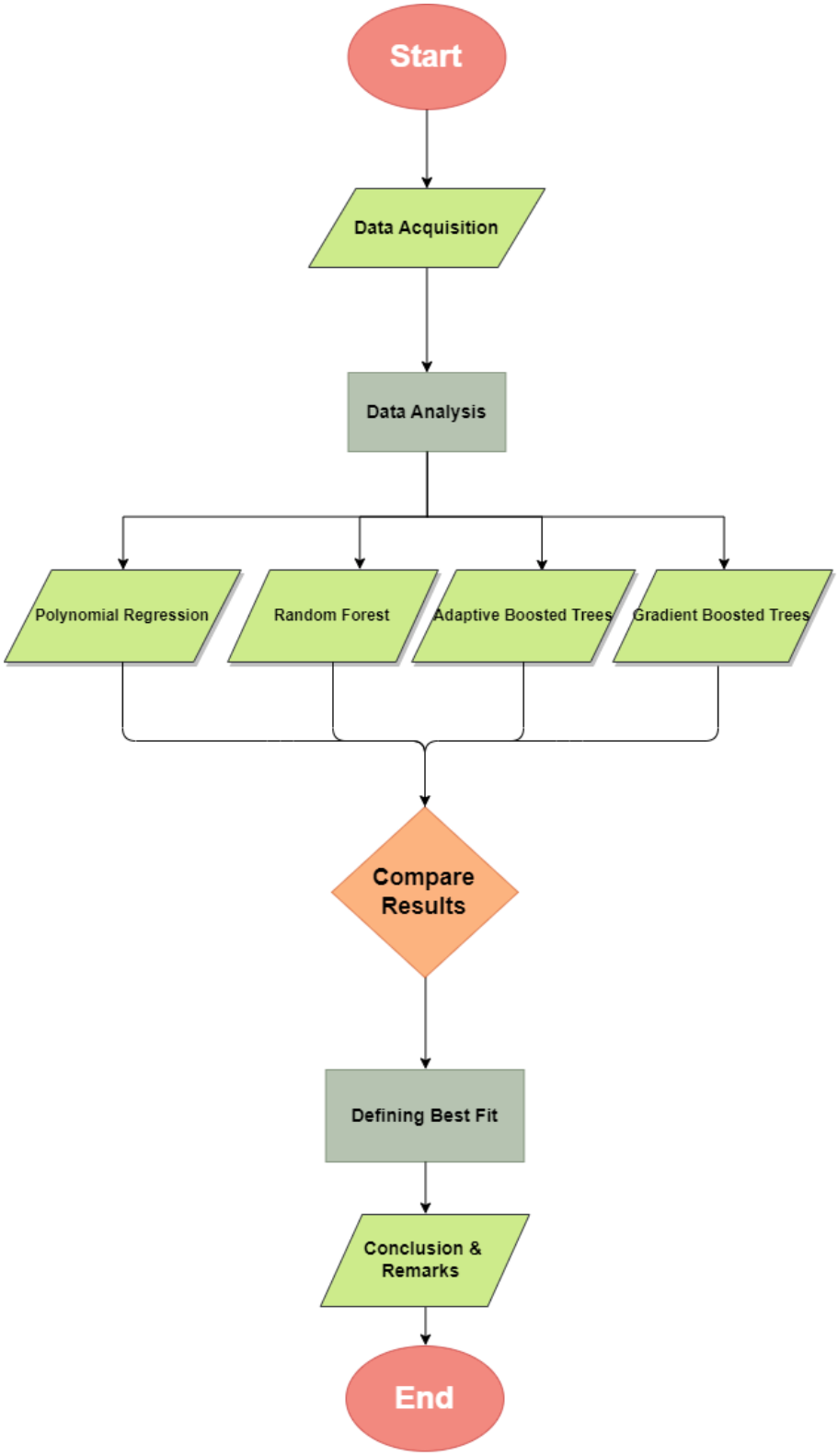


Figure 3-4 : This is a flowchart that shows the work flow of the study.

Firstly, the results obtained from Nassar's thesis (Nassar et al., 2021) will be the foundation to the whole study. It will act as the input to the machine learning models that will be used. As it is seen, there are 3 main parameters. Drill Bit Diameter, Spindle Speed & Feed Rate. Fd is the delamination factor, there is a factor for delamination in entry and exit. The machine learning models will predict 8 outputs. Each model will give prediction for entry and exit.

Table 3-1 : Drilling parameters and calculated delamination factors of developed bio-composites and pure PP.(Nassar et al., 2021)

Exp. Run	Control Drilling Parameters			Calculated Responses	
				F_d - Entry	F_d - Exit
	d **: Drill Bit Diameter (mm)**	***s*** **: Spindle Speed (rpm)**	***f*** **: Feed Rate (mm/min)**	**DPF/PP**	**DPF/PP**
1	4	1000	100	1.025	1.07
2	4	1000	200	1.045	1.068
3	4	1000	300	1.073	1.055
4	4	3000	100	1.07	1.053
5	4	3000	200	1.073	1.08
6	4	3000	300	1.078	1.058
7	4	5000	100	1.078	1.038
8	4	5000	200	1.095	1.065
9	4	5000	300	1.093	1.075
10	6	1000	100	1.035	1.038
11	6	1000	200	1.027	1.023
12	6	1000	300	1.035	1.03
13	6	3000	100	1.053	1.025
14	6	3000	200	1.058	1.033
15	6	3000	200	1.047	1.025
16	6	3000	200	1.053	1.015
17	6	3000	200	1.043	1.033
18	6	3000	200	1.053	1.035
19	6	3000	200	1.038	1.042
20	6	3000	300	1.052	1.058
21	6	5000	100	1.023	1.06
22	6	5000	200	1.025	1.05
23	6	5000	300	1.057	1.04
24	8	1000	100	1.033	1.031
25	8	1000	200	1.036	1.051
26	8	1000	300	1.02	1.059
27	8	3000	100	1.044	1.05
28	8	3000	200	1.039	1.045
29	8	3000	300	1.036	1.029
30	8	5000	100	1.053	1.044
31	8	5000	200	1.066	1.033
32	8	5000	300	1.055	1.038

The machine learning algorithms used are: Polynomial Regression, Random Forest Tree Regression, Adaptive Boosted Trees & Gradient Boosted Trees.

Google Colab will be used to program the models. Which is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning.

Google Colab is a hosted Jupyter Notebook service that requires no setup. The only thing needed is to upload the acquired results under the name “Data Input.csv”. CSV is short for comma separated value. Any excel file can be converted to a csv file which allows the researcher to work with it programming wise.

