PHOTOCOPY AND USE AUTHORIZATION

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission for extensive copying of my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature _____

Date _____

Artificial Intelligence Based Li-ion Battery Diagnostic Platform

by

Shovan Chowdhury

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in the Department of Mechanical Engineering

Idaho State University

Summer 2022

COMMITTEE APPROVAL

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of Shovan Chowdhury find it satisfactory and recommend that it be accepted.

Dr. Leslie Kerby
Committee Chair

Dr. Marco P. Schoen
Committee Member

Dr. Shu-Chuan 'Grace' Chen
Graduate Faculty Representative

DEDICATION

This thesis is dedicated to **my father**.

ACKNOWLEDGMENTS

First of all, I would like to thank to Almighty God for everything.

I would like to express my deepest gratitude to my advisor, Dr. Leslie Kerby, whose sincerity and encouragement I will never forget. Dr. Kerby has been an inspiration as I hurdled through the path of this Masters degree. I still remember the day when I got an email from Dr. Leslie Kerby after completing her Applied Neural Network course. The subject line of that email was "Neural Network Rockstar". She told me that "I am gifted at Data Science and should continue to pursue it". That's what I did and that is how she encouraged me. She is the epitome of leadership and the ideal role model for me. She taught me how to overcome the problem faced in research.

I am indebted to my another mentor Dr. Marco Schoen. As a mechanical engineer, I was not familiar with any kind of data science and machine learning stuff when I started my masters. In my first semester, I took a course from Dr. Schoen which was Intelligent Control System. I learned lots of stuff amazing stuff from that course and felt that "yes, that's my passion". He taught me that learning new things can be pleasurable. Thanks to him for his continued guidance and an endless supply of fascinating projects. His unassuming approach to research and science is a source of inspiration.

I also would like to thanks Dr. Shu-Chuan Chen for teaching nicely in her Applied Regression course. I have applied lots of stuff from her course in this thesis. Thanks to Dr. Ken Bosworth for his splendid teaching in math courses. I would like to thanks Dr. Alba Perez Gracia for picking me from Bangladesh to here with full funding. I can never forget that I started my research career under her supervision.

I am grateful for my parents' unwavering love and support, which keeps me driven and self-assured. They helped me achieve my goals and success because they believed in me. My siblings, who keep me grounded, remind me of what matters in life, and are always supportive of my endeavors, deserve my gratitude. Finally, I owe my deepest gratitude to Purba, who is my love. I will be eternally grateful for the unwavering love and support I received from her during the thesis

process and every day.

I am grateful also to several ISU students. I must need to mention one name "Golam Gause Zaman". I used to discuss any kind of problem related to study and life with him throughout my masters life.

I would like to thank K. A. Severson, P. M. Attia, William Chueh, and their co-authors for their generously providing the data used in their study for the use in this work. This work was supported under Idaho National Laboratory's Laboratory Directed Research and Development program (LDRD No. 19P45-013FP). The INL is operated by Battelle Energy Alliance under Contract No. DE-AC07-05ID14517 for the U.S. Department of Energy. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

I am grateful for all of the assistance I have received from everyone in order to attain this goal.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Battery performance datasets are typically non-normal and multicollinear. Extrapolating such datasets for model predictions needs attention to such characteristics. This study explores the impact of data normality in building machine learning models. In this work, tree-based regression models and multiple linear regressions models are each built from a non-normal dataset with multicollinearity and compared. Several techniques are necessary, such as data transformation, to achieve a good multiple linear regression model with this dataset; the most useful techniques are discussed. With these techniques, the best multiple linear regression model achieved an R^2 = 81.23% and exhibited no multicollinearity effect for the dataset used in this study. Tree-based models perform better on this dataset, as they are non-parametric, capable of handling complex relationships among variables and not affected by multicollinearity. I show that bagging, in the use of Random Forests, reduces overfitting. Our best tree-based model achieved accuracy of R^2 = 97.73%. This study explains why tree-based regressions promise as a machine learning model for non-normally distributed, multicollinear data. This work applies machine learning tools to achieve the early life prediction of li-ion battery life. The prediction accuracy of different machine learning algorithms are compared in the battery database. Among various algorithms, the random forest (RF) method exhibits the highest accuracy of 97.73% to predict the battery cycle life using early cycle discharge capacity. The best model predicts battery cycle life with 4.05% test error when battery reaches 97% of nominal capacity and 9.69% test error when battery reaches 99% of nominal capacity.

Keywords: Machine Learning, Random Forest, Battery Diagnostic, Lifetime prediction, Lithium-ion Battery, Data driven method, Decision Tree, Data Transformation, Stepwise Regression

Chapter 1. Introduction

Batteries enable portability and convenience. Wireless electronics such as cellphones, computers, watches, and remote control devices rely on batteries as their primary power source. Batteries are used in a wide range of portable equipment, including medical gadgets, lawn mowers, kitchen appliances, shop tools, and automobiles. Because of its high energy density and inexpensive maintenance, the lithium ion (Li-ion) battery is currently the most popular among batteries. Li-ion batteries will dramatically reduce greenhouse gas emissions if electric vehicles (EVs) replace the majority of gasoline-powered transportation.[1]. But there are some disadvantages of this battery. One of the major disadvantages for consumer electronics is that lithium ion batteries suffer from aging. Aging is dependent on the charge-discharge cycle that the battery has undergone. A Li-ion battery's cycle life is considered reached when discharge capacity drops below 80% of its initial value. It is not possible for manufacturers to measure the cycle life of a battery in its initial condition. Often, batteries reach their cycle life within 300-500 discharge cycles. The frequent need for battery replacement becomes a problem for technology, especially in applications such as electric vehicles. One cannot change the battery of electric vehicles often as this is not cost effective. To diminish this problem, a method which can detect battery cycle life by observing early discharge cycles would be beneficial. In this study, a machine learning approach was developed which can predict the life cycle of a battery by observing the early discharge cycles of the battery. Furthermore, the performance data set for battery is usually non-normally distributed and has multicollinearity. In this work, I also attempted to discover a strategy for dealing with non-normally distributed data when building models.

1.1 PROBLEM STATEMENT

Battery (service) life prediction presents a mission-critical aspect in the energy storage applications, including electrification of transportation. Such a prediction relies on detailed battery testing, data analysis with regression methods for parameterization, and extrapolation or projection with physics-based or heuristic model simulations. However, batteries are complex chemical systems, and the degradation with aging in the batteries is very complicated to predict. Moreover, Battery performance dataset used in this study are non-normal and multicollinear. Extrapolating such datasets for model predictions needs attention to such characteristics. Finding the right strategy for dealing with these types of non-normal data with multicollinearity and outlier problems is critical.

1.2 LITERATURE REVIEW

In the field of data science and machine learning, regression is a process of obtaining correlation between dependent and independent variables. When the response variable is continuous in nature, one can use regression algorithms for developing a predictive model. Among all the regression algorithms, linear regression is the most common and ancient technique. For multiple linear regression (more than one independent variable), some of the assumptions are: i) residuals should be normally distributed, ii) there should be a linear relationship between independent and dependent variables, iii) independent variables are not significantly correlated (i.e., no multicollinearity) and iv) homoscedasticity of errors [2]. However, there is significant argument about how important a normal distribution of variables is. Schimdt and Finan [3] concluded in their work that linear regression models are robust to violation of normality assumption of variables when there is a large sample size. Williams, et al. [4] argues that it is not mandatory to have variables with normal distributions for building regression models. Rather, the authors suggest focusing on other assumptions like normality and equal variance of error and some potential problems such as multicollinearity among the variables, outliers, etc.

Multicollinearity occurs when there is a strong correlation between predictors in a dataset [5]. The problem of multicollinearity might not affect the accuracy of an Multiple Linear Regression (MLR) model, but multicollinearity makes it difficult to interpret how predictor variables impact the response variable [6]. This problem can be solved by using the variable screening method. Stepwise regression is one of the variable screening methods [7] which can reduce negative multicollinearity impacts. Regarding outliers, one can remove the outlier, but this can lead to a biased model. Data transformation is suggested to treat this problem [8]. When variables are substantially non-normally distributed, data transformation often improves the MLR accuracy [9]. The use of the Box-Cox transformation can reduce both non-normality and outliers in the data [10].

In recent years, nonparametric algorithms have been developed to solve regression problems; these algorithms do not have any normality or multicollinearity assumptions and so transformation of the dataset is not required [11]. One such algorithm, the decision tree, is a tree-like structure consisting of a root node at the top, connecting to layers of intermediate nodes, and ending in a set of terminal nodes (leaves) at the bottom [12]. Each node contains a binary decision and layers of nodes continue until some stopping criterion is achieved. This method is usually capable of achieving high accuracy, but it might suffer from overfitting [13]. Random forests are an ensemble learning method consisting of many (often hundreds of) decision trees. It averages the prediction from each decision tree in the ensemble, leading to a reduction in bias and overfitting, and an increase in accuracy [14] [15].

Recently, various studies have demonstrated using machine learning and deep learning models to predict battery lifetimes. Severson et al. [16] used the linear regression data driven techniques to predict the cycle life before capacity degradation based on the early cycle discharge data. They used data from the discharge voltage curve as inputs to a regression model, and used the predicted number of cycles as the output of the regression model.[17] Their best models achieved a 9.1% average error rate for quantitatively predicting cycle life using the first 100 cycles and a 4.9% average error rate using the first 5 cycles to classify cycle life into two groups. Later, Shan Zhu et al.[18] worked

on the same dataset and used different machine learning algorithms and found the decision tree as the most successful model. They classified battery lifetime into two categories which are high lifetime and low lifetime. Their best model gives 95.2% accuracy to predict whether the battery can maintain above 80% initial capacity after 550 cycles or not. For predicting remaining useful life (RUL) of battery cell, Zhou et al.[19] applied k-nearest neighbor regression using the weighted average useful life of similar nearest cells which share a similar degradation.trend. Liu and Chen [20] used combination of indirect health indicator and Gaussian process regression for predicting RUL.

A number of other studies used supervised machine learning [21–24]and neural networks[25–28] for predicting RUL which is nicely represented at the review of Gao et al.[17] Few of these studies used bagging techniques like random forest. Random forest (combination of lots of decision trees) is a very popular ensemble learning technique which can be used for both classification and regression [13, 29].

1.3 PROPOSED THESIS PLAN

In this study, battery life prediction method is divided into three different sections. First section is focused on data collection and analysis, pattern recognition and target analysis. In second section, a highly skewed, non-normally distributed, multicollinear dataset that delineates a set of battery cycle life performance under high-rate charging is used. To develop a suitable life prediction model, the feasibility of several regression methods were studied. Through analyses, the suitability of tree-based approach is further explained. This part of study began with two multiple linear regression predictive models: one without data transformation and the other with the Box-Cox transformation. Both utilize stepwise regression to address multicollinearity. The importance of normally distributed variables is examined, and the impact on multicollinearity is explored, from the

results of these two models. Two tree-based regression models are then built: a decision tree model and a random forest regression model. For these tree-based models, no data transformation was performed. The suitability of all four models is compared. In final section, random forest and also other regression techniques are utilized with a variety of features and predicted battery cycle life.

Chapter 2. Background

2.1 MACHINE LEARNING ALGORITHM

Five machine learning techniques are used in this work: (1) a traditional multiple linear regression with stepwise technique, (2) Polynomial Regression, (3) Support Vector Regression, (4) tree-based regressions like decision tree, and (5) ensemble learning method such as random forest regression is used.

2.1.1 Multiple Linear Regression

Multiple Linear Regression (MLR) is a popular statistical technique to find out the relationship between predictor variables and response variables. In this method, a linear relationship is modeled between independent variable (predictors) and dependent variable (response). This is like the ordinary least square (OLS) method, but multiple variables can be used in place of a single variable. The formula for MLR is,

$$y_i = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_{ip} + \epsilon_i, \tag{2.1}$$

Where, $\mathbf{y}_i$ is the response variable, x_i are predictors, β_0 is constant coefficient and β_p are independent variables coefficient. There is an error term ϵ_i which add noise to the model. For minimizing this random error, sum of square error (SSE) function is used,

$$\text{SSE} = \sum_{i=1}^{n} (y_i - \mathbf{y}_i)^2 \tag{2.2}$$

where y_i is the observed value and $\mathbf{y}_i$ the predicted value from the model. The best fitted model will give the lowest SSE value. Among independent variables, there should not be high correlation

as in that case the same information will be presented by highly correlated variables to the model. For this reason, one can use stepwise regression for mitigating multicollinearity problems among variables and making the model less complex. Another assumption of MLR is that residuals should be independent and normally distributed with mean of 0 and variance of σ. One can use Durbin statistics for determining the correlation between residuals [30]. However, the size of the error in the prediction also should not change significantly across the values of the independent variable.

2.1.2 Stepwise Regression

Stepwise regression is the iterative creation of a regression model in which the independent variables to be utilized in the final model are chosen step by step. It entails incrementally adding or eliminating potential explanatory factors, with each iteration requiring statistical significance assessment.

The forward selection method starts with nothing and gradually adds new variables, assessing for statistical significance along the way. The backward elimination method starts with a comprehensive model with numerous variables and then removes one to see how important it is in terms of overall results.[31]

2.1.3 Polynomial Regression

In this form of regression analysis, the relationship between the dependent and independent variables are modeled in the nth degree polynomial. A curvilinear relationship is developed between the predictor and response variables. Assumptions of this method is same as multiple linear regression but it expects that there might be a curvilinear relationship between a dependent variable and a set of independent variables, instead of only linear relations. In general, the expected value of y can be modelled as an n^{th} degree polynomial, providing the polynomial regression model.

$$y_i = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots + \beta_n x^n + \epsilon, \tag{2.3}$$

Using a high degree of polynomial tries to overfit the data, while using a lower degree of polynomial tries to underfit the data, therefore we need to find out the best degree value [32]. In this work, a forward selection method is utilized to find the optimum degree value for the model. This strategy gradually increases the degree until it is large enough to define the best model conceivable.

