Department of Electrical Engineering

FINAL YEAR PROJECT REPORT

BENGEGU4-CDE-2020/21-HW-03

LED Lifetime Prediction based on Deep Learning

Student Name: YEUNG Tung Yan
Student ID: 55213094
Supervisor: Prof. WONG Hei
Assessor: Prof. TSE Michael

Bachelor of Engineering (Honours) in Computer and Data Engineering

Student Final Year Project Declaration

I have read the student handbook and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I declare that the work submitted for the final year project does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Student Handbook.

Project Title : LED Lifetime Prediction based on Deep Learning

Student Name : YEUNG Tung Yan

Student ID: 55213094

Signature

Date : 13/04/2021

Abstract

Accurate lifetime prediction of Light Emitting Diode (LED) is becoming increasingly pivotal among electronic components. Deep learning prediction model is an effective tool that harnesses the engineers to make estimation for lifetime of electronic components based on a number of parameters.

This project proposes a deep learning approach for lifetime prediction of LED by building data-driven predictive models with Deep Neural Network (DNN). The approach is applied to the real dataset of LED life from alumnus Mr. SO, Ho-Yin. Dataset includes performance hour, temperature, stressing current, input voltage, diode current, wavelength, and light intensity.

The DNN model is built with Classical Fully Connected Neural Networks, which output a numeric result. The first step of building a DNN is to prepare the dataset, including feature extraction, data cleaning, etc. After that, model is trained on the training set and evaluate the model on the testing set. Moreover, model is improved by regularization, optimization, tuning hyperparameters, etc. Finally, the model is implemented on the real dataset. The final deep learning model results show the estimated lifetime of LED under different conditions and color shift during the degradation process.

Acknowledgement

This project was supported by Prof. WONG, Hei at Department of Electrical Engineering in City University of Hong Kong throughout the whole academic year. Thank you for his guidance and support. Source of data was offered by alumnus Mr. SO, Ho-Yin of City University of Hong Kong. Thank you for his offer.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

The real dataset of LED life is conducted under stress test. In the experiment, gallium nitride based LEDs are being used. High drive current is being stressed which attributed to an increase in the temperature and decrease in the operation time of the LEDs.

An overview of the original dataset is shown in table 1. i and ii are controllable parameters, while iii to vii are measurable parameters. The value and rate of recording the sample are shown on the second and third column.

Parameter		Value	Difference
i)	Stressing Current	40 – 100mA	20
ii)	Input Voltage	-5 – 5V	0.05
iii)	Performance Hour	0 – 17 (max) hours	0.5 or 1
iv)	Temperature	25, 60°C	-
v)	Diode Current	-	-
vi)	Wavelength	-	-
vii)	Light Intensity	-	-

Table 1 Overview of original dataset

Original data are visualized in the following figures in the next page. Figure 1 shows the maximum light intensity over the spectrum versus hour group by LEDs. Figure 2 shows the average light intensity group by LEDs versus wavelength, which results in a spectrum graph.