The entry and exit are split and assigned into two data frames. First data frame is “df1” which holds X_Entry & Y_Entry while the second data frame “df2” holds X_Exit & Y_Exit.

```
df1=df.drop(['Exit'],axis=1)
X_Entry = df1.drop(['Entry'],axis=1)
Y_Entry = df1['Entry']


df2=df.drop(['Entry'],axis=1)
X_Exit = df2.drop(['Exit'],axis=1)
Y_Exit = df2['Exit']
```

The data is then split into test and train.

```
#Dividing the data into test and train
X_entry_train, X_entry_test, y_entry_train, y_entry_test
= train_test_split(X_Entry, Y_Entry, test_size=0.33)
X_exit_train, X_exit_test, y_exit_train, y_exit_test
= train_test_split(X_Exit, Y_Exit, test_size=0.33)
```

The machine learning models are then programmed with a call for the entry and exit outputs for each model. Then, an expansion code will be done to find the most accurate and lowest delamination.

```
def expand_grid(dictionary):
   return pd.DataFrame([row for row in product(*dictionary.values
())],
                       columns=dictionary.keys())

dictionary = {'Drill Bit Diameter': np.linspace(6,7,100),
              'Spindle Speed': np.linspace(3600,3700,200),
              'Feed Rate': np.linspace(160,170,100)}

new_testdf=expand_grid(dictionary)
```

In this function, it will expand all values in the drilling diameters. The numbers inside the function are the value range desired. For example, in Drill Bit Diameter, the numbers 6,7 and 100 are shown. It means that it will generate 100 values between 6 and 7. The same thing goes for Spindle Speed and Feed Rate. The numbers in the picture are for example and do not necessarily mean in these ranges resides the lowest delamination. All these values are stored into a new data frame called “new_testdf”.

The way this benefits the study is that by sorting the output from the models, it can be seen where the delamination is at its lowest. Then, the parameters value is chosen and put into this function to find the lowest delamination in range of said parameter values.

```
y_pred_entry_gbt = model_entry_gbt.predict(new_testdf)
y_pred_exit_gbt = model_exit_gbt.predict(new_testdf)
new_testdf['Entry_Predictions']=y_pred_entry_gbt
new_testdf['Exit_Predictions']=y_pred_exit_gbt
```

In the previous code, new variables were defined. This code showcases an example for the Adaptive Boosted Trees machine learning model. The “y_pred_entry” and “y_pred_exit’ variables are used for tying the new data frame acquired earlier with the machine learning algorithms to find the lowest delamination value.

The way this benefits the study is that by sorting the output from the models, it can be seen where the delamination is at its lowest. Then, the parameters value is chosen and put into this function to find the lowest delamination in range of said parameter values. Then the data frame is sorted using the last line in the code to show the needed value.

This has been done for each of the models to find the lowest delamination value and specify the value of the parameters that hold said value.

```
new_testdf.to_csv(GradientBoosted_Range.csv')
gbt_entry.to_csv('GradientBoosted_Entry.csv')
gbt_exit.to_csv('GradientBoosted_Exit.csv')
```

In this code it shows the code used to download the output as .csv files in order to work with the data sheets and analyze the results separately using the suitable tools. This also has been for the models and this is an example for the adaboost algorithm. The “new_testdf” is the data frame acquired from the previous function. The other 2 files contain the results for the entry and exit predictions for said model.

CHAPTER FOUR: STATITCAL ANALYSIS AND DISCUSSION

4.1 Introduction

In this chapter, the results obtained from the machine learning based models are shown. They are analyzed and then the model with the best results is decided.

4.2 Measured Delamination Factor

In **Table 3-1**, it shows the data which was taken from similar previous research using different models.(Nassar et al., 2021) With which the expected results of the models will be compared and analyze the results of the models by calculating how close they are to these results, and **Table 3-1** shows Drilling parameters and calculated delamination factors of developed bio-composites.

4.3 Accuracy of Developed Models

The experimental and the predicted values obtained from regression models are compared. The percentage of error is calculated using the following formula for the mathematical model validation:

$$\%.\varepsilon = \frac{Exp.-Pred.}{Exp.} \times 100\% \quad (1)$$

Then, the developed model's adequacy is evaluated using the average absolute error calculations. When the average absolute error is less than 10%, it means that the model is accepted for further application.

4.4 DPF/PP Delamination Factors Models Accuracy

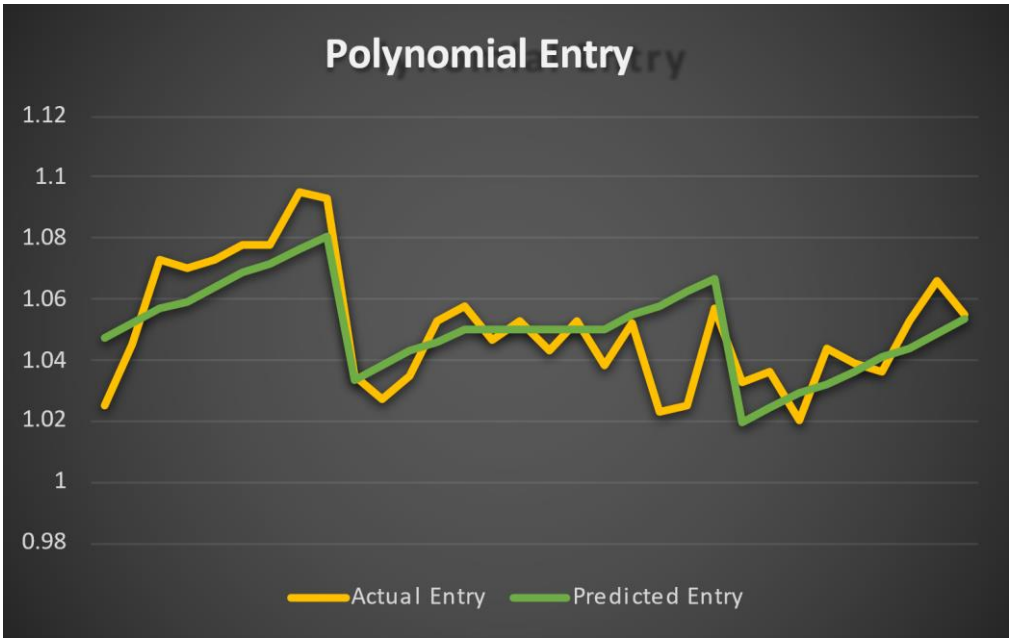
The DPF/PP *Fd* models are tests using the accuracy measurements and average absolute error. The results of the models will reveal that they can be used in further work to predict the responses as long as the error values are in the accepted ranges.

4.5 Predicted Result of Models

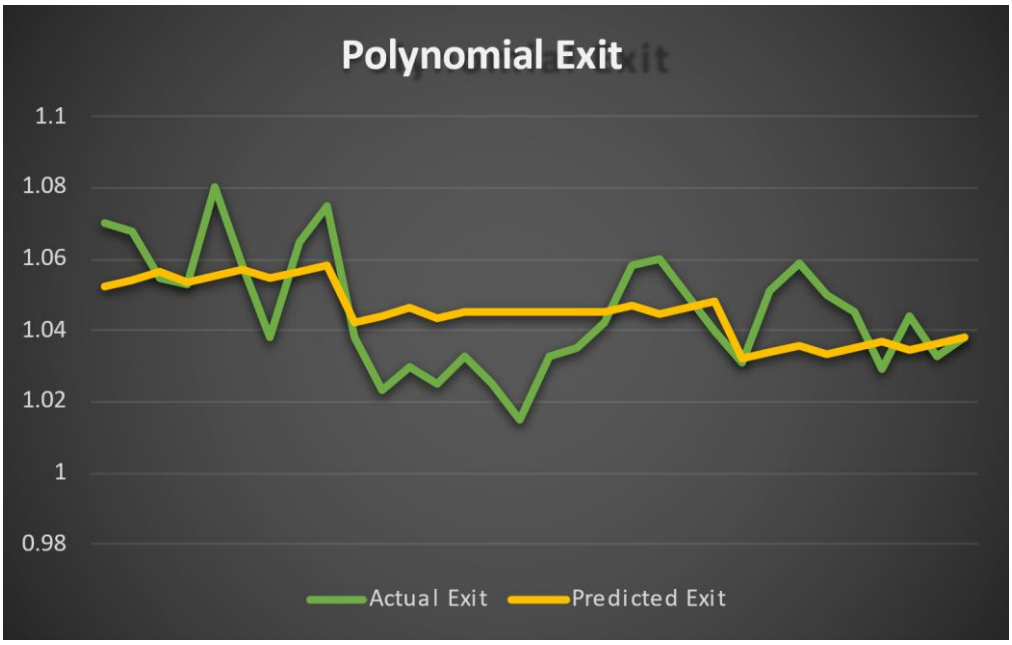
4.5.1 Predicted Results of Polynomial Regression Model