2.1.4 Support Vector Regression

The goal of most linear regression models is to reduce the sum of squared errors. These models are designed to minimize test errors.As long as the error is within a certain range, we may offer our model some leeway in determining the anticipated values. Support Vector Regression allows us to determine how much error is acceptable in our model and will fit the data with an appropriate line (or hyperplane in higher dimensions). Unlike OLS, the goal of SVR is to minimize the coefficients, specifically the $l2$-norm of the coefficient vector, rather than the squared error. Instead, we address the error word in the constraints, where we specify the absolute error to be less than or equal to a defined margin, called the maximum error (epsilon). We can adjust epsilon to achieve the model's desired accuracy. [33] The following is our new objective function and constraints:

$$\begin{aligned} \text{Minimize: } & MIN\frac{1}{2}\|w\|^2 \\ \text{Constraints: } & |y_i - w_i x_i| \leq \varepsilon \end{aligned} \tag{2.4}$$

Because some points may still fall beyond the margins, we must account for the probability of errors greater than ε. With slack variables, we can accomplish this. The idea behind slack variables is simple: any value that is outside of ε can be denoted by ξ as a divergence from the margin by the symbol. We are aware that these deviations are conceivable, but we would prefer to eliminate them as much as feasible. As a result, these deviations might be added to the objective function.

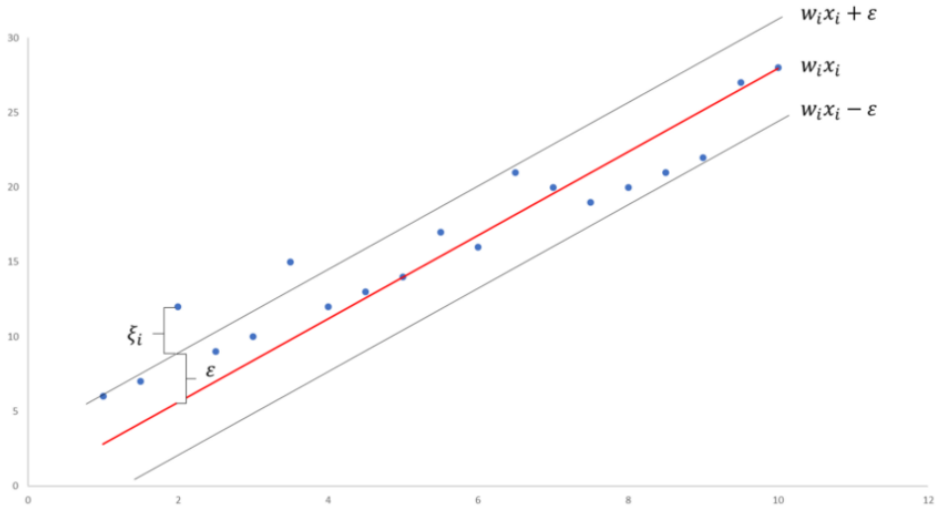


FIGURE 2.1. Illustrative Example of SVR with Slack Variables [34]

$$\begin{aligned} &\text{Minimize: } MIN\frac{1}{2}\|w\|^2 + c\sum_{i=1}^{n}|\xi_i| \\ &\text{Constraints: } |y_i - w_i x_i| \leq \varepsilon + |\xi_i| \end{aligned} \tag{2.5}$$

We may now fine-tune an extra hyperparameter, C. As C rises, so does our tolerance for points outside of ε. As C approaches zero, the tolerance approaches zero, and the equation collapses into the simplified one. [34] We can use kernel trick for converting the data into higher dimensions. When data is non-linear, this kernel trick is very helpful. [35] There are several types of kernel we can use in SVR like linear kernel, polynomial kernel, radial basis kernel etc. In this study, Radial basis kernel function is used.

2.1.5 Decision Tree

The Decision tree algorithm is a non-parametric supervised machine learning method that is used for both classification and regression. It is a tree like structure consists of several branch (see Figure-2.2), where each internal node denotes a test on an attribute, each branch represents

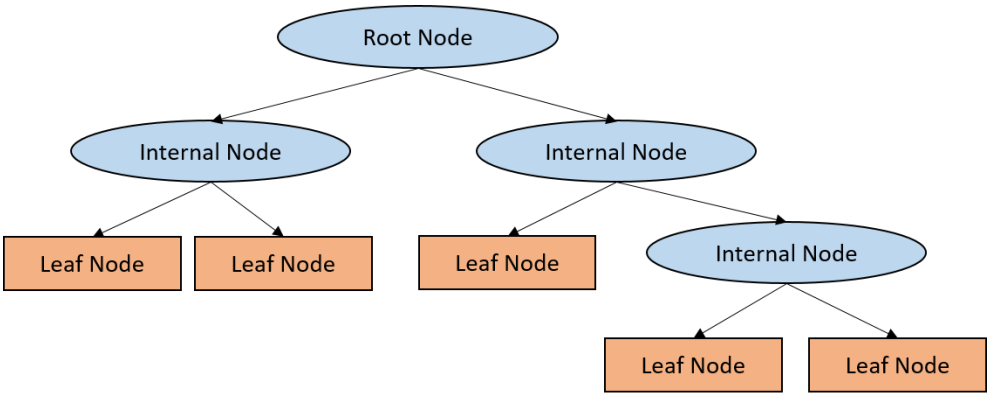


FIGURE 2.2. Decision Tree structure

an outcome of the test, and each leaf node (terminal node) holds a predicted value [36]. Among four types of decision tree algorithm: Iterative Dichotomiser (ID3), Classification and Regression Trees (CART), Chi-Square and Reduction in Variance, the CART algorithm is being widely used for regression problems. In CART regression, data is being split into two groups by finding the threshold that gives the smallest sum of square residual. We need an impurity metric that is suitable for continuous variables, so we define the impurity measure using the weighted mean squared error (MSE) of the children nodes.

$$MSE(t) = \frac{1}{N_t} \sum_{i \in D_t} (y^i - \mathbf{y}_t)^2 \tag{2.6}$$

Here, N_t is the number of training samples at node t, D_t is the training subset at t, y^i is the true target value and $\mathbf{y}_t$ is the predicted target value (sample mean):

$$\mathbf{y}_t = \frac{1}{N_t} \sum_{i \in D_t} y^i \tag{2.7}$$

This procedure is repeated for further splitting until some stopping criterion reached. If any stopping criterion doesn't set up, then the model will fit the training data perfectly, it probably means it is overfit and will not perform well with new data. In that case, the model will have low bias, but potentially high variance. For preventing the model from overfitting, there are some mitigation techniques. One of the techniques is to split observation when there is some minimum number of samples remaining. Another way, one can set the maximum depth of the tree.

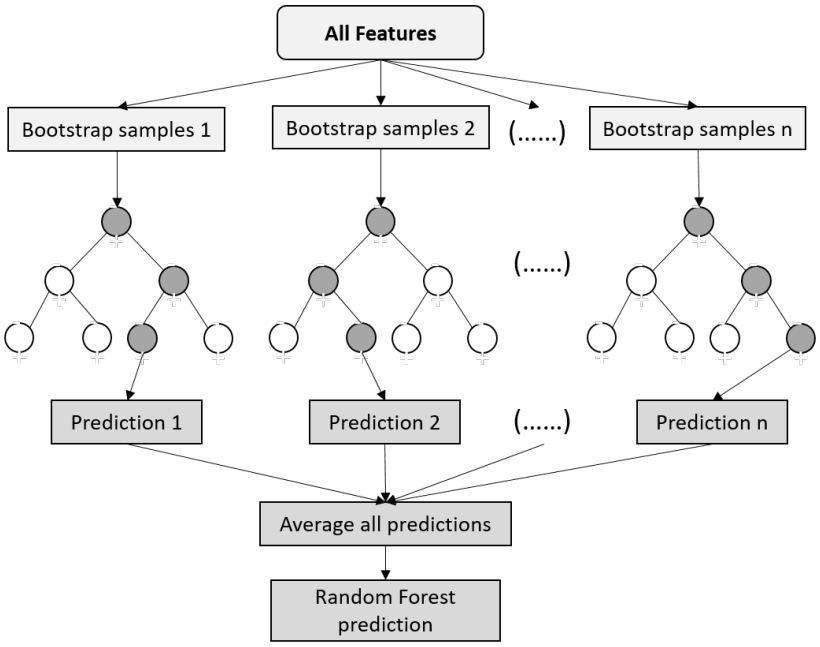


FIGURE 2.3. Random Forest Regression Algorithm

2.1.6 Random Forest Regression (RFR)

As discussed earlier, decision trees might suffer from high variance. To reduce this weakness, random forest regression (RFR) is introduced which constructs many decision trees in one model. This combining technique is called bootstrap aggregation or bagging, as shown in Figure-2.3, to reduce the variance of the model. In this method, the same dataset is not used for all decision trees. A separate bootstrap sample (with replacement) from the original dataset is used for building each tree and then average their result to find out the prediction. To minimize the effect of high collinearity among the trees in a forest, RFR uses a subset of features in each decision tree. For selecting m number of subset features from n total features, one rule is $m = \sqrt{n}$. Due to the splitting of predictors, strong predictors might not be able to dominate all the time which reduces overfitting. RFR model don't need separate cross validation procedures for determining the model performance because it uses out-of-bag (OOB) samples to validate the model built with training samples. [37] [38]

2.1.7 Bias vs Variance Trade-off

The inaccuracy caused by the model's basic assumptions in fitting the data is referred to as bias. A high bias indicates that the model is unable to capture data patterns, resulting in under-fitting. In contrast, Variance is the error caused by the sophisticated model's attempt to fit the data. When a model has a high variance, it passes over the majority of the data points, causing the data to be overfit.

When shown in the Figure 2.4 , as model complexity increases, the bias lowers but the variance increases, and vice versa. A machine learning model should, in theory, have low variance and bias. However, having both is nearly impossible. As a result, a trade-off must be made in order to develop a solid model that performs well on both train and unseen data.

2.2 RESULT PARAMETER

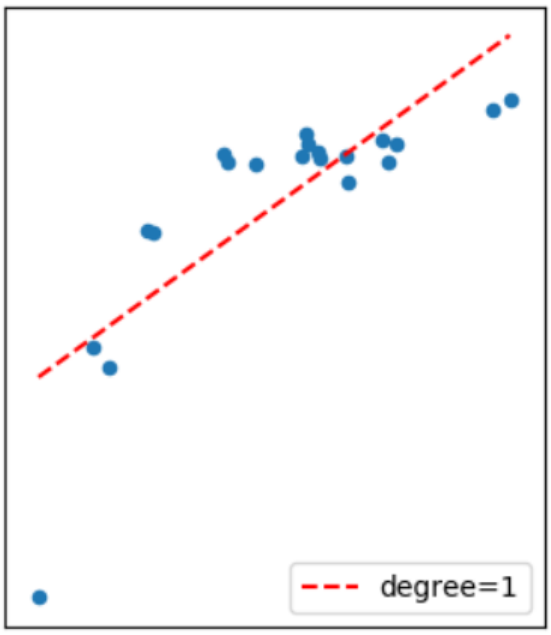
Results are evaluated by the coefficient of determination, root mean squared error, mean absolute percentage error and the correlation coefficient.

2.2.1 Coefficient of Determination

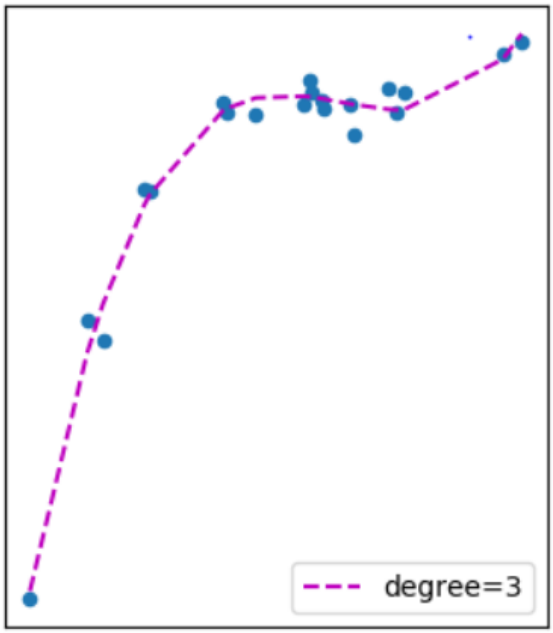
The coefficient of determination (R^2) is the proportion of the variance in response variable that is explained by the model. It is said to be an accuracy of the regression model. The formula:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{2.8}$$

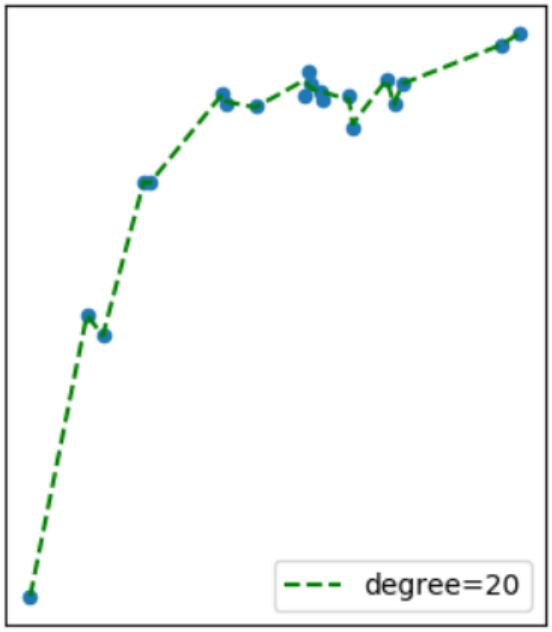
where, $SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y})^2$ is the total sum of squares and $SS_{res} = \sum_{i=1}^{n}(y_i - p_i)^2$ the sum of squares of residuals. Here, p_i is the predicted value from the model.



(a) Underfit (High bias, low variance)



(b) Correct fit (Low bias, low variance)



(c) Overfit (Low bias, high variance)

FIGURE 2.4. Bias- Variance trade-off [39]

2.2.2 Adjusted R^2

R^2 has a tendency to overestimate the linear regression's fit. As the number of effects in the model grows, it always rises. The adjusted R^2 tries to compensate for the overestimation. If a given effect does not improve the model, adjusted R^2 may fall. The adjusted R^2 is calculated as follows:

$$\bar{R^2} = 1 - \frac{SS_{res}/df_{res}}{SS_{tot}/df_{tot}} \tag{2.9}$$

where df_{res} is the degrees of freedom of the estimate of the population variance around the model, and df_{tot} is the degrees of freedom of the estimate of the population variance around the mean. df_{res} is given in terms of the sample size n and the number of variables p in the model, $df_{res} = n - p$. df_{tot} is given in the same way, but with p being unity for the mean, i.e. $df_{tot} = n - 1$.

2.2.3 The root-mean-square error

The root-mean-square error (RMSE) is a measure of the differences between values predicted by a model and the observed values.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(p_i - x_i)^2}{N}} \tag{2.10}$$

where p_i is the predicted value and x_i is the original value for N number of observations.