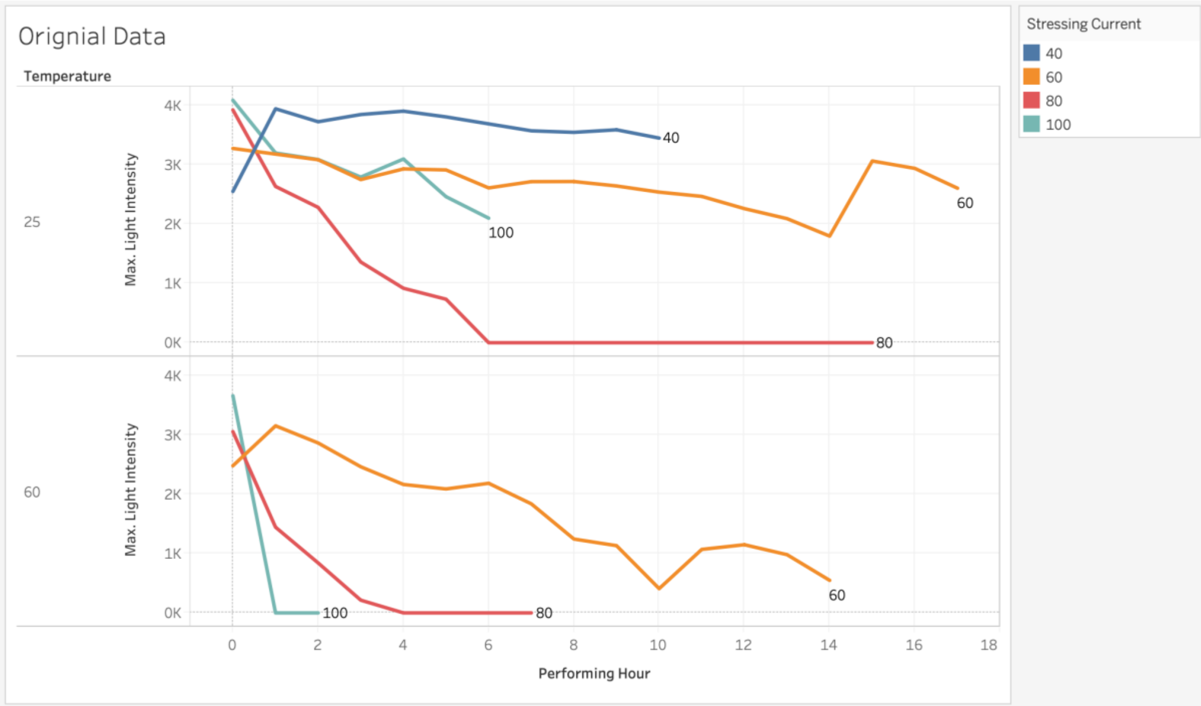

Figure 1 Original data of maximum light intensity vs performance hour

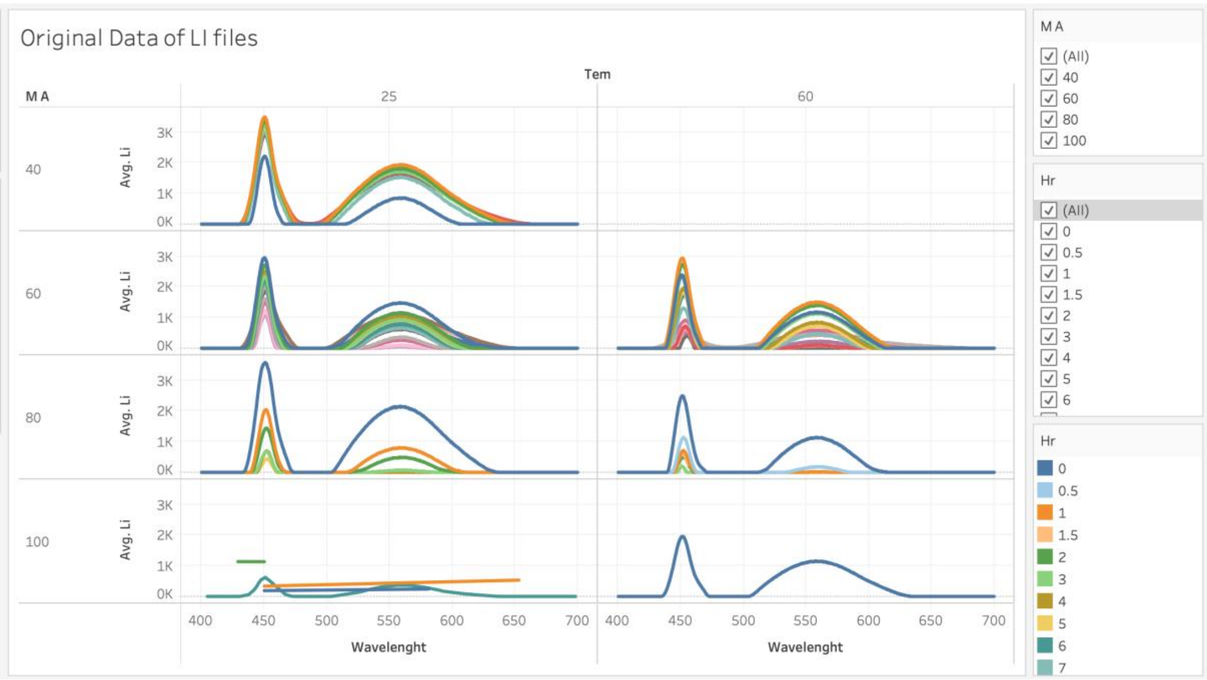

Figure 2 Original data of light intensity vs wavelength

In this project, the definition of “lifetime degradation for LED” is defined as the following two factors, one is decrease in light intensity, another one is color shift. The reasons will be explained in the feature selection part.