Table 4-1: Comparison of experimental and predicted values of DPF/PP delamination factors based on the Polynomial Regression Based Model

Control Drilling Parameters			**Predicted Response**					
d **:Drill Bit Diameter (mm)**	***s*****:Spindle Speed (rpm)**	***f*****:Feed Rate (mm/min)**	**F_d - Entry**			**F_d - Exit**		
			Exp.	**Pred.**	**ε (%)**	**Exp.**	**Pred.**	**ε (%)**
4	1000	100	1.025	1.047	2.146	1.16	1.053	9.224
4	1000	200	1.045	1.052	0.67	1.04	1.054	1.346
4	1000	300	1.073	1.057	1.491	1.06	1.056	0.377
4	3000	100	1.07	1.059	1.028	1.08	1.054	2.407
4	3000	200	1.073	1.064	0.839	1.02	1.055	3.431
4	3000	300	1.078	1.069	0.835	1.06	1.057	0.283
4	5000	100	1.078	1.071	0.649	1.05	1.055	0.476
4	5000	200	1.095	1.076	1.735	1.04	1.056	1.538
4	5000	300	1.093	1.081	1.098	1.03	1.058	2.718
6	1000	100	1.035	1.034	0.097	1.04	1.042	0.192
6	1000	200	1.027	1.038	1.071	1.04	1.044	0.385
6	1000	300	1.035	1.043	0.773	1.05	1.046	0.381
6	3000	100	1.053	1.046	0.665	1.08	1.043	3.426
6	3000	200	1.058	1.05	0.756	1.05	1.045	0.476
6	3000	200	1.047	1.05	0.287	1.05	1.045	0.476
6	3000	200	1.053	1.05	0.285	1.05	1.045	0.476
6	3000	200	1.043	1.05	0.671	1.05	1.045	0.476
6	3000	200	1.053	1.05	0.285	1.05	1.045	0.476
6	3000	200	1.038	1.05	1.156	1.05	1.045	0.476
6	3000	300	1.052	1.055	0.285	1.06	1.047	1.226
6	5000	100	1.023	1.058	3.421	1.08	1.044	3.333
6	5000	300	1.025	1.062	3.61	1.08	1.046	3.148
6	5000	200	1.057	1.067	0.946	1.06	1.048	1.132
8	3000	200	1.033	1.02	1.258	1.04	1.032	0.769
8	5000	200	1.036	1.025	1.062	1.02	1.034	1.373
8	3000	300	1.02	1.029	0.882	1.03	1.036	0.583
8	3000	100	1.044	1.032	1.149	1.06	1.033	2.547
8	5000	100	1.039	1.037	0.192	1.03	1.035	0.485
8	5000	300	1.036	1.041	0.483	1.06	1.037	2.17
8	1000	300	1.053	1.044	0.855	1.03	1.034	0.388
8	1000	200	1.066	1.049	1.595	1.04	1.036	0.385
8	1000	100	1.055	1.053	0.19	1.02	1.038	1.765
Av. abs. Error					**1.015%**			**1.511%**



a)



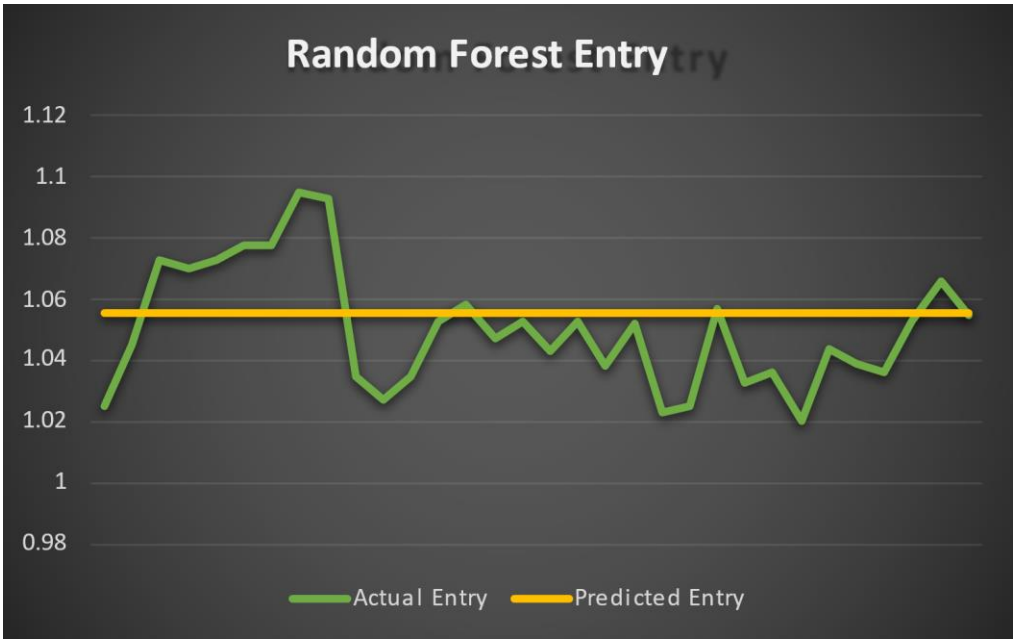
b)

Figure 4-1 : Comprsion of the exprrimental and calculated delamination factors of Polynomial Regression at Drill Bit Entry & Exit for DPF/PP **a)** F_d – Entry and **b)** F_d - Exit