2.2.4 MAPE

The mean absolute percentage error (MAPE) is a measure of prediction accuracy of a regression method in statistics. It expresses the accuracy by the following formula:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - P_i}{A_i}\right| \tag{2.11}$$

where A_i is the actual value, P_i the predicted value and n the number of observations.

2.2.5 Correlation coefficient

A correlation coefficient is a numerical measurement of the strength of linear relationship between two variables. Given a pair of random variables (X, Y), the formula for correlation coefficient (r) is:

$$r_{(X,Y)} = \frac{conv(X.Y)}{\sigma_X \sigma_Y} \tag{2.12}$$

Where, $conv$ is the covariance and σ_X and σ_Y are the standard deviation for X and Y respectively.

2.2.6 Mallow's C_p

Mallows's C_p, named after Colin Lingwood Mallows, is a statistic that is used to evaluate the fit of a regression model calculated using ordinary least squares. It's used in model selection when a number of predictor variables are available for predicting a given result, and the goal is to find the optimal model using only a subset of them. A low C_p value indicates that the model is fairly accurate. Given a linear model in equation 2.1 and if P regressors are selected from a set of $K > P$, the C_p statistics for that particular set of regressors is defined as:

$$C_p = \frac{SSE_p}{S^2} - N + 2(P + 1) \tag{2.13}$$

Where SSE_p is the error sum of squares for the model with p regressors, S^2 is the residual mean square after regression on the complelte set f K regressors and can be estimated by the mean square error MSE and N is the sample size.

For stepwise regression, the C_p statistic is frequently employed as a stopping criteria. The statistic was proposed by Mallows as a criterion for choosing among many different subset regressions.

C_p has an expectation almost equal to P if the model does not have a significant lack of fit (bias); otherwise, the expectation is about P plus a positive bias factor. For a list of subsets ordered by increasing P, it is advised that one choose a subset with Cp approaching P. In practice, the positive bias can be compensated for by choosing a model from the order list of subset such that $C_p < 2P$. More mathematical approach and usefulness of mallow's C_p is given in [40] and [41].

2.2.7 Variance Inflation Factor

The variance inflation factor (VIF) is statistics for diagnosing multicollinearity in multiple linear regression. Given a regression model in 2.1, VIF is calculated with the following formula:

$$VIF_i = \frac{1}{1 - R_i^2} \tag{2.14}$$

Where, R_i^2 is the coefficient of determination of the regression equation in step one with X_i on the left hand side and other predictor varaibles on the right side. Considering the size of $VIF(\beta_\mathbf{i})$, we can analyze the magnitude of multicollinearity.[42][43] Generally, a rule of thumb is that if $VIF(\beta_i) > 10$ then multicollinearity is considered high .

2.2.8 K-fold Cross Validation

Cross-validation is a resampling technique for evaluating machine learning models on a small sample of data. In k-fold cross validation, the original sample is randomly partitioned into k equal-sized subsamples. A single subsample from the k subsamples is kept as validation data for testing the model, while the remaining $k - 1$ subsamples are used as training data. The cross-validation procedure is then performed k times, with each of the k subsamples serving as validation data exactly once. After that, the k results can be averaged to get a single estimate.

For any data sample, the k value must be carefully chosen. A mis-representative view of the model's skill may emerge from a badly chosen value for k, such as a score with a high variance or a

high bias[44] [45]. More details on estimation and sensitivity analysis of k-fold cross validation is described in [46][47].

2.2.9 Durbin-Watson Statistics

A test for autocorrelation in residuals from a statistical model or regression study is the Durbin Watson (DW) statistic. A number between 0 and 4 will always be assigned to the Durbin-Watson statistic. A score of 2.0 implies that the sample contains no autocorrelation. Positive autocorrelation is defined as a value between 0 and less than 2, whereas negative autocorrelation is defined as a value between 2 and 4 [48]. Kenneth J. White elaborated on his work how durbin-watson statistics is helpful in non-linear model [49].

Chapter 3. Data Analysis

The dataset used in this research was the data made available in the publication paper of Severson et al.[16]. The analyzed dataset consists of data collected by testing 124 commercial lithium iron phosphate batteries (A123, Livonia, Michigan) cycled with a variety of fast charging protocols to reach the end-of-life (EOL) condition defined as 80% of the nominal capacity. Figure- 3.1 shows how all battery cells reaches it's cycle life with respect to discharge capacity.

3.1 BINOMIAL MODEL DEVELOPMENT

Using this dataset, a mechanistic binomial model was developed and used in Lin et al. [50] to provide the mechanistic insight of the dataset regarding the capacity loss attributes and their contributions to the capacity degradation (see Figure-6) .Table-1 outlines the corresponding parameters of this binomial model. From this model, the values of the first exponential term (d) can be obtained for each battery at various points in its cycle life: when discharge capacity dropped to 99%, 98%, 97% and 80% of its initial nominal value. A small portion of the data is shown in Table-2 in descending order from a total of 124 observations. Here, the exponential term d represents the extent of loss of active material (LAM) in the binomial model at different points in its cycle life.

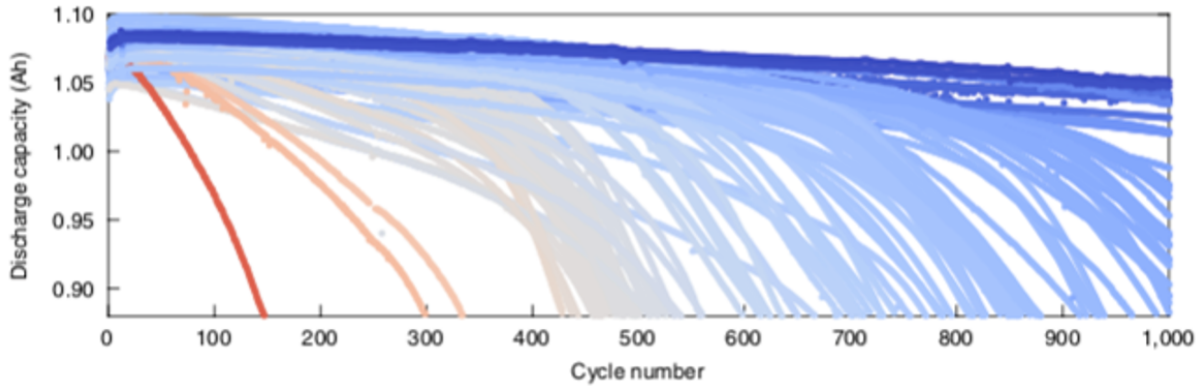


FIGURE 3.1. Discharge Capacity vs cycle number for 124 Li-ion battery cell

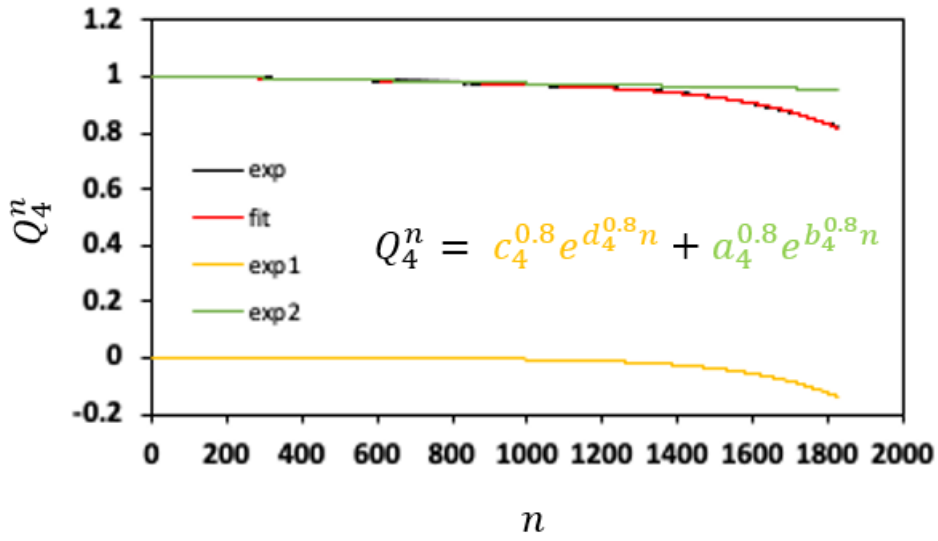


FIGURE 3.2. A binomial model describing the fading of the normalized capacity in cycle aging and life.

TABLE I. Parameter Definition of the Binomial Model

Parameter	Value	Physical Meaning
n		Cycle number
i		Cell number
Q_i^n		Capacity of cell i at cycle n, normalized by the initial capacity
n_i^Q		Cycle number when the normalized capacity of cell i drops below Q
a_i^Q	1.003	Initial Q_i^n, close to 1 at the beginning of life
b_i^Q	-2.67×10^{-5}	Parameter related to the capacity loss due to LLI of cell i, obtained from the initial cycle to cycle n_i^Q
c_i^Q	-8.92×10^{-5}	Initial loss due to LAM, close to 0 at the beginning of life
d_i^Q	4.02×10^{-3}	Parameter related to the capacity loss due to LAM of cell i, obtained from the initial cycle to cycle n_i^Q
R^2	9.99×10^{-1}	Correlation Coefficient

TABLE II. Sample Dataset

$n_i^{0.8}$ (CNEOL)	$\frac{1}{n_i^{0.8}}$	$d_i^{0.99}$	$d_i^{0.98}$	$d_i^{0.97}$	$d_i^{0.80}$
1935	0.0005168	0.009260363	0.006162207	0.004907865	0.003719588
1836	0.00054466	0.008406241	0.005927148	0.004872629	0.003975488
1801	0.00055525	0.009136795	0.006179498	0.005038435	0.004043477
1642	0.00060901	0.009246823	0.006631469	0.005672362	0.004429832

Four different d values are presented when the capacity retention (Q) reaches 99%, 98%, 97% and 80% of its initial nominal value, respectively.

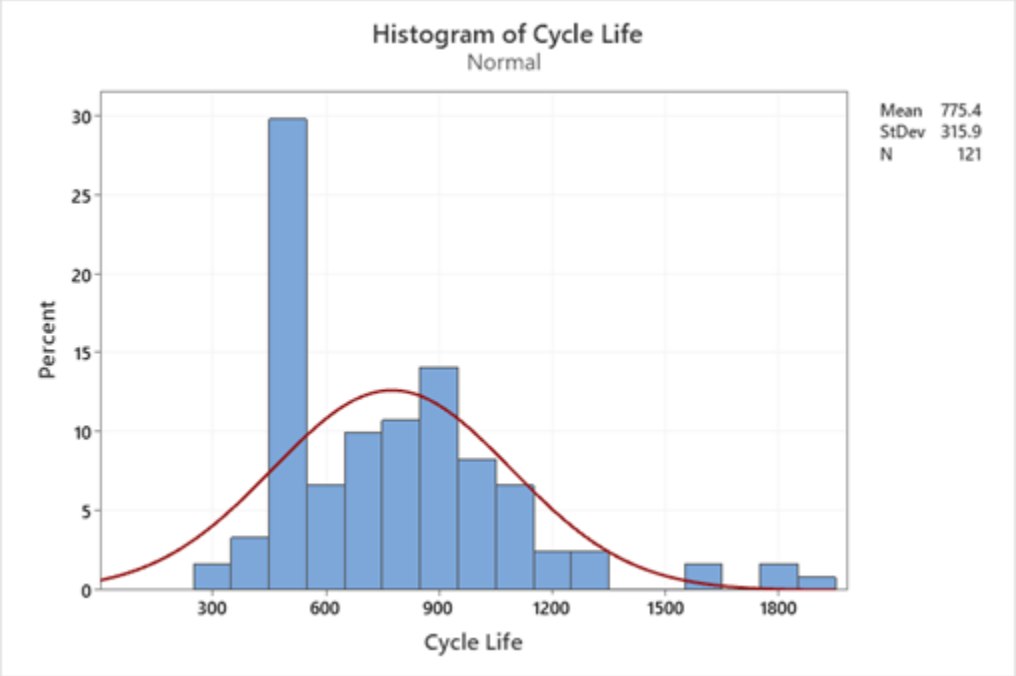


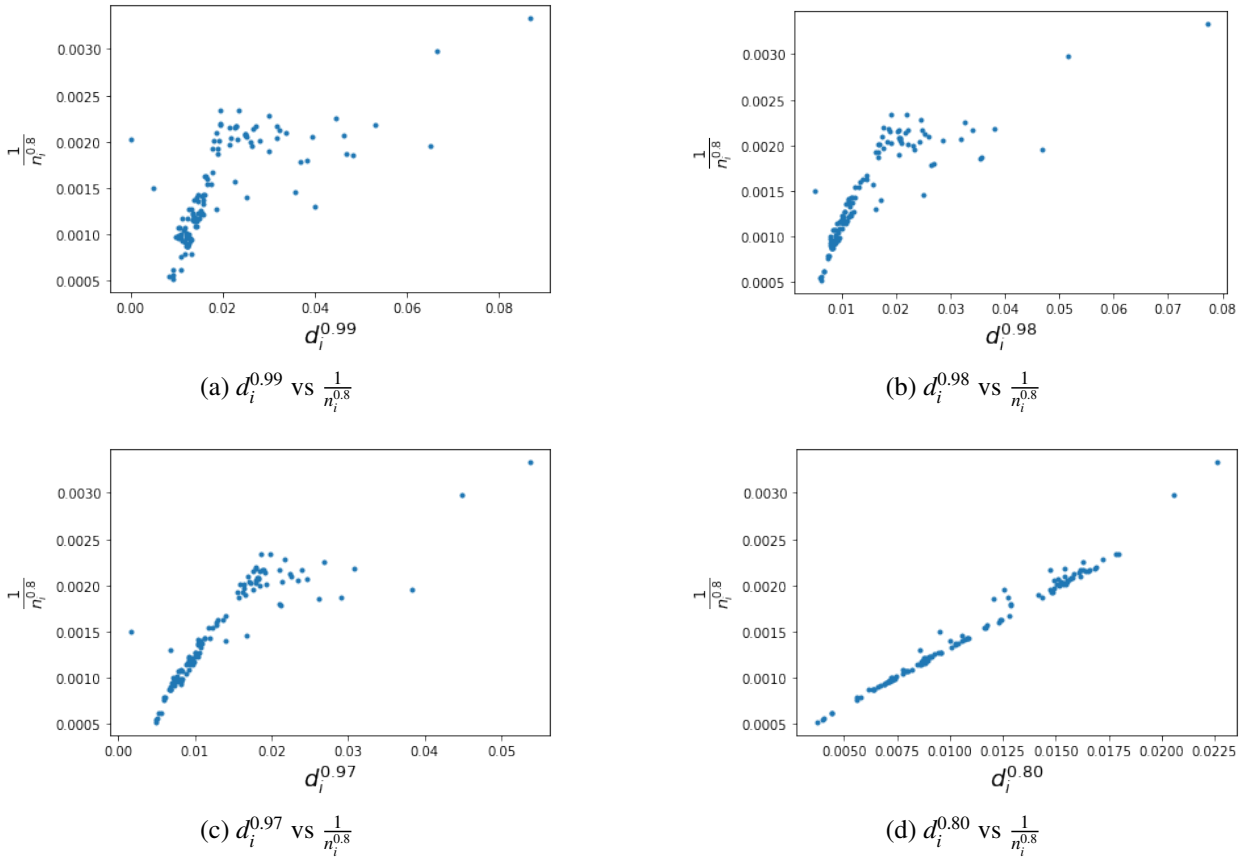
FIGURE 3.3. Relative frequency Histogram for CNEOL ($n_i^{0.8}$)

3.2 TARGET ANALYSIS

It is observed from the relative frequency histogram of the cycle life in Figure 3.3 that only a few batteries have high cycle life in our dataset. Most of the batteries have a cycle life between 500 and 1000 cycles.