According to literature reviews and actual deployment of different methods, the DNN model is built with the following methods. In the feedforward process, ReLU is used as activation function and drop out as regularization method. In the backpropagation process, L1 norm is used as regularization method and RMSprop as optimization method. For model evaluation, Mean Absolute Error (MAE) and Mean Squared Error (MSE) are used as loss function and metrics for estimate model accuracy.

1.2 Objectives

The tech world is ever-changing and evolving. Electronic devices are getting complicated. In the past, one or few formulas could possibly be used to make predictions, for example, trend, seasonality, to name just a few using spreadsheet calculation derived from the past data. To predict the lifetime of an electronic device, a slew of formulas and calculations need to be applied.

However, for this kind of approach, results are not promised to be accurate as unknown factors potentially affect the results and lead to non-accurate results. Deep learning developments have introduced a data-driven approach to this issue. Using data without having to know any formulas or physics theories behind to predict the lifetime.

This project aims to develop a deep learning model showing the prediction of the lifetime and color shift of LED. How different parameters reducing the lifetime of LED will also be discussed. There are a number of factors that affect lifetime and are responsible for LED. These factors are suggested to be considered and accounted for in any experiments related to LED in order to minimize the error to give accurate and precise experimental results.

2. Methodology

Deep learning is a feasible method to predict the lifetime of a LED. In fact, deep learning technically is a subset of machine learning. In machine learning, the goal is to find the target function, which gives us a formula of the lifetime of a LED.

I assume the light intensity of a LED, denoted by *y*, correlates to the temperature, performing hour, stressing current, input voltage, wavelength (denoted by *x1, x2, x3, x4, x5* respectively). Therefore, there is a function, denoted by *f*, which give us the light intensity of a LED in a certain moment by *x1, x2, x3, x4, x5,* i.e. *f*(x1,x2,x3,x4,x5) = *y*. *x* is denoted as vector (*x1, x2, x3, x4, x5*), so that *f(x)* give us *y*. Aiming to know the lifetime of a LED at different conditions, which the way is to find out by *f*, as knowing about when the light intensity reaches a low level as the LED is unusable.

Of course, target function *f* is unknown. However, estimating it can be tried. The potential formula is called which has the same output of *f* as hypothesis, denoted by *h*, and the set of all potential hypotheses is called hypothesis set. The goal of us is to pick the final hypothesis from the hypothesis set, denoted by *g*, in which *g(x)* is close to *f(x).*

As having a set of sample data, denoted by *D*, a hypothesis *h* can fit well on D. The different *h(x)* and *f(x)* is called error, denoted by *E(h)*. *E* of data in *D*, called error in sample, *E_in(h)*, *E* of data out of *D*, called error out sample, *E_out(h)*. Calculate *E_in(h)* of a hypothesis by some method, but *E_out(h)*is always a unknow. Even if *E_in(h)* belongs to a acceptable range, the *E_out(h)* can be so large. If the model is feasible, difference between *E_in(h)* and *E_out(h)* should be low.

By Hoeffding's inequality, [1]

$$P(\,|E_{in}(h) - E_{out}(h)| > \varepsilon) \le 2e^{-2\varepsilon^2 N} \quad \text{...equation 1}$$

where, N is the size of *D*, ε is tolerance error. By equation 1, as N is large enough, the probability that the difference between *E_in(h)* and *E_out(h)* becomes larger than the tolerance is low. It means that if there has enough sampling data, $g(x)$ perform well on both sample data and out of sample data.

Once having an estimator of LED light intensity, in this project it will be a deep neural network, the error between the estimation on LED light intensity and the actual result can be calculated. There is an upper bound of probability that the difference between the performance of the estimator on in-sample data and out sample data. Therefore, machine learning is a feasible method for LED lifetime prediction.

The General Data Protection Regulation (GDPR) of the European Union took action in 2018, since then, a growing awareness and issues about the interpretability of Machine Learning algorithms have been risen. Accuracy versus interpretability has become a major concern when deciding what methods to use in the area of Machine Learning.

Deep Learning is a subset of Machine Learning. Interpretability of some traditional Machine Learning algorithms prevails over Deep Learning, while Deep Learning outperforms traditional Machine Learning algorithms with the increment of data size. Since achieving high accuracy (performance) is the crucial goal for lifetime prediction model, Deep Learning is used in this project.

2.1 Comparison of Methodology

Linear regression and GBDT (Gradient Boosting Decision Tree) are the examples of traditional Machine Learning algorithms commonly used in regression problems. The detailed reasons for choosing the methodology are explained below.