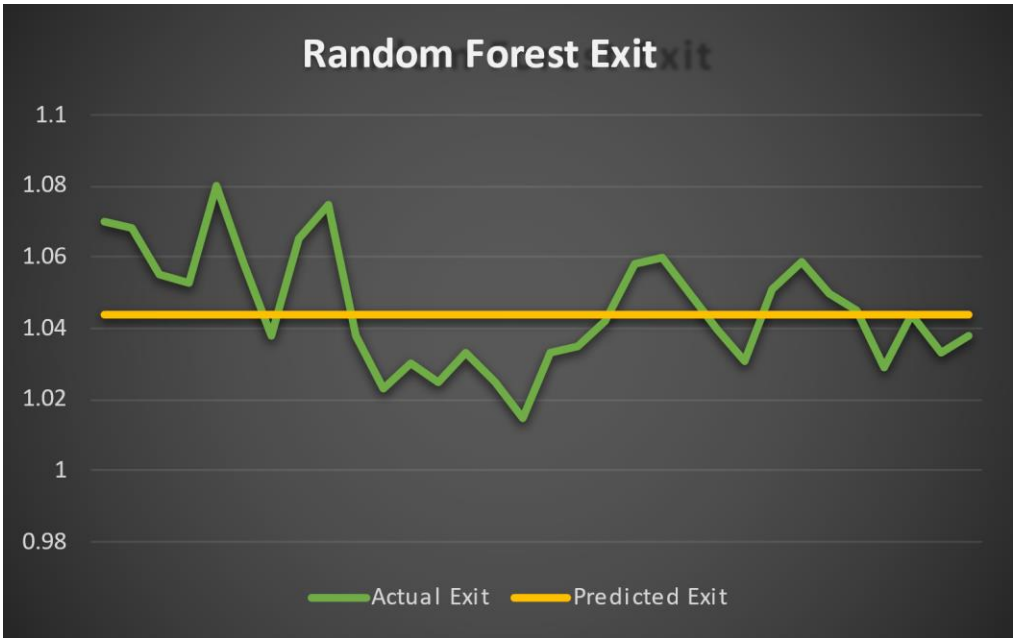
4.5.2 Predicted Results of Random Forest Model

Table 4-2: Comparison of experimental and predicted values of DPF/PP delamination factors based on Random Forest Based Model

Control Drilling Parameters			**Predicted Response**					
			F_d - Entry			**F_d - Exit**		
d **:Drill Bit Diameter (mm)**	***s*****:Spindle Speed (rpm)**	***f*****:Feed Rate (mm/min)**	**Exp.**	**Pred.**	**ε (%)**	**Exp.**	**Pred.**	**ε (%)**
4	1000	100	1.025	1.055	2.927	1.16	1.044	10
4	1000	200	1.045	1.055	0.957	1.04	1.044	0.385
4	1000	300	1.073	1.055	1.678	1.06	1.044	1.509
4	3000	100	1.07	1.055	1.402	1.08	1.044	3.333
4	3000	200	1.073	1.055	1.678	1.02	1.044	2.353
4	3000	300	1.078	1.055	2.134	1.06	1.044	1.509
4	5000	100	1.078	1.055	2.134	1.05	1.044	0.571
4	5000	200	1.095	1.055	3.653	1.04	1.044	0.385
4	5000	300	1.093	1.055	3.477	1.03	1.044	1.359
6	1000	100	1.035	1.055	1.932	1.04	1.044	0.385
6	1000	200	1.027	1.055	2.726	1.04	1.044	0.385
6	1000	300	1.035	1.055	1.932	1.05	1.044	0.571
6	3000	100	1.053	1.055	0.19	1.08	1.044	3.333
6	3000	200	1.058	1.055	0.284	1.05	1.044	0.571
6	3000	200	1.047	1.055	0.764	1.05	1.044	0.571
6	3000	200	1.053	1.055	0.19	1.05	1.044	0.571
6	3000	200	1.043	1.055	1.151	1.05	1.044	0.571
6	3000	200	1.053	1.055	0.19	1.05	1.044	0.571
6	3000	200	1.038	1.055	1.638	1.05	1.044	0.571
6	3000	300	1.052	1.055	0.285	1.06	1.044	1.509
6	5000	100	1.023	1.055	3.128	1.08	1.044	3.333
6	5000	300	1.025	1.055	2.927	1.08	1.044	3.333
6	5000	200	1.057	1.055	0.189	1.06	1.044	1.509
8	3000	200	1.033	1.055	2.13	1.04	1.044	0.385
8	5000	200	1.036	1.055	1.834	1.02	1.044	2.353
8	3000	300	1.02	1.055	3.431	1.03	1.044	1.359
8	3000	100	1.044	1.055	1.054	1.06	1.044	1.509
8	5000	100	1.039	1.055	1.54	1.03	1.044	1.359
8	5000	300	1.036	1.055	1.834	1.06	1.044	1.509
8	1000	300	1.053	1.055	0.19	1.03	1.044	1.359
8	1000	200	1.066	1.055	1.032	1.04	1.044	0.385
8	1000	100	1.055	1.055	0	1.02	1.044	2.353
Av. abs. Error					**1.582 %**			**1.617%**



a)



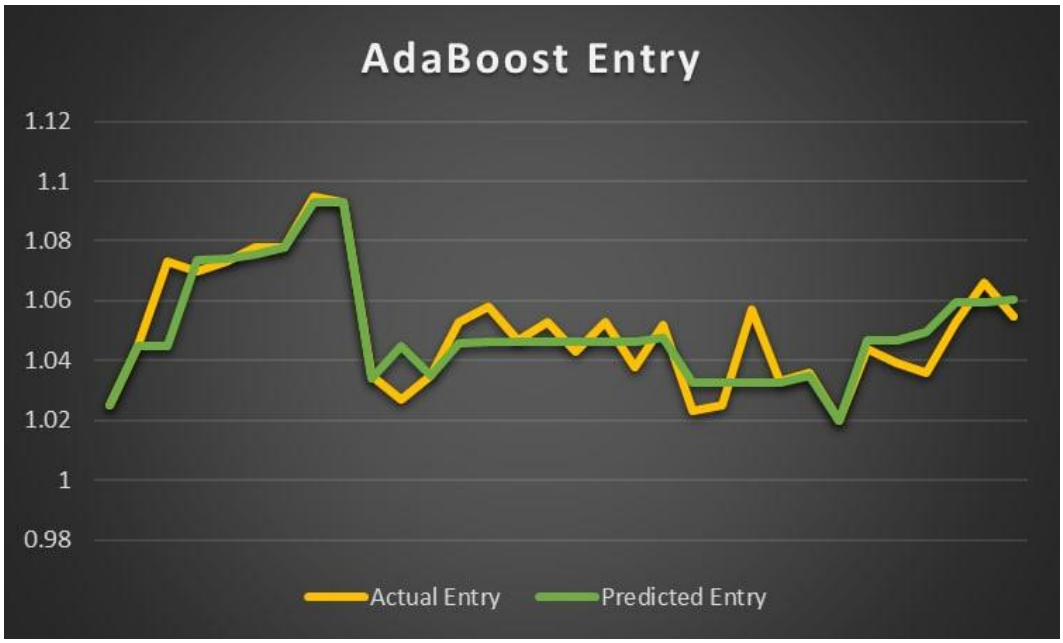
b)

Figure 4-2 : Comprsion of the exprrimental and calculated Delamination factors of Random Forest Model at Drill Bit Entry & Exit DPF/PP **a)** F_d – Entry and **b)** F_d - Exit