Correlation between the value of 1/(cycle number at EOL or n_i^Q) and d value at different stages of cycle life is given in Figure-3.4. There is a strong correlation between $d_i^{0.8}$ and $\frac{1}{n_i^{0.8}}$. However, our goal is to predict the cycle life from the early cycle d value, i.e., $d_i^{0.99}$, $d_i^{0.98}$ & $d_i^{0.97}$. As there is a good linear relationship between $d_i^{0.8}$ and $n_i^{0.8}$, predicting the cycle life from the early cycle d values shall depend on how well the conformity of these early cycle d values to such a linear relationship is, so the accuracy of the life prediction can be further assessed.

In this work, multiple linear regression and tree-based regression algorithms were utilized for this assessment. Comparison of the results from different regression methods shall help us understand any limitation of linear regression over random forest regression and importance of data transformation while data is not normally distributed. For the highly skewed non-normal dataset



(a) $d_i^{0.99}$ vs $\frac{1}{n_i^{0.8}}$

(b) $d_i^{0.98}$ vs $\frac{1}{n_i^{0.8}}$

(c) $d_i^{0.97}$ vs $\frac{1}{n_i^{0.8}}$

(d) $d_i^{0.80}$ vs $\frac{1}{n_i^{0.8}}$

FIGURE 3.4. Correlation between $\frac{1}{n_i^{0.8}}$ and each variable

used in this study, tree-based models outperform linear regression models in terms of accuracy and robustness. Normal distribution of variables is turned out to be an important factor for getting a good multiple linear regression model, whereas it is not necessary for the tree-based models. Using Box-Cox transformation for the dataset increases the prediction accuracy.

Chapter 4. Performance Evaluation of Tree based regression over multiple linear regression for non-normally distributed dataset

4.1 METHODOLOGY

The scope of this part of study is to compare and understand the performance of MLR-based and tree-based models on this non-normal dataset. The entire process of work is shown in Figure-4.1. First, data visualization is done to find whether our features are normally distributed or not. Then for one of the MLR models, data is transformed using Box-Cox transformation. For MLR models, k-fold cross validation is used to determine the model validation. One MLR model is developed using transformed data and another with the original data. Effect of non-normality in the dataset is scrutinized by observing the performance of these two models. In the RFR and decision tree model, 80 percent of the data is used for training and the rest of the data is used for testing. Finally, the accuracy of the different algorithms is compared using some resulting parameter.

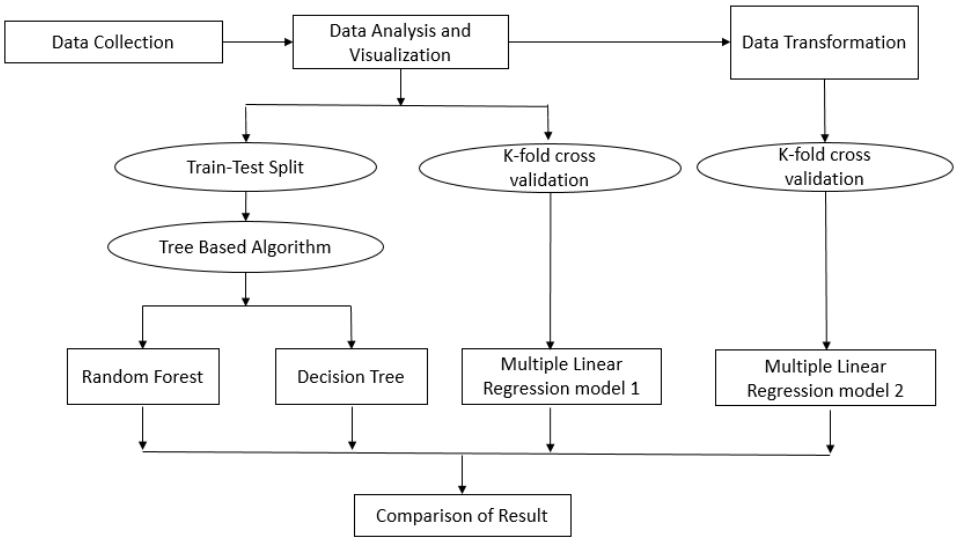
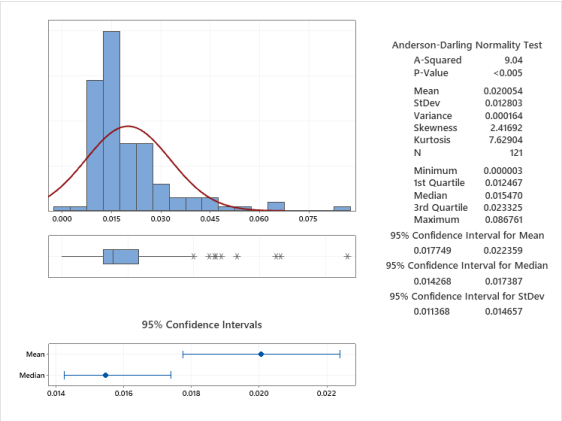
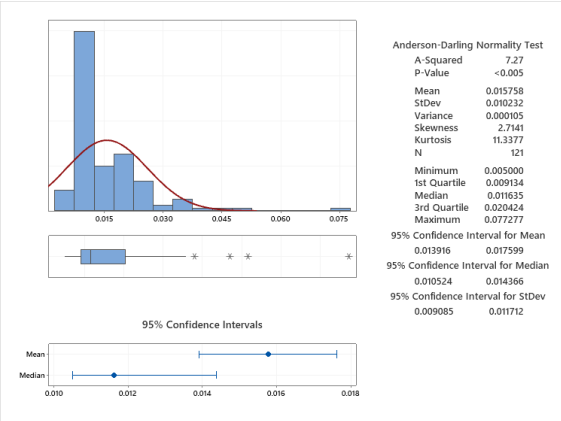


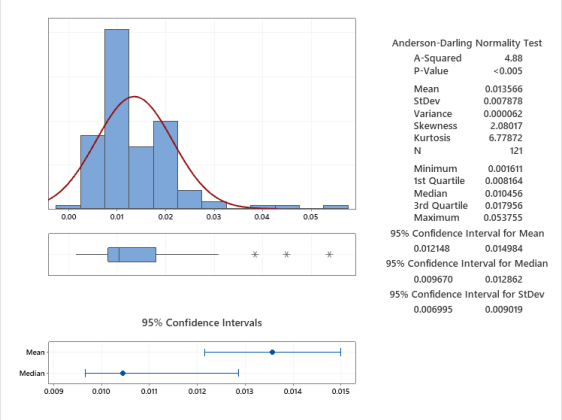
FIGURE 4.1. Flowchart of the performance evaluation method



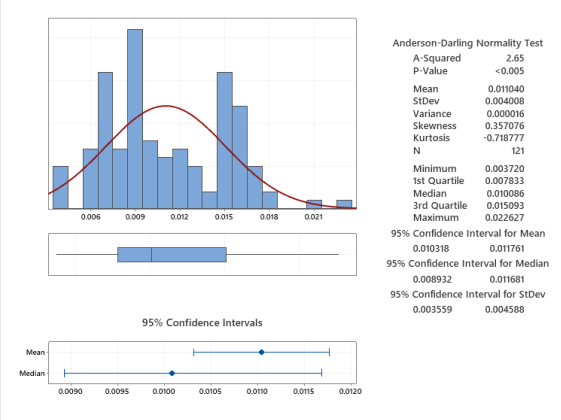
(a) Statistics for $d_i^{0.99}$



(b) Statistics for $d_i^{0.98}$



(c) Statistics for $d_i^{0.97}$



(d) Statistics for $d_i^{0.80}$

FIGURE 4.2. Histogram and descriptive statistics of features

4.1.1 Data Visualization

When skewness of a variable falls between -0.80 to 0.80, it may be considered as normally distributed. Histograms of the independent and dependent features are presented in Figure-4.2. It is clearly observed from the histogram that none of the features is normally distributed. All the independent features are right skewed. There are some outliers observed in the data histogram. Correlation between each independent feature is presented in TABLE III. High correlation (0.98) between $d_i^{0.98}$ and $d_i^{0.97}$ is detected, meaning the data suffers from multicollinearity.

TABLE I. Correlation Matrix Among Independent Features

	$d_i^{0.99}$	$d_i^{0.98}$	$d_i^{0.97}$
$d_i^{0.99}$	1	0.98	0.90
$d_i^{0.98}$	0.93	1	0.98
$d_i^{0.97}$	0.90	0.98	1

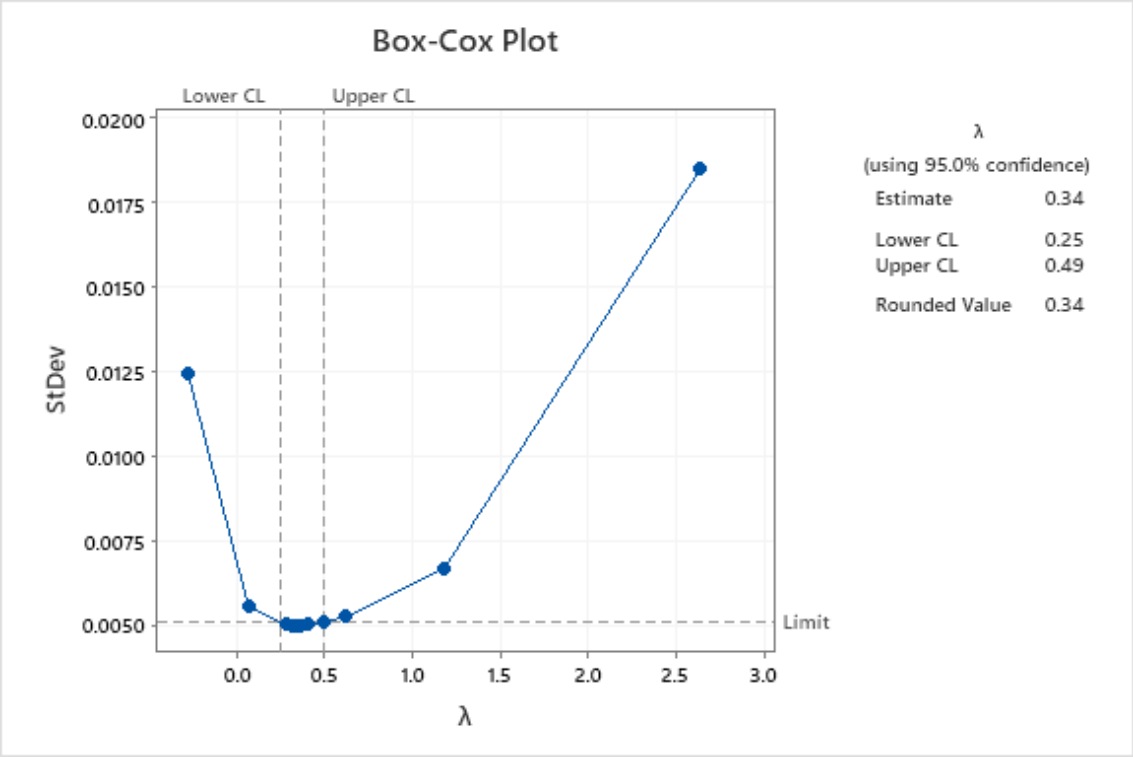
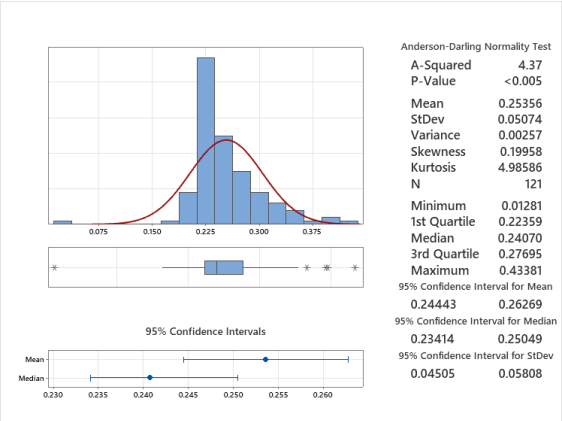
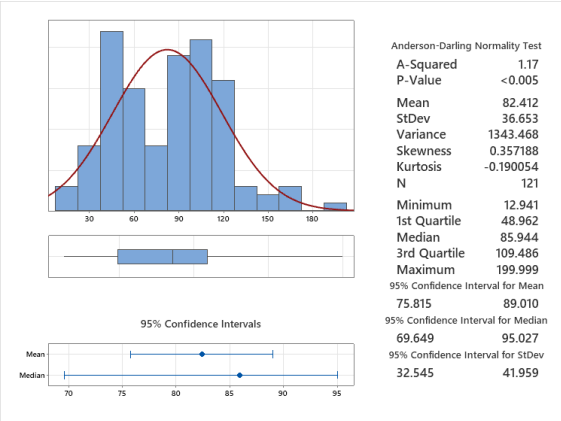


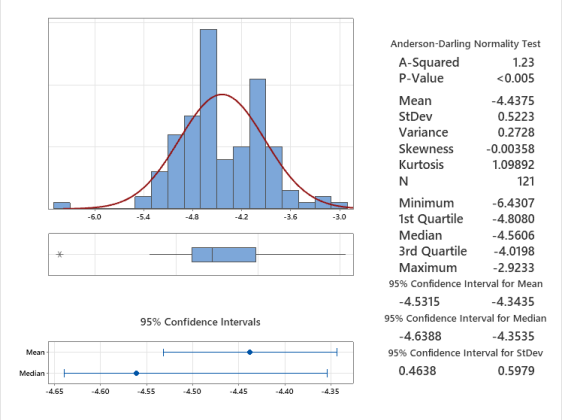
FIGURE 4.3. Box-cox transformation of $d_i^{0.99}$