Linear Regression is a very traditional model, technically a baseline of machine learning. For prediction models, if applying the actual data for making estimation and the data fits in the linear model, a good result can be obtained. This is an ideal option for working on a prediction model if the straight line has low variance as the Sums of Squares are very similar for different datasets. This might give good predictions consistently. However, if the actual data does not fit the straight line to the training set, it is required to do extra work on feature selection (e.g. polynomial featuring, synthetic featuring, feature cross, etc) and more handcrafted rules in order to improve the model. This part of work is considerably a subjective process. And, the bias, which is the inability to capture the true relationship exists. No matter how well the actual data fit to the model, the straight line can never capture the true relationship between the parameters the project wishes to compare. For the reason that the goal is to attain an accurate prediction model, the option of Deep Learning is over Linear Regression in this project.

Principle of GBDT is classification or regression (linear or non-linear). A literature about GBDT is being reviewed. GBDT has been successfully applied in Remaining Useful Life Prediction for Lithium-Ion Battery.[2] It is considered one of the best Machine Learning technique to be employed to examine the non-linear effect. Not much work of data featuring engineering is needed. GBDT introduces a strong regression ability for RUL prediction model based on the extracted features. This makes GBDT to be one of the main competitors of methodology in this paper. Nonetheless, parameter selection of this method is hard. Some parameters, to name just a few, learning rate and leaf nodes, determine the structure of the model that affects the prediction accuracy. This leads to extra work of developing appropriate strategies to determine the parameters. Taking this into account, Deep Learning is a more ideal option compared to GBDT.

In this paper, Deep Neural Networks (DNN) is chosen to build the lifetime prediction model of LED. A research of occurrence of a variety of disease prediction has been done by DNN and GBDT. In the literature, the results show DNN outperforms GBDT with using lesser amounts of data. Despite this, it demonstrates that DNN is indeed capable of obtaining promising recognition accuracy on a separated testing dataset as compared to traditional regression method.[3] Ideally, the hidden neurons are helping us to synthetic featuring, feature cross, etc. Also, the linearity of DNN is lower than linear regression. Furthermore, there are work using the approach of regression have done on this topic and other prediction models. Applying deep learning on the lifetime prediction of LED would be a pioneer case.

However, the reasons explained above do not conclude that DNN is the best model for solving the problem in this project. Ensemble modelling is a commonly used method on machine learning, which is a method in which several different models are developed to predict an outcome, either through the use of many different modeling algorithms or through the use of different training datasets. It is possible to get an even more reliable model by ensemble DNN and other traditional machine learning models in this project. By considering the size of dataset, the focus will only be on implementing DNN on LED lifetime prediction.

2.2 Architecture

The following part will be explained followed by the architecture hierarchy as drawn in Figure 3.

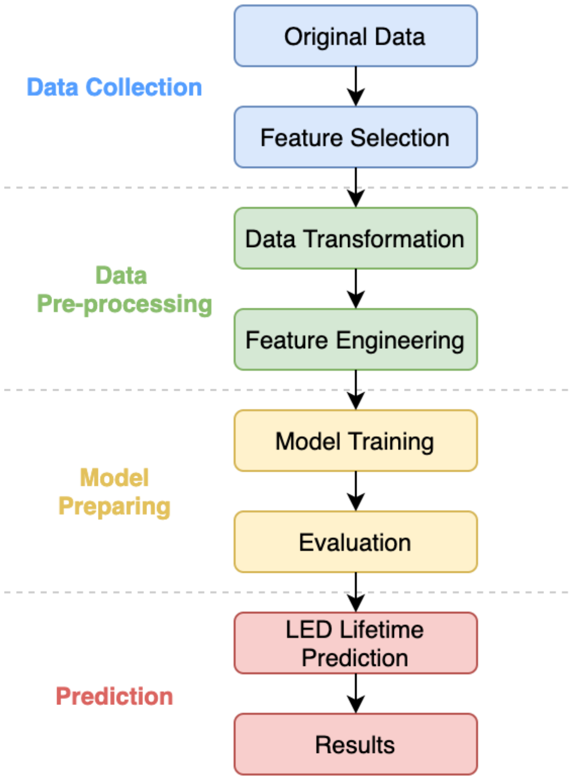

Figure 3 LED lifetime prediction model framework

Figure 3 shows the LED lifetime prediction model framework. Firstly, in the data selection step, performance hour, temperature, stressing current, wavelength, and light intensity are being selected. Input voltage and diode current are not considered. The reasons for data selection have been covered in the background section.

Secondly, the data preprocessing stage includes two steps: data transformation and feature engineering. Figure 4 shows the original data files printed in xxC means temperature, xxmA means stress current and xhr means performance hour in the title.