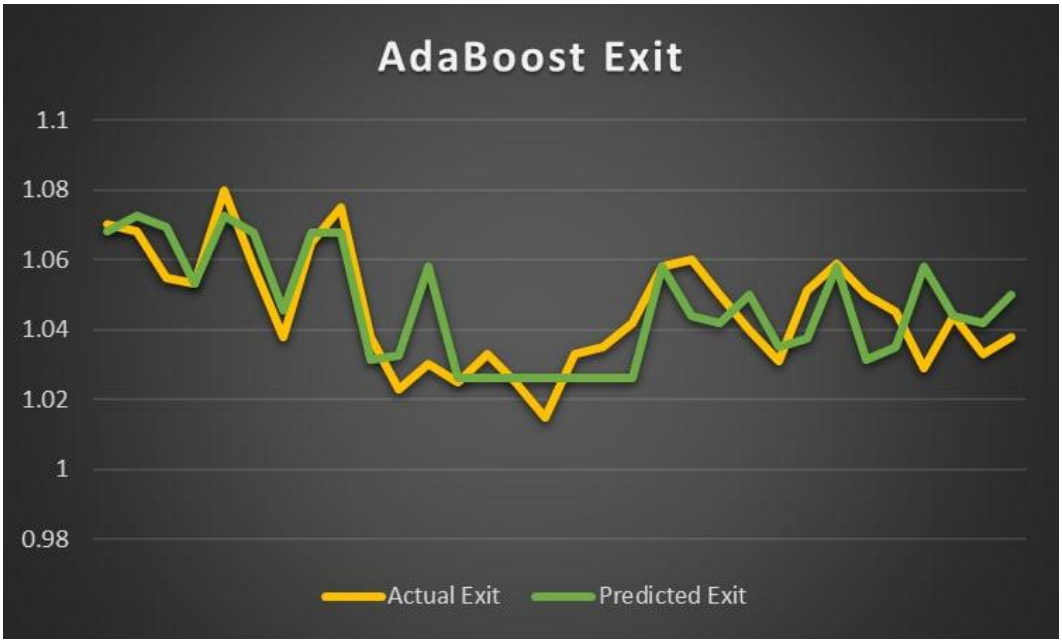
4.5.3 Predicted Results of AdaBoost Model

Table 4-3: Comparison of experimental and predicted values of DPF/PP delamination factors based on Adaptive Boosted Based Model

Control Drilling Parameters			**Predicted Response**					
			F_d - Entry			**F_d - Exit**		
d **:Drill Bit Diameter (mm)**	***s*****:Spindle Speed (rpm)**	***f*****:Feed Rate (mm/min)**	**Exp.**	**Pred.**	**ε (%)**	**Exp.**	**Pred.**	**ε (%)**
4	1000	100	1.025	1.025	0	1.16	1.068	7.931
4	1000	200	1.045	1.045	0	1.04	1.073	3.173
4	1000	300	1.073	1.045	2.61	1.06	1.069	0.849
4	3000	100	1.07	1.074	0.374	1.08	1.053	2.5
4	3000	200	1.073	1.074	0.093	1.02	1.073	5.196
4	3000	300	1.078	1.075	0.278	1.06	1.068	0.755
4	5000	100	1.078	1.078	0	1.05	1.046	0.381
4	5000	200	1.095	1.093	0.183	1.04	1.068	2.692
4	5000	300	1.093	1.093	0	1.03	1.068	3.689
6	1000	100	1.035	1.034	0.097	1.04	1.031	0.865
6	1000	200	1.027	1.045	1.753	1.04	1.032	0.769
6	1000	300	1.035	1.035	0	1.05	1.058	0.762
6	3000	100	1.053	1.046	0.665	1.08	1.026	5
6	3000	200	1.058	1.046	1.134	1.05	1.026	2.286
6	3000	200	1.047	1.046	0.096	1.05	1.026	2.286
6	3000	200	1.053	1.046	0.665	1.05	1.026	2.286
6	3000	200	1.043	1.046	0.288	1.05	1.026	2.286
6	3000	200	1.053	1.046	0.665	1.05	1.026	2.286
6	3000	200	1.038	1.046	0.771	1.05	1.026	2.286
6	3000	300	1.052	1.048	0.38	1.06	1.058	0.189
6	5000	100	1.023	1.032	0.88	1.08	1.044	3.333
6	5000	300	1.025	1.032	0.683	1.08	1.042	3.519
6	5000	200	1.057	1.032	2.365	1.06	1.05	0.943
8	3000	200	1.033	1.033	0	1.04	1.035	0.481
8	5000	200	1.036	1.035	0.097	1.02	1.038	1.765
8	3000	300	1.02	1.02	0	1.03	1.058	2.718
8	3000	100	1.044	1.047	0.287	1.06	1.031	2.736
8	5000	100	1.039	1.047	0.77	1.03	1.035	0.485
8	5000	300	1.036	1.05	1.351	1.06	1.058	0.189
8	1000	300	1.053	1.059	0.57	1.03	1.044	1.359
8	1000	200	1.066	1.059	0.657	1.04	1.042	0.192
8	1000	100	1.055	1.061	0.569	1.02	1.05	2.941
Av. abs. Error					**0.571%**			**2.160%**



a)



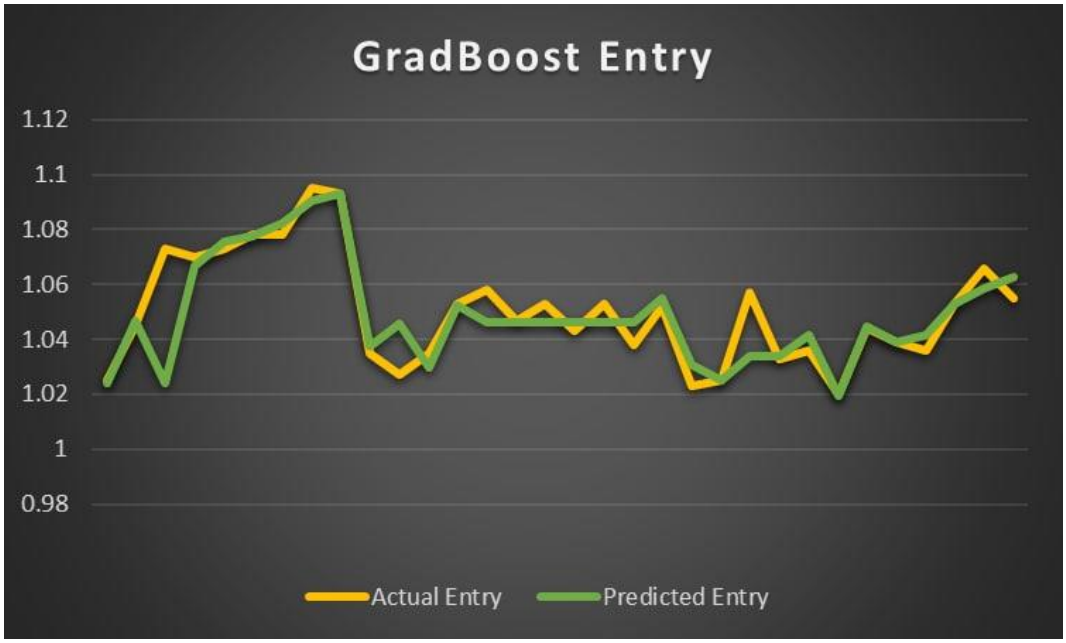
b)

Figure 4-3: Comprsion of the exprrimental and calculated Delamination Factors of AdaBoost Model at Drill Bit Entry & Exit for DPF/PP **a)** F_d – Entry and **b)** F_d - Exit