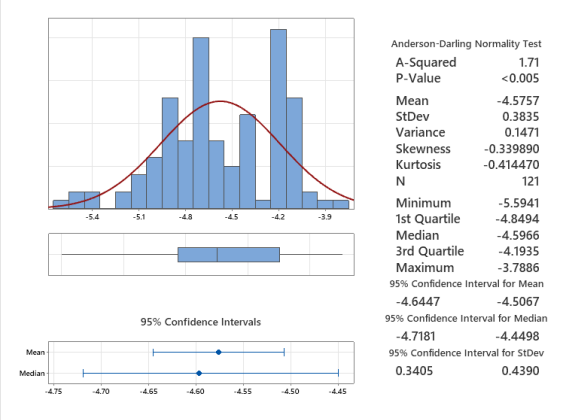
(a) Statistics for $d_i^{0.99}$ after transformation



(b) Statistics for $d_i^{0.98}$ after transformation



(c) Statistics for $d_i^{0.97}$ after transformation



(d) Statistics for $d_i^{0.80}$ after transformation

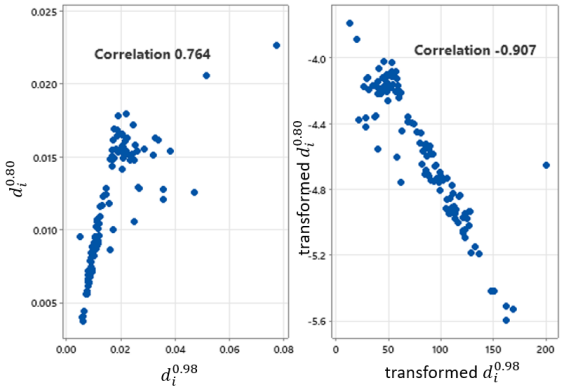
FIGURE 4.4. Histogram and descriptive statistics of features after transformation

4.1.2 Data Transformation

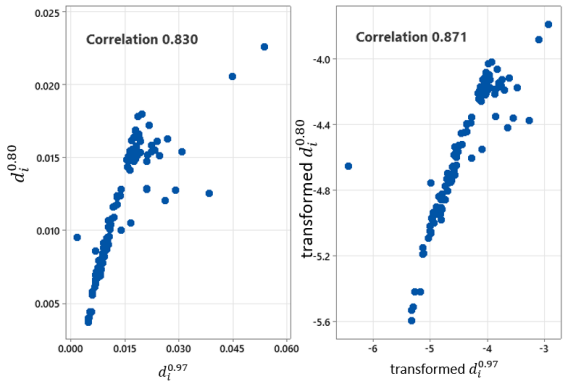
When data is not normally distributed, data transformation can be used to reduce skewness and make it more normal. There are various types of transformation techniques: log transformation, inverse transformation, Box-Cox transformation, etc. In this work, Box-Cox transformation is utilized for transforming the features. The mathematical formula for Box-Cox transformation is given below:

$$y(\lambda) = \begin{cases} \frac{y^{\lambda-1}}{\lambda}, & \text{if } \lambda \neq 0. \\ \log y, & \text{if } \lambda = 0. \end{cases} \tag{4.1}$$

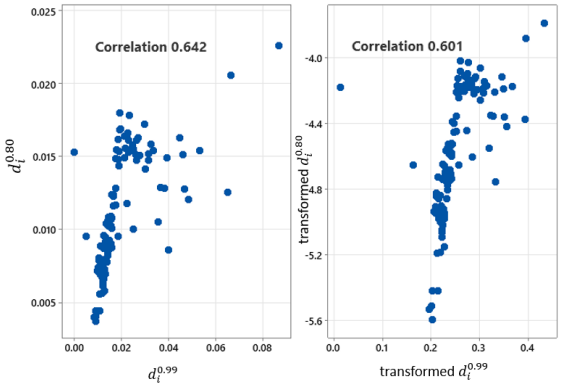
The lambda values usually vary from -5 to 5 and the optimal lambda value is one which



(a) Correlation increased for feature $d_i^{0.98}$ after transformation



(b) Correlation increased for feature $d_i^{0.97}$ after transformation



(c) Correlation decreased for feature $d_i^{0.99}$ after transformation

FIGURE 4.5. Correlation between independent and dependent features before and after transformation

represents the best skewness of the distribution. When the value of lambda is zero, then log transformation will be used. Figure-4.3 shows how optimum lambda value is calculated for $d_i^{0.99}$. Figure-4.4 represents the histogram and descriptive statistics of transformed features which show that the value of skewness is reduced and the data is more normal. Ideally, skewness should be closer to 0. The skewness of independent variable $d_i^{0.99}$ is reduced from 2.41 to 0.19, $d_i^{0.98}$ is reduced from 2.71 to 0.35, $d_i^{0.97}$ is reduced from 2.08 to -0.003 and dependent feature $d_i^{0.80}$ is reduced from 0.35 to -0.33. All the transformed features have skewness between -0.80 to 0.80 which confirms that the transformed features are more normal than the original data. The p value of transformed feature is less than 0.05; so we can tell that data is not fully normally distributed even after transformation. Though it is not fully normally distributed, reduced skewness will definitely help in building model.

The linear relationship between the predictor and response variable is very important for linear regression. The higher the correlation between predictor and response variable, the higher the model accuracy. Here, correlation with the response variable is increased for the $d_i^{0.97}$ and $d_i^{0.98}$ due to the transformation of the data represented in Figure-4.5a & 5.5b. But for the $d_i^{0.99}$, correlation decreased slightly (see Figure-5.5c). The most significant change is observed for transformed feature $d_i^{0.98}$: it shows an almost perfect linear relationship with transformed $d_i^{0.80}$, with the exception of an outlier.

4.1.3 Model Building

Minitab software is utilized for building the MLR models. Two MLR models are developed for observing the effect of non-normality in the model: one model uses an original dataset with non-normal variables, the second model uses a transformed dataset with more normal variables. Stepwise regression is utilized for each of the MLR models, where the choice of predictive variables is carried out by an automatic procedure which objectively determines which independent variables are the most important predictors for the model. This method selects variables with the help of t-test value. This procedure is continued until no further independent variables can be found that yield significant t-values (at the specified α level) in the presence of the variables already in the model. This is also called variable screening procedure. In our models, we set the value of α at 0.15. K-fold cross validation is utilized with 10 folds. Lastly, residual analysis is done to observe whether the residuals are normally distributed and have equal variance or not.

The sklearn python environment is used for developing the tree-based models. The decision tree and RFR models are non-parametric, meaning they are capable of handling non-linear effects. For that reason, the original dataset (without transformation) is used for the tree-based models. As we discussed before, cross validation is not required for the RFR model. This RFR model is built with 80% of the data being used in training and the remaining 20% set aside for testing. The RFR is built with 100 decision trees where each tree uses a bootstrap sample from the original dataset. Then,

TABLE II. Regression Summary for MLR 1

Stepwise Selection of Terms

Candidate terms: $d_i^{0.99}$, $d_i^{0.98}$, $d_i^{0.97}$

	—Step 1—		—Step 2—		—Step 3—	
	Coef	**P**	**Coef**	**P**	**Coef**	**P**
Constant	0.0053		0.0044		0.0047	
$d_i^{0.97}$	0.4222	0.00	1.125	0.00	1.067	0.00
$d_i^{0.98}$			-0.551	0.00	-0.367	0.001
$d_i^{0.99}$					-0.120	0.002
S	0.00224		0.00197		0.00190	
R-sq	68.87%		76.06%		77.87%	
R-sq(adj)	68.61%		75.65%		77.30%	
Mallows C_p	47.60		11.60		4.00	
AICc	-1128.31		-1157.93		-1165.30	
BIC	-1120.13		-1147.09		-1151.84	

α to enter = 0.15, α to remove = 0.15

Model Summary

S	**R-sq**	**R-sq (adj)**	**PRESS**	**R-sq (pred)**	**10-fold S**	**10-fold R-sq**
0.0019	77.87%	77.30%	0.0005386	72.06%	0.0021673	70.52%

Coefficients

Term	**Coef**	**SE Coef**	**95% CI**	**T-Value**	**P-Value**	**VIF**
Constant	0.004756	0.000385	(0.00399, 0.00551)	12.37	0.000	
$d_i^{0.99}$	-0.1202	0.0388	(-0.1970, -0.0434)	-3.10	0.002	8.11
$d_i^{0.98}$	-0.367	0.107	(-0.579, -0.154)	-3.41	0.001	39.73
$d_i^{0.97}$	1.067	0.118	(0.834, 1.300)	9.07	0.000	28.24

feature importance is obtained to illuminate which feature(s) play a significant role in prediction.

4.2 RESULTS

4.2.1 Result of MLR model 1 without transformation

Stepwise regression with a forward selection method is adopted to build this model. Table-II shows how an independent variable is selected in each step. At the end of three steps, adjusted R^2 value becomes 77.30%. However, from the model summary in Table-II, 10-fold cross validation R^2 score is observed as 70.52% which concludes that the model is overfit. Multicollinearity is determined by variance inflation factor (VIF) and values greater than 10 suffer from severe multicollinearity. From the coefficients in Table-II, we see that two of the variables have VIF greater than 10 and hence there is a serious multicollinearity problem. Although p-values are all less than 0.05, and model accuracy is decent, this model suffers from overfitting and multicollinearity.

TABLE III. Regression Summary for MLR 2

Stepwise Selection of Terms

Candidate terms: $d_i^{0.99}$, transformed $d_i^{0.98}$, transformed $d_i^{0.97}$

	—Step 1—		—Step 2—	
	Coef	**P**	**Coef**	**P**
Constant	-3.7935		-3.5999	
transformed $d_i^{0.98}$	-0.00949	0.000	-0.0107	0.000
$d_i^{0.99}$			-4.68	0.006
S	0.162076		0.157588	
R-sq	82.29%		83.39%	
R-sq(adj)	82.14%		83.11%	
Mallows C_p	8.30		2.46	
AICc	-92.79		-98.47	
BIC	-84.61		-87.63	

α to enter = 0.15, α to remove = 0.15

Model Summary

S	**R-sq**	**R-sq (adj)**	**PRESS**	**R-sq (pred)**	**10-fold S**	**10-fold R-sq**
0.15788	83.39%	83.11%	3.54007	79.94%	0.165785	81.15%

Coefficients

Term	**Coef**	**SE Coef**	**95% CI**	**T-Value**	**P-Value**	**VIF**
Constant	-3.59	0.0776	(-3.75,-3.44)	-46.4	0.000	
$d_i^{0.99}$	-4.68	1.67	(-7.99,-1.38)	-2.8	0.006	2.21
transformed $d_i^{0.98}$	-0.01	0.00058	(-0.011,-0.009)	-18.3	0.000	2.21

Analysis of Variance

Source	**DF**	**Seq SS**	**Contribution**	**Adj SS**	**Adj MS**	**F-Value**	**P-Value**
Regression	2	14.717	83.39%	14.7167	7.35837	296.30	0.000
$d_i^{0.99}$	1	6.354	36.01%	0.1955	0.19552	7.87	0.006
transformed $d_i^{0.98}$	1	8.363	47.39%	8.3627	8.36266	336.74	0.000
Error	118	2.930	16.61%	2.9304	0.02843		
Total	120	17.647	100.00%				

4.2.2 Result of MLR model 2 with transformation

After trying numerous models with different features, the best model from MLR is found by using transformed features (except for $d_i^{0.99}$) and stepwise regression. The results presented in Table-III show how stepwise regression found that the transformed $d_i^{0.98}$ is the most important independent variable able to explain 82.29% variation of the model by itself, followed by $d_i^{0.99}$ which when added increases the R^2 score to 83.39%. Stepwise regression eliminates one feature $d_i^{0.97}$ and makes the model less complicated. It reaches Mallow's C_p of 2.46 in step 2 which is almost equal to the number of predictors and good enough to stop the regression in that step. The regression equation for this model is,

$$\texttt{transformed\_}d_i^{0.80} = -3.5999 - 4.68[d_i^{0.99}] - 0.010701[\texttt{transformed\_}d_i^{0.98}] \tag{4.2}$$

Cross fold validation with 10 folds is used in MLR model 2. From the model summary in Table-III, it is observed that the 10-fold cross validation score is 81.15% and the adjusted R2 score is 83.11%, demonstrating little overfitting. Table-III also represents an ANOVA table where the transformed

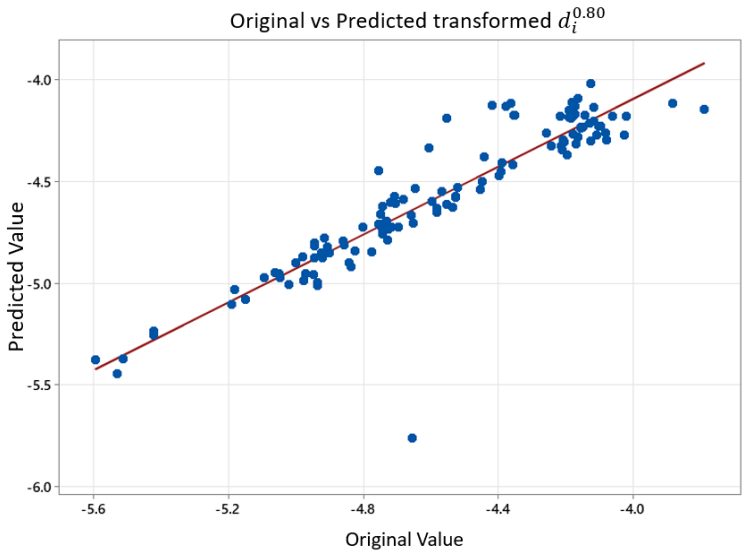
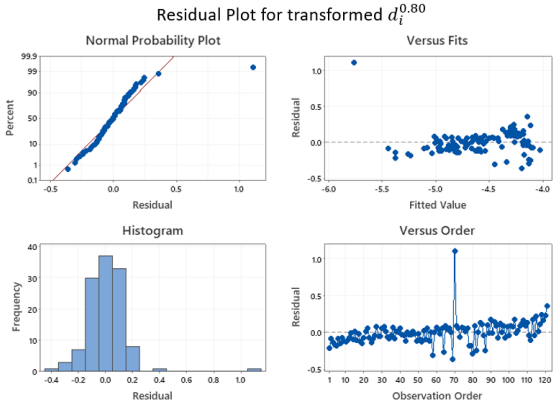


FIGURE 4.6. Original vs Predicted Observation in MLR model 2

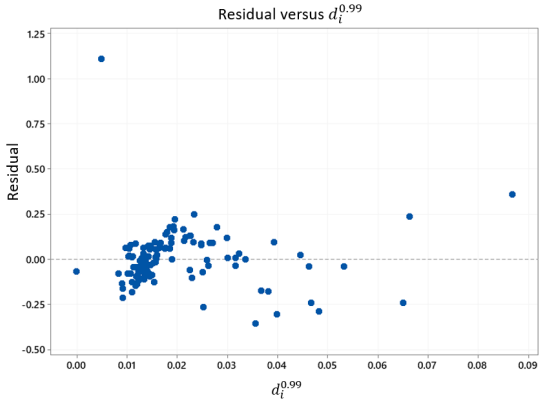
$d_i^{0.98}$ contributes 47.39% to the model and $d_i^{0.99}$ contributes 36.01% to the model. None of the p-values are greater than 0.05. From coefficients, notice that none of the VIF values are greater than 10, meaning there is no multicollinearity. Figure-4.6 shows the relationship between the original transformed $d_i^{0.80}$ and the values predicted by the MLR model. There is a correlation of 0.91 between original and predicted value. This model is well fitted but there is an outlier observable. The variance not explained by the model may be due to this outlier. One can remove that outlier but doing so makes the model less viable.