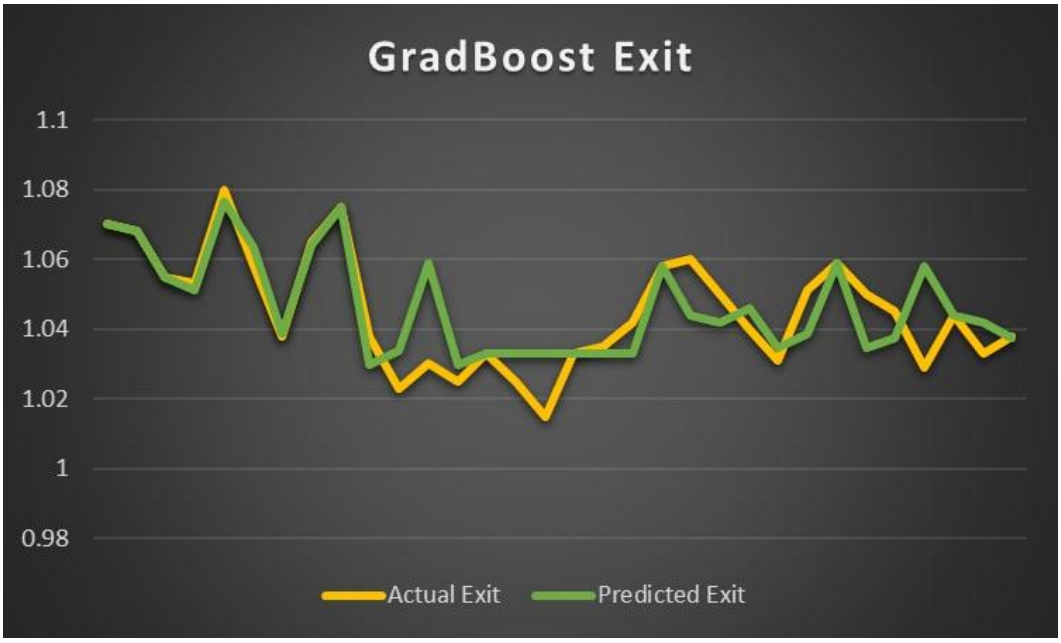
4.5.4 Predicted Results of Gradient-Boosted Trees Model

Table 4-4: Comparison of experimental and predicted values of DPF/PP delamination factors based on Gradient-Boosted Trees Based Model

Control Drilling Parameters			**Predicted Response**					
			F_d - Entry			**F_d - Exit**		
d **:Drill Bit Diameter (mm)**	***s*****:Spindle Speed (rpm)**	***f*****:Feed Rate (mm/min)**	**Exp.**	**Pr.**	**ε (%)**	**Exp.**	**Pr.**	**ε (%)**
4	1000	100	1.025	1.024	0.098	1.16	1.07	7.759
4	1000	200	1.045	1.047	0.191	1.04	1.068	2.692
4	1000	300	1.073	1.024	4.567	1.06	1.055	0.472
4	3000	100	1.07	1.067	0.28	1.08	1.051	2.685
4	3000	200	1.073	1.075	0.186	1.02	1.077	5.588
4	3000	300	1.078	1.078	0	1.06	1.063	0.283
4	5000	100	1.078	1.082	0.371	1.05	1.039	1.048
4	5000	200	1.095	1.091	0.365	1.04	1.064	2.308
4	5000	300	1.093	1.093	0	1.03	1.075	4.369
6	1000	100	1.035	1.037	0.193	1.04	1.03	0.962
6	1000	200	1.027	1.046	1.85	1.04	1.034	0.577
6	1000	300	1.035	1.03	0.483	1.05	1.059	0.857
6	3000	100	1.053	1.052	0.095	1.08	1.03	4.63
6	3000	200	1.058	1.046	1.134	1.05	1.033	1.619
6	3000	200	1.047	1.046	0.096	1.05	1.033	1.619
6	3000	200	1.053	1.046	0.665	1.05	1.033	1.619
6	3000	200	1.043	1.046	0.288	1.05	1.033	1.619
6	3000	200	1.053	1.046	0.665	1.05	1.033	1.619
6	3000	200	1.038	1.046	0.771	1.05	1.033	1.619
6	3000	300	1.052	1.055	0.285	1.06	1.058	0.189
6	5000	100	1.023	1.031	0.782	1.08	1.044	3.333
6	5000	300	1.025	1.025	0	1.08	1.042	3.519
6	5000	200	1.057	1.034	2.176	1.06	1.046	1.321
8	3000	200	1.033	1.034	0.097	1.04	1.034	0.577
8	5000	200	1.036	1.042	0.579	1.02	1.038	1.765
8	3000	300	1.02	1.019	0.098	1.03	1.059	2.816
8	3000	100	1.044	1.045	0.096	1.06	1.034	2.453
8	5000	100	1.039	1.039	0	1.03	1.038	0.777
8	5000	300	1.036	1.042	0.579	1.06	1.058	0.189
8	1000	300	1.053	1.053	0	1.03	1.044	1.359
8	1000	200	1.066	1.058	0.75	1.04	1.042	0.192
8	1000	100	1.055	1.063	0.758	1.02	1.038	1.765
Av. abs. Error					**0.578%**			**2.006%**



a)



b)

Figure 4-4 : Comprsion of the exprrimental and calculated Delamination Factors of Gradient-Boosted Trees Based Model at Drill Bit Entry & Exit for DPF/PP **a)** F_d – Entry and **b)** F_d – Exit

4.5.5 The most affective parameter in the models

The most affective parameter in predicting the delamination factor was **Drill Bit Diameter**. In this segment which was an output of the Polynomial Based Model it shows it was the most affective.

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.0642	0.014	78.602	0.000	1.036	1.092
Drill Diameter	-0.0069	0.002	-4.046	0.000	-0.010	-0.003
Spindle Speed	6e-06	1.7e-06	3.524	0.001	2.51e-06	9.49e-06
Feed Rate	4.722e-05	3.4e-05	1.387	0.176	-2.25e-05	0.000

As it shows, the drill bit diameter has the lowest error and it is the most affective.

4.6 The lowest delamination in the models.

The machine learning based models proved useful and efficient. The lowest delamination has been calculated in each model and the parameters were determined as follows.

4.6.1 Adaboost

These are the values of the parameters that predict the lowest delamination possible in AdaBoost Model.

Drill Diameter	Spindle Speed	Feed Rate	Entry	Exit
6	**1000**	**144.4444**	**1.034**	**1.031333**

4.6.2 Gradient Boosted

These are the values of the parameters that predict the lowest delamination possible in Gradient Boosted Model.

Drill Diameter	Spindle Speed	Feed Rate	Entry	Exit
6	**4111.111111**	**166.6667**	**1.025**	**1.041917**

4.6.3 Polynomial

These are the values of the parameters that predict the lowest delamination possible in Polynomial Model. As well as this model has predicted the lowest delamination factor value of all of the models.

Drill Diameter	Spindle Speed	Feed Rate	Entry	Exit
8	**1000**	**100**	**1.019844**	**1.032337**

4.6.4 Random Forest

Sadly, this model cannot determine the lowest delamination factor due to it failing in giving acceptable results.

4.7 Summary

The results show the value of the delamination factor that was predicted by the machine learning based models that were programmed.

- The Random Forest Model fails to give good data, and the reason is that it needs very large amounts of data in order to work well and give better, more accurate results. (Han et al., 2021)

- The Gradient-Boosted Trees Model provided the best and closest results in Entry to the actual values.

- The Polynomial Regression Model provided the best and closest results in Exit to the actual values.

- Drill Bit Diameter proved to be the most affective parameter based on the Polynomial Regression Based Model.

- Polynomial Regression Model predicted the lowest delamination factor out of all of the models.

CHAPTER FIVE: CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this project, all of the objectives have been met. First, developing new machine learning based models using Polynomial Regression, Random Forest Trees Regression, Gradient Boosted Trees & Adaptive Boosted Trees Regression algorithms. Second, determining the optimal drilling parameters to reduce delamination in bio composite materials (Date Palm Fronds Reinforced Polypropylene (DPF/PP)) drilling. The following are the main points that can be concluded based on this research:

- Drill Bit Diameter was the most affective parameter in predicting the values based on the Polynomial Regression Model.
- If more experimental data were provided , the predicted results of the models will be much better and closer to the experimental data.
- Random Forest didn't yield results that were acceptable due to the training data being small. (Han et al., 2021)
- The Gradient-Boosted Trees Model provided the best and closest Entry results to the actual values.
- The Gradient-Boosted Trees Model provided the best and closest Exit results to the actual values.
- The use of machine learning models in the drilling process effectively saves time and effort and gives much better results

5.2 Future Work

- To use other more advanced machine learning models to compare results with.
- Optimize the data using different methods like Zeroth Order Optimization.

REFERENCES :

Barbosa, L. F. F. M., Nascimento, A., Mathias, M. H., & de Carvalho, J. A. (2019). Machine Learning Methods Applied to Drilling Rate of Penetration Prediction and Optimization - A Review. *Journal of Petroleum Science and Engineering*, *183*(August), 106332. https://doi.org/10.1016/j.petrol.2019.106332

Bayraktar, Ş., & Turgut, Y. (2020). Determination of Delamination in Drilling of Carbon Fiber Reinforced Carbon Matrix Composites/Al 6013-T651 Stacks. *Measurement: Journal of the International Measurement Confederation*, *154*. https://doi.org/10.1016/j.measurement.2020.107493

Belaadi, A., Boumaaza, M., Amroune, S., & Bourchak, M. (2020). Mechanical Characterization and Optimization of Delamination Factor in Drilling Bidirectional Jute Fibre-Reinforced Polymer Biocomposites. *International Journal of Advanced Manufacturing Technology*, *111*(7–8), 2073–2094. https://doi.org/10.1007/s00170-020-06217-6

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., & Zdeborová, L. (2019). Machine Learning and the Physical Sciences. *Reviews of Modern Physics*, *91*(4), 45002. https://doi.org/10.1103/RevModPhys.91.045002

Díaz-Álvarez, A., Díaz-Álvarez, J., Feito, N., & Santiuste, C. (2021). Drilling of Biocomposite Materials: Modelling and Experimental Validation. *Simulation Modelling Practice and Theory*, *106*, 102203. https://doi.org/10.1016/j.simpat.2020.102203

Han, S., Williamson, B. D., & Fong, Y. (2021). Improving Random Forest Predictions in Small Datasets from Two-Phase Sampling Designs. *BMC Medical Informatics and Decision Making*, *21*(1), 1–9. https://doi.org/10.1186/s12911-021-01688-3

Jordan, M. I., & Mitchell, T. M. (2015). *Machine Learning: Trends, Perspectives, and Prospects*. *349*(6245).

Naqa, I. El, & Murphy, M. J. (2015). What is Machine Learning? *Machine Learning in Radiation Oncology*, 3–11. https://doi.org/10.1007/978-3-319-18305-3

Nassar, M. M. A., Alzebdeh, K. I., Alsafy, M. M. M., & Piya, S. (2021). *Analysis of the Machining Performance of Natural Fiber Composites using Fuzzy Logic and Artificial Neural Network Approaches*.

Patil, K. J. (2020). Process Capability Improvement of HSS Drill Bit Manufacturing Process. *International Journal for Research in Applied Science and Engineering Technology*, *8*(12), 5–18. https://doi.org/10.22214/ijraset.2020.32383

Paturi, U. M. R., & Cheruku, S. (2020). Application and Performance of Machine Learning Techniques in Manufacturing Sector from the Past Two Decades: A Review. *Materials Today: Proceedings*, *38*(xxxx), 2392–2401. https://doi.org/10.1016/j.matpr.2020.07.209

R., K. (2012). Biocomposite Materials. In *Composites and Their Applications*.

https://doi.org/10.5772/48302

Sasikumar, K. S. K. (2015). Optimization of Drilling Parameters on Delamination Based on Taguchi Method in Drilling of Natural Fiber Reinforced (Agave) Composite. *International Journal of Recent Technology and Engineering (IJRTE)*, *4*(1), 53–55.

Tsao, C. C., & Hocheng, H. (2007). Effect of Tool Wear on Delamination in Drilling Composite Materials. *International Journal of Mechanical Sciences*, *49*(8), 983–988. https://doi.org/10.1016/j.ijmecsci.2007.01.001

Zarif Karimi, N., Heidary, H., Minak, G., & Ahmadi, M. (2013). Effect of the Drilling Process on the Compression Behavior of Glass/Epoxy Laminates. *Composite Structures*, *98*, 59–68. https://doi.org/10.1016/j.compstruct.2012.10.044

Zhang, Y., & Xu, X. (2021). Predicting the Delamination Factor in Carbon Fibre Reinforced Plastic Composites During Drilling Through the Gaussian Process Regression. *Journal of Composite Materials*, *55*(15), 2061–2068. https://doi.org/10.1177/0021998320984245

APPENDICES

Polynomial Regression Model Code

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt


df=pd.read_csv("Data Input.csv")
df.head()


df1=df.drop(['Exit'],axis=1)
X_Entry = df1.drop(['Entry'],axis=1)
Y_Entry = df1['Entry']


df2=df.drop(['Entry'],axis=1)
X_Exit = df2.drop(['Exit'],axis=1)
Y_Exit = df2['Exit']


X_entry_train, X_entry_test, y_entry_train, y_entry_test = train_tes
t_split(X_Entry, Y_Entry, test_size=0.33)
X_exit_train, X_exit_test, y_exit_train, y_exit_test = train_test_sp
lit(X_Exit, Y_Exit, test_size=0.33)


#Polynomial regression
def polyreg(X,Y):
  X1 = sm.add_constant(X)
  model= sm.OLS(Y, X1).fit()
  print(model.summary())
  y_pred=model.predict(X1)
  X2=X
  X2['Actual']=Y
  X2['Predicted']=y_pred
  print("Error is: {}".format(mean_squared_error(Y,y_pred)))
```