4.2.3 Residual Analysis for MLR model 2

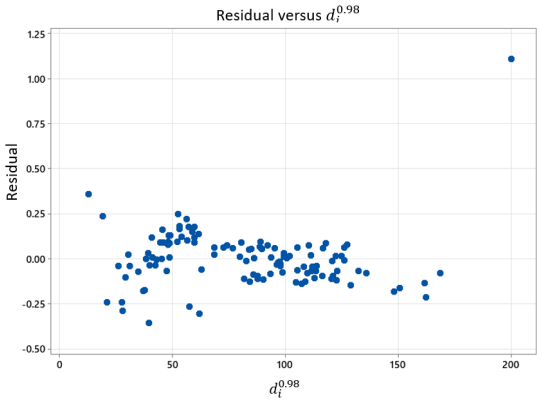
For the MLR model, some of the assumptions are that residuals should be normally distributed and there should be homogeneity of variance. The residuals histogram and normal probability plot for model 2 can be found from Figure-4.7 which shows that the residuals are almost normal with exception of one outlier. Residual relation with fitted value and observation order also can be obtained from the same figure indicating that the residual has equal variance except one outlier. Figure-4.7b and Figure-4.7c represents the predictor vs residual plot for two selected predictors. There is also one unusual observation seen in that plot. The Durbin-Watson statistics value is 1.86;



(a) Residual plot for MLR model 2



(b) Residuals vs $d_i^{0.99}$ plot for MLR model 2



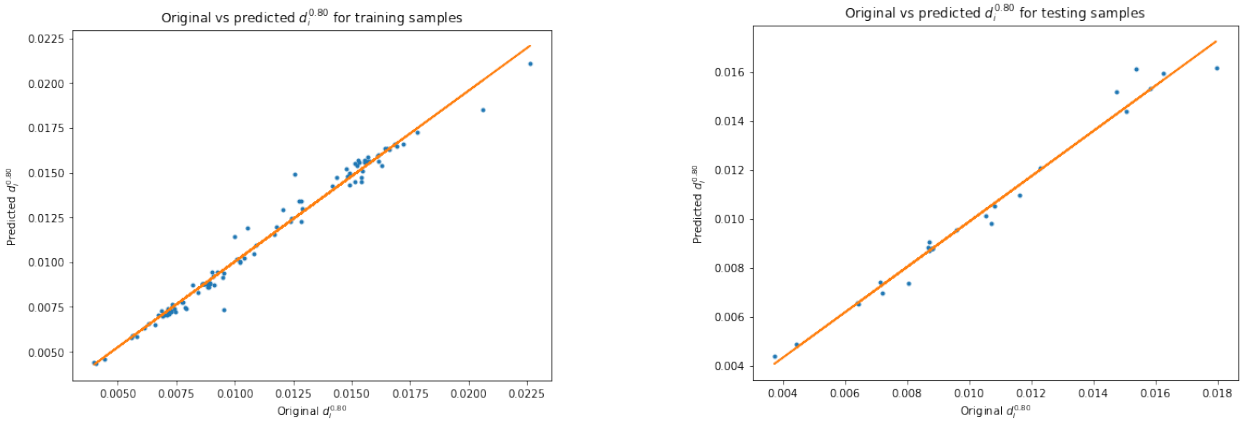
(c) Residuals vs transformed $d_i^{0.98}$ plot for MLR model 2

FIGURE 4.7. Residual Analysis for MLR 2

this value is close to 2 which demonstrates that there is no residual correlation. Further investigation should be done for finding the unusual observation and find out if there is any potential option to remove it.

4.2.4 Result of Decision Tree Model

The decision tree model obtained training and testing accuracies of 99.73% and 97.54% accordingly. Correlations between original and predicted values are 0.99. This model is overfit to training data.



(a) For training samples

(b) For testing samples

FIGURE 4.8. Original vs Predicted $d_i^{0.8}$ for training and testing data in RFR model

TABLE IV. Performance of RFR Model

RFR	Correlation	R^2 value	MAPE	RMSE
Traning Sample	0.99	0.98	3.21	0.0005
Testing Sample	0.99	0.97	4.22	0.0005

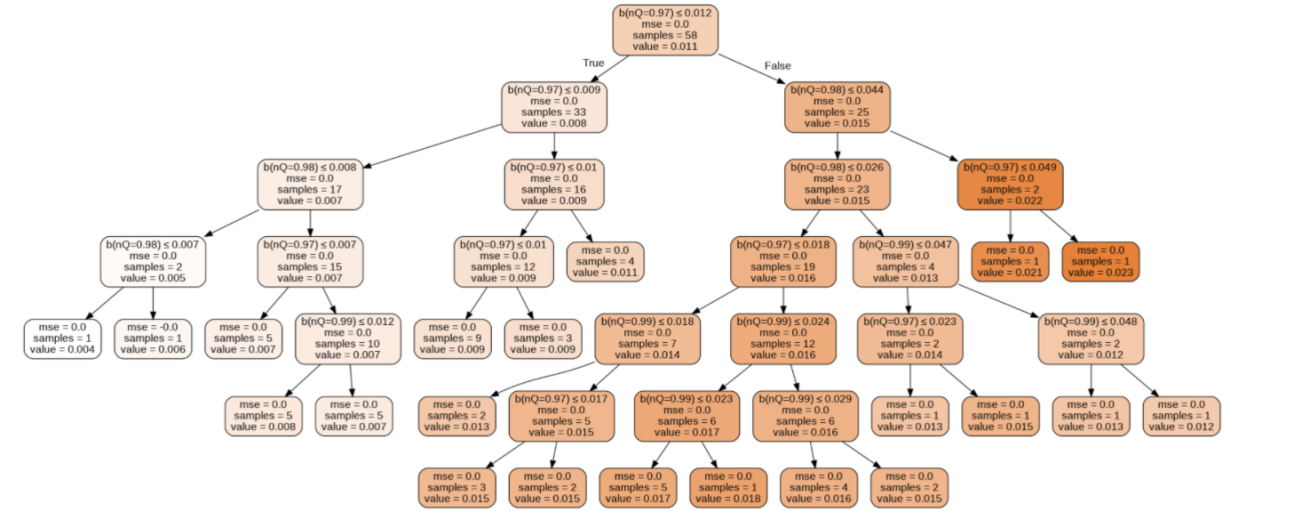


FIGURE 4.9. A Sample Decision Tree from the RFR model

4.2.5 Result of RFR Model

Table-IV represents the results of the RFR model. This model is giving 98.02% training accuracy and 97.73% testing accuracy which validates the model with no overfitting. Error is calculated through mean average percentage error (MAPE) and root mean square error (RMSE). For this

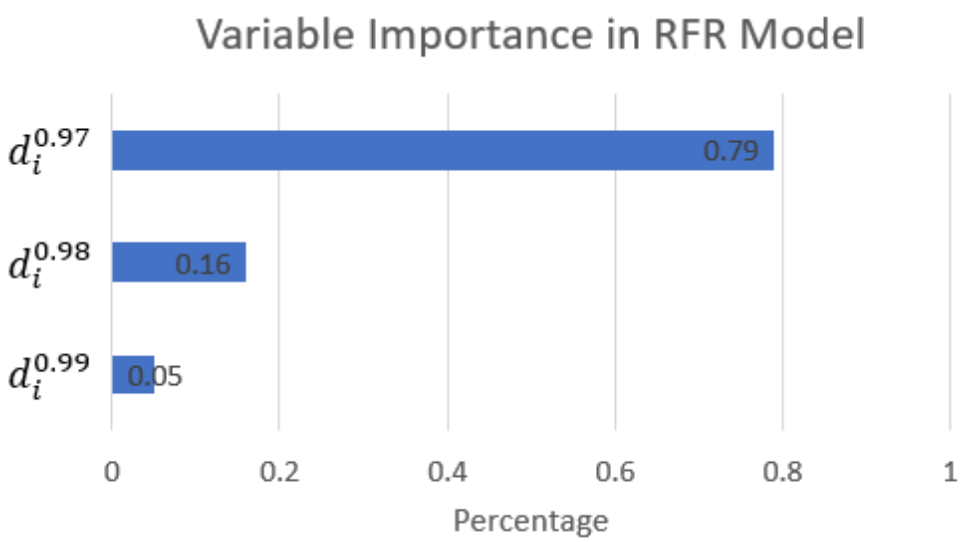


FIGURE 4.10. Variable importance in RFR model

TABLE V. Performance Comparison of Four Model

Model	Training Accuracy	K-fold/ Testing Accuracy	Correlation Coeffi-cient	Model Overfit	Multi-collinearity problem	Outlier Problem
MLR 1	77.30%	70.52%	0.88	Yes	Yes	No
MLR 2	83.11%	81.15%	0.91	Yes	No	Yes
Decision Tree	99.73%	97.54%	0.99	Yes	No	No
RFR	98.02%	97.73%	0.99	No	No	No

analysis, MAPE will be effective as our feature values are so small. In this method, the MAPE value for training samples is 3.21% and 4.22% for testing samples. Original vs predicted $d_i^{0.80}$ plots for training and testing samples are shown in Figure-4.8. The RFR model fits both training and testing samples well with 0.99 correlation value. A random tree from the forest is shown in Figure- 4.9 One can find the variable importance from this model which is shown in Figure-4.10. For this model, $d_i^{0.97}$ has the highest importance with 79%, followed by $d_i^{0.98}$ with 16% importance, and lastly $d_i^{0.99}$ with 5% importance. This RFR model utilized all three of the variables without any transformation.

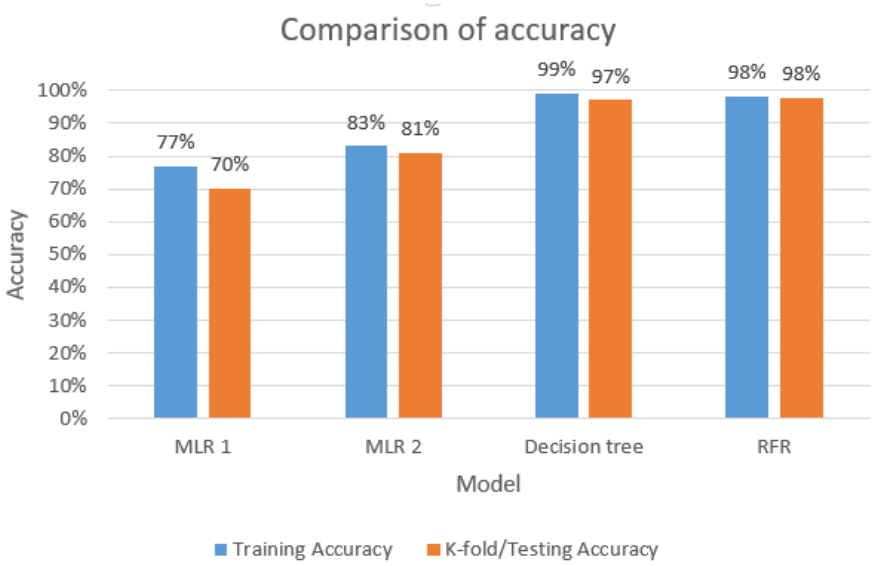


FIGURE 4.11. Accuracy comparison among four models

4.3 COMPARISON OF RESULTS AND DISCUSSION

An overall comparison of multiple linear regression models versus tree-based regression models is presented in Table-V and Figure-4.11. MLR model 1 is built without data transformation, suffers from overfitting and multicollinearity and achieves a comparatively poor accuracy. On the other hand, MLR model 2 utilizes data transformation and stepwise regression, which leads to a model with reduced overfitting and no multicollinearity. While MLR 2 has improved accuracy compared to MLR 1, it is still significantly worse than tree-based models. Both MLR models contain predictive outliers. The decision tree model obtains very good accuracy (97.5%) but is overfit. The RFR model is an ensemble method and therefore retains the accuracy of the decision tree model (and slightly improves upon it) while significantly reducing overfitting. Furthermore, there are no outliers in either tree-based model.

The tree-based models are not negatively impacted by the multicollinearity and non-normality inherent in this dataset, making them an ideal choice for datasets with skewed distributions and multicollinear features. Tree-based models are non-parametric and able to cope with the highly non-linear effect of the features. Residual analysis is done for the MLR model for justifying the

residual assumption of MLR which is not needed for the RFR model. The correlation coefficient between original and predicted value is higher in tree-based models. For the best tree-based model (RFR), $d_i^{0.97}$ turns out to be the most important predictor where the best MLR model (model 2) finds transformed $d_i^{0.98}$ as the most valuable predictor.

There is a very good linear relationship between $d_i^{0.80}$ and $n_i^{0.8}$(cycle number at EOL). As it is now possible to predict $d_i^{0.80}$ from the early cycle b values with the help of tree-based regression very precisely, then it will be feasible to get the battery cycle life from that predicted $d_i^{0.80}$. In this study, all the early cycle d values is utilized to predict $d_i^{0.80}$. In future study, combination of independent features (early cycle d values) will be used to predict battery cycle life.

However, this dataset is limited to only 123 battery cell and low in high cycle life battery cell. If it is possible to accumulate more battery cell in the dataset and reduce the skewness of the features, the predictive model will be more robust.