```
  return model,X2


model_entry_poly,reg_entry = polyreg(X_Entry,Y_Entry)
reg_entry



model_exit_poly,reg_exit=polyreg(X_Exit,Y_Exit)
reg_exit


def expand_grid(dictionary):
   return pd.DataFrame([row for row in product(*dictionary.values())
],
                       columns=dictionary.keys())

dictionary = {'Drill Bit Diameter': np.linspace(6,8,10),
              'Spindle Speed': np.linspace(1000,5000,10),
              'Feed Rate': np.linspace(100,300,10)}

new_testdf=expand_grid(dictionary)



y_pred_entry_poly = model_entry_poly.predict(new_testdf)
y_pred_exit_poly = model_exit_poly.predict(new_testdf)
new_testdf['Entry_Predictions']=y_pred_entry_poly
new_testdf['Exit_Predictions']=y_pred_exit_poly


new_testdf.to_csv('Polynomial_Range_result.csv')
reg_entry.to_csv('Polynomial_Entry.csv')
reg_exit.to_csv('Polynomial_Exit.csv')
```

Random Forest Model Code

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt


df=pd.read_csv("Data Input.csv")
df.head()


df1=df.drop(['Exit'],axis=1)
X_Entry = df1.drop(['Entry'],axis=1)
Y_Entry = df1['Entry']


df2=df.drop(['Entry'],axis=1)
X_Exit = df2.drop(['Exit'],axis=1)
Y_Exit = df2['Exit']

X_entry_train, X_entry_test, y_entry_train, y_entry_test = train_tes
t_split(X_Entry, Y_Entry, test_size=0.33)
X_exit_train, X_exit_test, y_exit_train, y_exit_test = train_test_sp
lit(X_Exit, Y_Exit, test_size=0.33)


def randomreg(X,Y):
  X_train, X_test, y_train, y_test = train_test_split(X, Y, test_siz
e=0.33,random_state = 0)
  scaler = StandardScaler()
  scaler.fit(X_train)
  X_train = scaler.transform(X_train)
  X_test = scaler.transform(X_test)
  estimators = np.arange(10, 200, 10)
  scores = []
  for n in estimators:
    model = RandomForestRegressor(n_estimators = n).fit(X_train,y_tr
ain)
    scores.append(model.score(X_test, y_test))
```

```
  model = RandomForestRegressor(n_estimators = 200).fit(X_train,y_tr
ain)
  y_pred = model.predict(X)
  X1 = X

  X1['Actual']=Y
  X1['Predicted']=y_pred

  plt.title("Effect of n_estimators")
  plt.xlabel("n_estimator")
  plt.ylabel("score")
  plt.plot(estimators, scores)
  print("Error is: {}".format(mean_squared_error(y_test,model.predic
t(X_test))))
  return model,X1


model_entry,random_entry=randomreg(X_Entry,Y_Entry)
random_entry


model_exit,random_exit=randomreg(X_Exit,Y_Exit)
random_exit


def expand_grid(dictionary):
   return pd.DataFrame([row for row in product(*dictionary.values())
],
                       columns=dictionary.keys())

dictionary = {'Drill Bit Diameter': np.linspace(6,8,10),
              'Spindle Speed': np.linspace(1000,5000,10),
              'Feed Rate': np.linspace(100,300,10)}

new_testdf=expand_grid(dictionary)

y_pred_entry=model_entry.predict(new_testdf)
y_pred_exit=model_exit.predict(new_testdf)
new_testdf['Entry_Predictions']=y_pred_entry
new_testdf['Exit_Predictions']=y_pred_exit



new_testdf.to_csv('RandomForest_Range_result.csv')
random_entry.to_csv('RandomForest_Entry.csv')
random_exit.to_csv('RandomForest_Exit.csv')
```

AdaBoost Model Code

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.ensemble import AdaBoostRegressor


df=pd.read_csv("Data Input.csv")
df.head()



df1=df.drop(['Exit'],axis=1)
X_Entry = df1.drop(['Entry'],axis=1)
Y_Entry = df1['Entry']


df2=df.drop(['Entry'],axis=1)
X_Exit = df2.drop(['Exit'],axis=1)
Y_Exit = df2['Exit']



X_entry_train, X_entry_test, y_entry_train, y_entry_test = train_tes
t_split(X_Entry, Y_Entry, test_size=0.33)
X_exit_train, X_exit_test, y_exit_train, y_exit_test = train_test_sp
lit(X_Exit, Y_Exit, test_size=0.33)


def adareg(X,Y):
  X_train, X_test, y_train, y_test = train_test_split(X, Y, test_siz
e=0.33,random_state = 0)
  model = AdaBoostRegressor().fit(X_train,y_train)
  y_pred = model.predict(X)
  X1=X
  X1['Actual']=Y
  X1['Predicted']=y_pred
  print("Error is: {}".format(mean_squared_error(y_test,model.predic
t(X_test))))
  return model,X1
```

```
model_ada_exit,ada_exit=adareg(X_Exit,Y_Exit)
ada_exit



def expand_grid(dictionary):
   return pd.DataFrame([row for row in product(*dictionary.values())
],
                       columns=dictionary.keys())

dictionary = {'Drill Bit Diameter': np.linspace(6,8,10),
              'Spindle Speed': np.linspace(1000,5000,10),
              'Feed Rate': np.linspace(100,300,10)}

new_testdf=expand_grid(dictionary)



y_pred_entry_ada=model_ada_entry.predict(new_testdf)
y_pred_exit_ada=model_ada_exit.predict(new_testdf)
new_testdf['Entry_Predictions']=y_pred_entry_ada
new_testdf['Exit_Predictions']=y_pred_exit_ada

new_testdf.to_csv('AdaBoost_Range_result.csv')
ada_entry.to_csv('AdaBoost_Entry.csv')
ada_exit.to_csv('AdaBoost_Exit.csv')
```

Gradient-Boosted Trees Model Code

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.ensemble import GradientBoostingRegressor

df=pd.read_csv("Data Input.csv")
df.head()



df1=df.drop(['Exit'],axis=1)
X_Entry = df1.drop(['Entry'],axis=1)
Y_Entry = df1['Entry']


df2=df.drop(['Entry'],axis=1)
X_Exit = df2.drop(['Exit'],axis=1)
Y_Exit = df2['Exit']

#Dividing the data into test and train
X_entry_train, X_entry_test, y_entry_train, y_entry_test = train_tes
t_split(X_Entry, Y_Entry, test_size=0.33)
X_exit_train, X_exit_test, y_exit_train, y_exit_test = train_test_sp
lit(X_Exit, Y_Exit, test_size=0.33)



def gbtree(X,Y):
  X_train, X_test, y_train, y_test = train_test_split(X, Y, test_siz
e=0.33,random_state = 0)
 params={'n_estimators':3,'max_depth':3,'learning_rate':1,'criterion
':'mse'}
 model = GradientBoostingRegressor(**params).fit(X_train,y_train)
  y_pred = model.predict(X)
  X1=X
  X1['Actual']=Y
  X1['Predicted']=y_pred
```

```
  print("Error is: {}".format(mean_squared_error(y_test,model.predic
t(X_test))))
  return model,X1


model_entry_gbt,gbt_entry=gbtree(X_Entry,Y_Entry)
gbt_entry


model_exit_gbt,gbt_exit=gbtree(X_Exit,Y_Exit)
gbt_exit


def expand_grid(dictionary):
   return pd.DataFrame([row for row in product(*dictionary.values())
],
                       columns=dictionary.keys())

dictionary = {'Drill Bit Diameter': np.linspace(6,8,10),
              'Spindle Speed': np.linspace(1000,5000,10),
              'Feed Rate': np.linspace(100,300,10)}

new_testdf=expand_grid(dictionary)



y_pred_entry_gbt = model_entry_gbt.predict(new_testdf)
y_pred_exit_gbt = model_exit_gbt.predict(new_testdf)
new_testdf['Entry_Predictions']=y_pred_entry_gbt
new_testdf['Exit_Predictions']=y_pred_exit_gbt


new_testdf.to_csv('GradientBoosted_Range.csv')
gbt_entry.to_csv('GradientBoosted_Entry.csv')
gbt_exit.to_csv('GradientBoosted_Exit.csv')
```