Chapter 5. Cycle Life Prediction of Li-ion Battery Using Supervised Machine Learning Techniques

5.1 METHODOLOGY

It is showed in data analysis section that $d_i^{0.80}$ has a good linear relationship with cycle number at the end of the life (CNEOL). For predicting CNEOL, I connected two regression models to predict the cycle life. The first model predicts $d_i^{0.80}$ utilizing features $d_i^{0.97}$,$d_i^{0.98}$ and $d_i^{0.99}$ and the second model predicts cycle life with feature $d_i^{0.80}$ (the output of our first model). This process is shown in Figure -5.1. For $d_i^{0.80}$ prediction, three separate analyses were performed using three different sets of features from $d_i^{0.97}$, $d_i^{0.98}$ and $d_i^{0.99}$ to observe how early we can predict cycle life.. The first analysis utilized all three early cycle d values. The second analysis utilized only the $d_i^{0.98}$ and $d_i^{0.99}$ values. The last analysis only utilized the single feature $d_i^{0.99}$. For each analysis, 80% of the data is used in the training set and the rest of the data used in the testing set. Five machine learning regression techniques were utilized to predict $d_i^{0.80}$ of the battery for each analysis: random forests, decision trees, support vector regression, multiple linear regression and polynomial regression. Accuracy was compared using the R^2 score. Our initial predictions suffered from over-fitting. Parameter tuning and model pruning was deployed to ameliorate this. A linear regression model is built with $d_i^{0.80}$ and 1/CNEOL. After predicting the $d_i^{0.80}$ value, one can predict cycle life utilizing that model.

5.2 RESULTS

Results are evaluated by the coefficient of determination, root mean squared error, mean absolute percentage error, and the correlation coefficient.

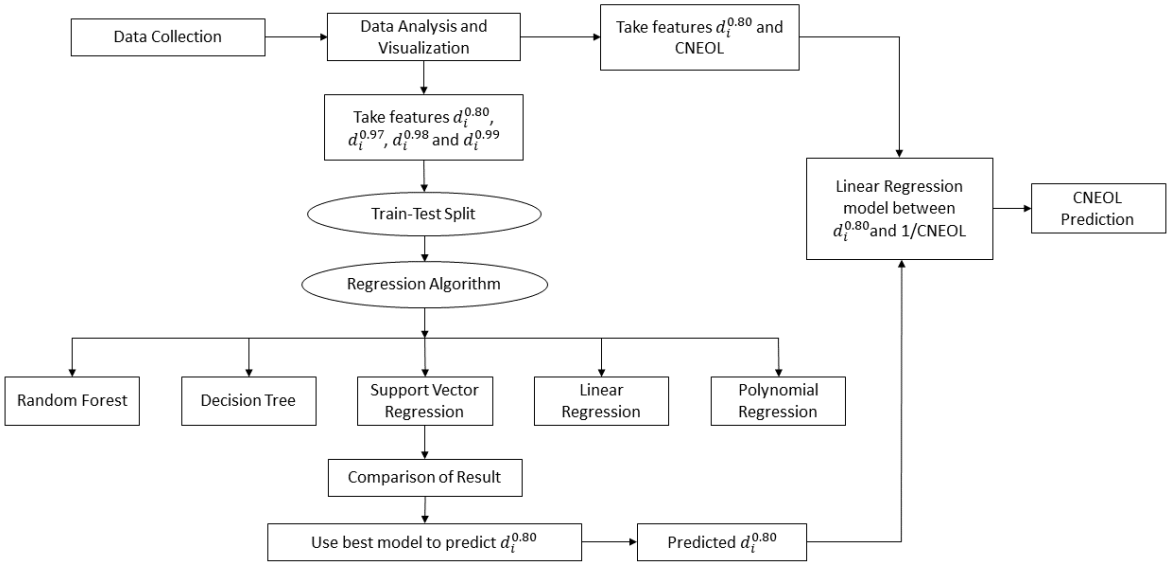


FIGURE 5.1. Flowchart of the process

5.2.1 Result of $d_i^{0.80}$ prediction

The final results for the prediction of $d_i^{0.80}$ with the five different machine learning algorithms across the three different feature sets are displayed in Table -I. Note that the random forests model performs consistently better than the other models, and does not have the overfitting issue decision trees do.

Figure -5.2 plots the test data accuracy for each machine learning model by feature set. Random forests perform better than all models for each feature set. Decision trees are next, but recall they suffer from overfitting. Support vector regression also performs well and is not overfit (and may in fact be underfit). Multiple linear regression and polynomial regression do not perform as well, particularly with only one feature $d_i^{0.99}$. I presented the detailed analysis of random forest and linear regression results in the previous section of this thesis. As noted, the random forest machine learning regression model performs the best so this model is utilized in the second stage to predict the cycle life.

TABLE I. $d_i^{0.80}$ prediction result from different regression techniques for three sets of features

	Three Features $d_i^{0.97}$,$d_i^{0.98}$,$d_i^{0.99}$		Two Features $d_i^{0.98}$,$d_i^{0.99}$		One Feature $d_i^{0.99}$	
	Testing Accuracy	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy	Training Accuracy
Random Forest	0.977	0.980	0.955	0.972	0.894	0.886
Decision Tree	0.975	0.997	0.955	0.998	0.876	0.998
Support Vector Regression	0.954	0.919	0.943	0.880	0.835	0.804
Multiple Linear Regression	0.838	0.810	0.629	0.614	0.423	0.400
Polynomial Regression	0.962	0.925	0.844	0.767	0.567	0.446

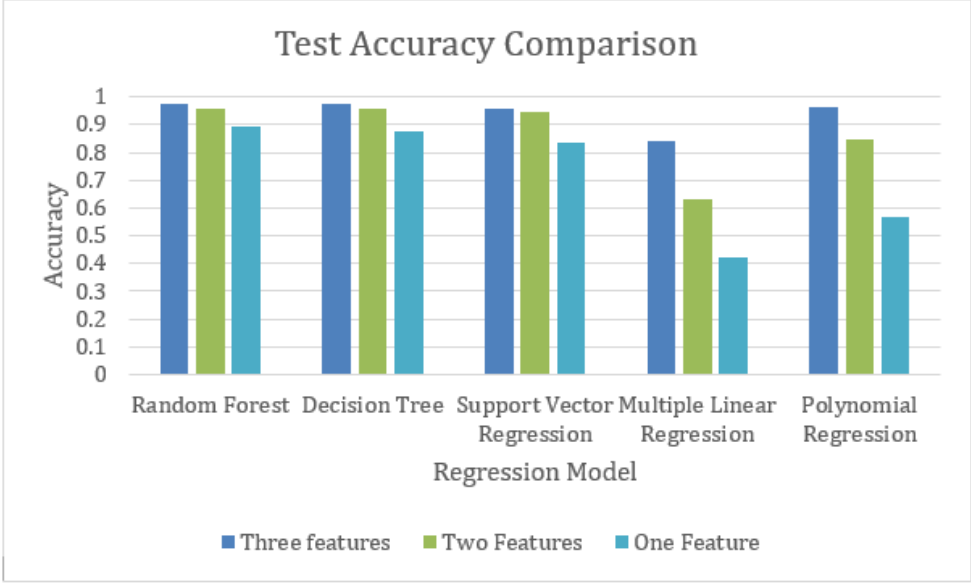


FIGURE 5.2. Accuracy comparison among different alghorithms

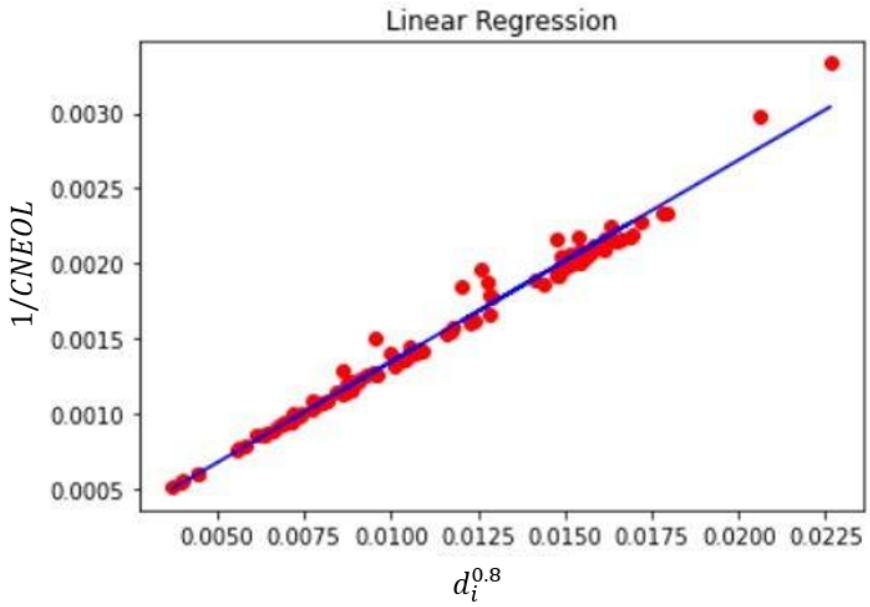


FIGURE 5.3. Linear regression model between 1/CNEOL and predicted $d_i^{0.80}$

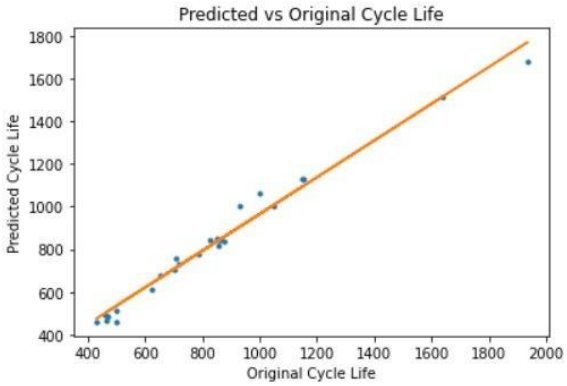
TABLE II. Performance measurement among three sets of features

	Three Features $d_i^{0.97}$,$d_i^{0.98}$,$d_i^{0.99}$	Two Features $d_i^{0.98}$,$d_i^{0.99}$	One Feature $d_i^{0.99}$
Correlation Value	0.99	0.98	0.94
RMSE	64.98	79.72	170.67
MAPE	4.05	5.91	9.69

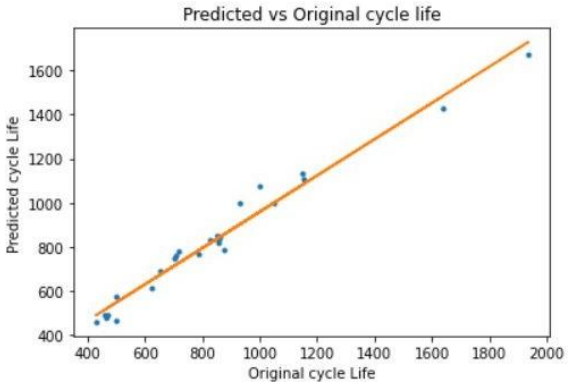
5.2.2 Result of cycle life prediction

Using this predicted $d_i^{0.80}$ value, a linear regression model was utilized to predict cycle life (or technically, 1/cycle-life). Figure -5.3 shows the linear regression model with equation $y = 0.1341x + 0.00000593$ and correlation value of 0.99. The predicted $d_i^{0.80}$ value from the random forests machine learning regression model is the input for this linear model and cycle life is the output.

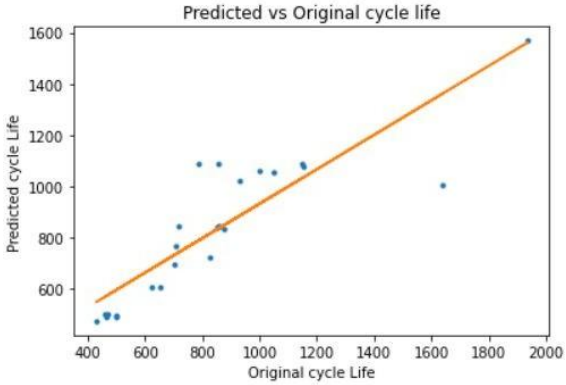
When we used three features and two features, the predicted value and original value of cycle life is almost accurate. But using single features, there is some deviation observable. The predicted CNEOL values are compared to the actual values. The predicted vs original values are strongly linear for the three- and two-features analyses, with a Pearson correlation value of 0.99 and 0.98, respectively. The single feature analysis is more loosely linear, with a correlation of 0.94. Table-II



(a) Using Three Features



(b) Using Two Features

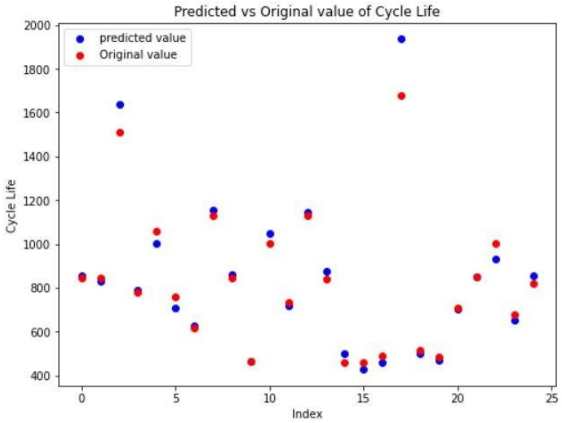


(c) Using One Feature

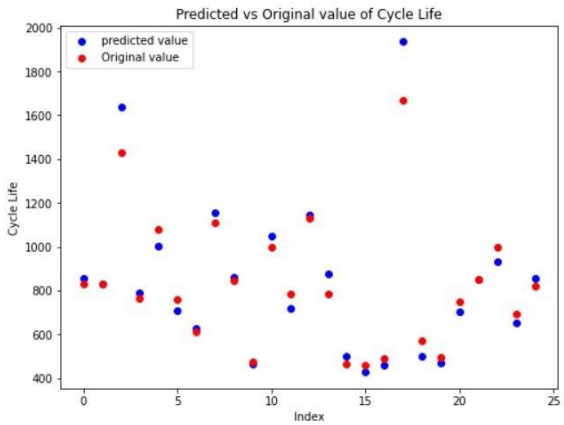
FIGURE 5.4. Predicted vs actual cycle life for testing data for three analysis

shows that RMSE is significantly lower for three feature and two feature models than single feature model. For three feature analysis, error (MAPE) is only 4.05%. In fact, error is less than 10% when we use only one feature. Figure -5.4 displays predicted vs actual cycle life for the different feature sets.

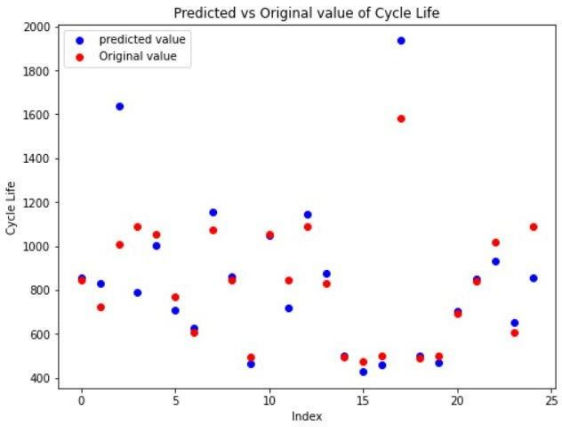
Figure- 5.5 shows the scatter plot of predicted and actual cycle life for each battery in the test set, across the three different feature sets. This gives a good visual picture of deviation. It is observed that the greatest deviations exist at high cycle life – which also contributes significantly to RMSE.



(a) Using Three Features



(b) Using Two Features



(c) Using One Feature

FIGURE 5.5. Scatter plot of predicted and actual cycle life for each battery

5.3 DISCUSSION

Among these three analyses, three features and two features both give very good results and very accurately predict the cycle life. The single feature analysis also gave a good result but has higher RMSE. Our data is limited due to the fact that we only have 124 battery cells. Because the majority of these batteries have a short to medium cycle life, the data is unbalanced. This is one of the reasons why the model is less accurate for batteries with a higher cycle life. The problem can be fixed if more data can be accumulated. Oversampling technique can be used to minimize this problem. However, I was able to achieve greater outcomes in this study than in previous studies using the same dataset [16].

Chapter 6. Conclusion

Data driven prediction of battery cycle life is a promising development for Lithium ion batteries in manufacturing. In this present research, I have developed a unique model of cycle life, building from the statistical model developed at Idaho National Laboratory [50], and including two connected regression models, to predict cycle life with an error of 4.05% MAPE. The exponential model developed by the INL gives us the 'd' value at different discharge capacities. Using the early cycle 'd' values, I predicted the battery cycle life with improved accuracy. Five machine learning regression algorithms were tested for predicting $d_i^{0.80}$: random forests, decision trees, support vector regression, multiple linear regression, and polynomial regression. Then I compared the results to find the best model and that best model (which is random forest) is utilized to predict $d_i^{0.80}$, and subsequently the cycle life using a linear regression model. The predicted $d_i^{0.80}$ from the random forest regression is the input for the second model. The final result of the models is the prediction of cycle life utilizing different feature sets which is either three features ($d_i^{0.97}$, $d_i^{0.98}$ and $d_i^{0.99}$), two features ($d_i^{0.98}$ and $d_i^{0.99}$), and a single feature $d_i^{0.99}$. Plots of predicted cycle life vs actual cycle life are displayed in 5.4, for each feature set.

From this study, one can also conclude that when a dataset exhibits multicollinearity and is highly skewed or non-normal, it is beneficial to use a model, such as tree-based regression, which is non-parametric and non-linear. The RFR model is fairly robust and outperforms all other models in terms of accuracy (achieving 97.7% testing accuracy). The RFR model uses all the features and offers insight into the relative importance of them. Interestingly, in this study the tree-based models also exhibited no predictive outliers, whereas the linear regression models did. If linear regression is to be used, this study demonstrates the benefit of transforming the data and utilizing stepwise regression to address multicollinearity and lack of normality. The MLR model which used data transformation provides better result than the MLR model without data transformation with

ideal variation inflation factor (VIF) and Durbin-Watson statistics value. The best MLR model also reduces the number of feature for prediction which is good for early prediction of cycle life.

The dataset is limited with only 124 battery cells. Most of these batteries have low to medium cycle life value, making the data imbalanced. This is part of the reason the model is less accurate for high cycle life batteries. With more data the model can be improved. Higher accuracy is achieved with my model than that reported in the previous work with same dataset [16] [18]. Mean absolute percentage error of test data from our best model is 4.05%, where this test error was 9.1% in [16]. Shan Zhu et al. [18] classify the Li-ion battery into higher and lower cycle life where in this research one can find out the value of cycle life (how many cycle battery can sustain before reaching cycle life).

Before supplying the Li-ion batteries to the consumer, this model might be used in a lithium-ion battery manufacturing company. With the help of this developed model and the early cycle discharge capacity, one may identify the battery that has a shorter cycle life. However, before being used on a large scale, this model should be tested on a variety of battery cells under various charging conditions.

6.1 FUTURE WORK

This study was done in a very limited time and funding. There are several ares which can be investigated and improved in future study:

- Increase the size of the dataset, in particular with more batteries of high cycle life battery.
- Utilize weighted sampling and biased sampling to address the lack of batteries with high cycle life.

- Evaluate machine learning models utilizing a smaller subset of the data to investigate how small a sample size is necessary to accurately predict battery performance.
- Include the battery discharge and charge cycle data, as well as temperature and other data available, in our machine learning prediction of cycle life.
- Further research is needed to determine the cause of outlier found in linear regression model.
- Investigate outliers in the dataset of both low and high cycle life.
- Modify machine learning models used and test other algorithms (such as neural networks if we include the charging cycle data).
- Further analyze physical behavior of good vs bad batteries during the life cycle.
- Develop operation recommendations for Li-ion batteries.
- Using functional data analysis for making data normally distributed.

REFERENCES

[1] Naoki Nitta et al. “Li-ion battery materials: present and future”. In: *Materials today* 18.5 (2015), pp. 252–264.

[2] Jason W. Osborne and Elaine Waters. “Four assumptions of multiple regression that researchers should always test”. In: *Practical Assessment, Research, and Evaluation* 8 (2002). DOI: https://doi.org/10.7275/r222-hv23.

[3] AF Schmidt and C Finan. “Linear regression and the normality assumption”. In: *J Clin Epidemiol* 98 (2018), pp. 146–151. DOI: https://doi.org/10.1016/j.jclinepi.2017.12.006.

[4] Matt N. Williams, Carlos Alberto Gomez Grajales, and Dason Kurkiewicz. “Assumptions of Multiple Regression: Correcting Two Misconceptions”. In: *Practical Assessment, Research, and Evaluation* 18 (2013). DOI: https://doi.org/10.7275/55hn-wk47.

[5] Akhil Garg and Kang Tai. “Comparison of statistical and machine learning methods in modelling of data with multicollinearity”. In: *International Journal of Modelling, Identification and Control* 18.4 (2013), pp. 295–313. DOI: https://doi.org/10.1504/IJMIC.2013.053535.

[6] Ranjit Kumar Paul. “Multicollinearity: Causes, effects and remedies”. In: *IASRI, New Delhi* 1.1 (2006), pp. 58–65.

[7] Claudio Agostinelli. “Robust Stepwise Regression”. In: *Journal of Applied Statistics* 29.6 (2002), pp. 825–840. DOI: https://doi.org/10.1080/02664760220136168.

[8] Jason W Osborne and Elaine Waters. “Multiple Regression Assumptions. ERIC Digest.” In: (2002).

[9] Jason Osborne. “Improving your data transformations: Applying the Box-Cox transformation”. In: *Practical Assessment, Research, and Evaluation* 15.1 (2010), p. 12. DOI: https://doi.org/10.7275/qbpc-gk17.

[10] Remi M Sakia. “The Box-Cox transformation technique: a review”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 41.2 (1992), pp. 169–178. DOI: https://doi.org/10.7275/qbpc-gk17.

[11] Leo Breiman et al. “Statistical modeling: The two cultures (with comments and a rejoinder by the author)”. In: *Statistical science* 16.3 (2001), pp. 199–231. DOI: https://doi.org/10.1214/ss/1009213726.

[12] Min Xu et al. “Decision tree regression for soft classification of remote sensing data”. In: *Remote Sensing of Environment* 97.3 (2005), pp. 322–336. DOI: https://doi.org/10.1016/j.rse.2005.05.008.

[13] Leo Breiman et al. *Classification and regression trees*. CRC press, 1984.

[14] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32. DOI: https://doi.org/10.1023/A:1010933404324.

[15] D Richard Cutler et al. “Random forests for classification in ecology”. In: *Ecology* 88.11 (2007), pp. 2783–2792. DOI: https://doi.org/10.1890/07-0539.1.

[16] Kristen A Severson et al. “Data-driven prediction of battery cycle life before capacity degradation”. In: *Nature Energy* 4.5 (2019), pp. 383–391.

[17] Tianhan Gao and Wei Lu. “Machine learning toward advanced energy storage devices and systems”. In: *Iscience* (2020), p. 101936.

[18] Shan Zhu, Naiqin Zhao, and Junwei Sha. “Predicting battery life with early cyclic data by machine learning”. In: *Energy Storage* 1.6 (2019), e98.

[19] Yapeng Zhou, Miaohua Huang, and Michael Pecht. “Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differential evolution optimization”. In: *Journal of Cleaner Production* 249 (2020), p. 119409.

[20] Jian Liu and Ziqiang Chen. “Remaining useful life prediction of lithium-ion batteries based on health indicator and Gaussian process regression model”. In: *Ieee Access* 7 (2019), pp. 39474–39484.

[21] Meru A Patil et al. “A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation”. In: *Applied energy* 159 (2015), pp. 285–297.

[22] Datong Liu et al. “Lithium-ion battery remaining useful life estimation with an optimized relevance vector machine algorithm with incremental learning”. In: *Measurement* 63 (2015), pp. 143–151.

[23] Yapeng Zhou et al. “A novel health indicator for on-line lithium-ion batteries remaining useful life prediction”. In: *Journal of Power Sources* 321 (2016), pp. 1–10.

[24] Yang Chang, Huajing Fang, and Yong Zhang. “A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery”. In: *Applied energy* 206 (2017), pp. 1564–1578.

[25] Ji Wu, Chenbin Zhang, and Zonghai Chen. “An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks”. In: *Applied energy* 173 (2016), pp. 134–140.

[26] Xunfei Zhou et al. “Cycle life estimation of lithium-ion polymer batteries using artificial neural network and support vector machine with time-resolved thermography”. In: *Microelectronics Reliability* 79 (2017), pp. 48–58.

[27] Lei Ren et al. “Remaining useful life prediction for lithium-ion battery: A deep learning approach”. In: *IEEE Access* 6 (2018), pp. 50587–50598.

[28] Phattara Khumprom and Nita Yodo. “A data-driven predictive prognostic model for lithium-ion batteries based on a deep learning algorithm”. In: *Energies* 12.4 (2019), p. 660.

[29] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24.2 (1996), pp. 123–140.

[30] RJ Hill and HD Flack. “The use of the Durbin–Watson d statistic in Rietveld analysis”. In: *Journal of Applied Crystallography* 20.5 (1987), pp. 356–361. DOI: https://doi.org/10.1107/S0021889887086485.

[31] Adam Hayes. *Stepwise Regression*. 2022. URL: https://www.investopedia.com/terms/s/stepwise-regression.asp#:~:text=Stepwise\%20regression\%20is\%20the\%20step,statistical\%20significance\%20after\%20each\%20iteration..

[32] Abhigyan. *Understanding Polynomial Regression*. 2020. URL: https://medium.com/analytics-vidhya/understanding-polynomial-regression-5ac25b970e18.

[33] Alex J Smola and Bernhard Schölkopf. “A tutorial on support vector regression”. In: *Statistics and computing* 14.3 (2004), pp. 199–222.

[34] Tom Sharp. *An Introduction to Support Vector Regression (SVR)*. 2020. URL: https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2.

[35] Mariette Awad and Rahul Khanna. “Support vector regression”. In: *Efficient learning machines*. Springer, 2015, pp. 67–80.

[36] Shovan Chowdhury and Marco P Schoen. “Research Paper Classification using Supervised Machine Learning Techniques”. In: *2020 Intermountain Engineering, Technology and Computing (IETC)*. IEEE. 2020, pp. 1–6. DOI: https://doi.org/10.1109/IETC47856.2020.9249211.

[37] Issoufou Ouedraogo, Pierre Defourny, and Marnik Vanclooster. “Application of random forest regression and comparison of its performance to multiple linear regression in modeling groundwater nitrate concentration at the African continent scale”. In: *Hydrogeology Journal* 27.3 (2019), pp. 1081–1098. DOI: https://doi.org/10.1007/s10040-018-1900-5.

[38] Marjan Čeh et al. “Estimating the performance of random forest versus multiple regression for predicting prices of the apartments”. In: *ISPRS international journal of geo-information* 7.5 (2018), p. 168. DOI: https://doi.org/10.3390/ijgi7050168.

[39] Animesh Agarwal. *Polynomial Regression*. 2018. URL: https://towardsdatascience.com/polynomial-regression-bbe8b9d97491.

[40] Masahito Kobayashi and Shinichi Sakata. "Mallows' Cp criterion and unbiasedness of model selection". In: *Journal of Econometrics* 45.3 (1990), pp. 385–395.

[41] Ryuhei Miyashiro and Yuichi Takano. "Subset selection by Mallows' Cp: A mixed integer programming approach". In: *Expert Systems with Applications* 42.1 (2015), pp. 325–331.

[42] Jeremy Miles. "Tolerance and variance inflation factor". In: *Wiley StatsRef: Statistics Reference Online* (2014).

[43] Trevor A Craney and James G Surles. "Model-dependent variance inflation factor cutoff values". In: *Quality engineering* 14.3 (2002), pp. 391–403.

[44] Jason Brownlee. *A Gentle Introduction to k-fold Cross-Validation*. 2018. URL: https://machinelearningmastery.com/k-fold-cross-validation/.

[45] Davide Anguita et al. "The 'K'in K-fold cross validation". In: *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. i6doc. com publ. 2012, pp. 441–446.

[46] Tadayoshi Fushiki. "Estimation of prediction error by using K-fold cross-validation". In: *Statistics and Computing* 21.2 (2011), pp. 137–146.

[47] Juan D Rodriguez, Aritz Perez, and Jose A Lozano. "Sensitivity analysis of k-fold cross validation in prediction error estimation". In: *IEEE transactions on pattern analysis and machine intelligence* 32.3 (2009), pp. 569–575.

[48] Will Kenton. *Durbin Watson Statistic Definition*. 2021. URL: https://www.investopedia.com/terms/d/durbin-watson-statistic.asp.

[49] Kenneth J White. "The Durbin-Watson test for autocorrelation in nonlinear models". In: *The Review of Economics and Statistics* (1992), pp. 370–373.

[50] Yuxiao Lin and Boryann Liaw. "A mechanistic binomial model for battery performance prediction and prognosis". In: (2020).