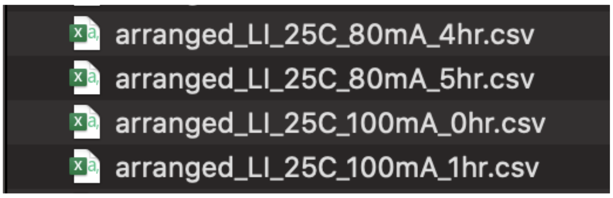

Figure 4 Spreadsheet titles

Since the data of temperature, stress current and performance hour are not included inside the spreadsheets, the next step would be adding these parameters into each data spreadsheet. The later step of data transformation is to combine all the separated data spreadsheets into one dataframe as shown in Figure 5. After the data transformation, the step of combination of spreadsheets and transforming them into one dataframe led to duplicated records. So, data cleaning has done, which loops the data frame once and remove those duplicated records.

	wavelenght	tem	mA	hr	Li	LED
0	400.04	25	60	12.0	0.0	A
1	400.42	25	60	12.0	0.0	A
2	400.80	25	60	12.0	0.0	A
3	401.18	25	60	12.0	0.0	A
4	401.56	25	60	12.0	0.0	A
...	...	...	...	...	...	...
774	698.09	60	60	13.0	NaN	E
775	698.48	60	60	13.0	NaN	E
776	698.87	60	60	13.0	NaN	E
777	699.26	60	60	13.0	NaN	E
778	699.65	60	60	13.0	NaN	E

245470 rows × 6 columns

Figure 5 Dataframe after data transformation

Feature engineering is the method of converting raw data into features that better describe the core problem to predictive models, improving model accuracy on previously unseen data. Since the original samples for temperature are 25 and 60 degree celsius. If the temperature in Celsius scale is being fed to the model, it is possible for the model to learn such a large multiplier between two values, which is 2.4 times as shown in Table 2. As aiming the model to learn the difference between two values, the straightforward choice would be to switch the scale by using Kelvin, the absolute thermodynamic temperature scale.

	Temperature	Multiplier	Difference
Celsius	25, 60	2.4x	35
Kelvin	298.15, 333.15	1.117x	35

Table 2 Temperatures before and after feature engineering

Regardless of the temperature scale, feature engineering has been worked on LED labels (i.e. LED A - E) for overfitting prevention. In this early stage, prevention of the model overfit has been attempted. Due to the limited samples that there are only five LEDs for each condition, hiding the LED labels would be a solution. Also, it is not necessary for the model to give identification for LED, so hiding the LED labels is a viable way to prevent overfitting as an early regularization method.

Thirdly, the model preparing stage includes model training and evaluation. Architecture and details of the model training will be discussed in the next part. Evaluating the loss by cross validation will be used in the evaluation step.

Lastly, LED lifetime prediction will be presented in data visualisation by light intensity and lifetime, while color shift will be presented by wavelength, light intensity and lifetime.

2.3 Feature Selection

Due to the reason that European Community (EC) published international standards on how lifetime should be declared on LED luminaires[4], light intensity of an LED drops to zero would be defined as failure mode (i.e. end of lifetime) of LED. Apart from this, from the paper published by the Department of Energy in the US[5], the content states that luminaires tend to discolor due to oxidation effects during ageing and leading to shifts in the yellow direction in the colour spectrum. Second goal of the project is to verify and evaluate the color shift of LED during the degradation process.

The lifetime of a LED can be given by the light intensity of it after different performing time. When its light intensity drops to zero, the elapsed time as its lifetime has been denoted.

As the color of visible light depends on its wavelength, the color shift of a LED can be given by the change of light intensity of it at different wavelengths after different performing time. The color of a LED depends on the wavelength which has the highest light intensity. After different performing time, when the wavelength which has the highest light intensity change, color shift has happened.

The lifetime and color shift of a LED at different temperatures and stressing currents can be given by the light intensity of it at different wavelengths and different performing time.

Therefore, stress current, performance hour, temperature, wavelength and light intensity are selected as features.

2.4 Model Training

The model is made up of interconnected basic processing factors that are connected together. Each neural network relation has a weight assigned to it. One of the most widely used machine learning techniques for multi-layer networks is often the Feedforward with Backpropagation Neural Network algorithm. An input layer, hidden layers, and an output layer are all common components of a Feedforward Backpropagation Neural Network. The elements within the network are interconnected in the feedforward process. The link weights have been set to their initial values. How the model minimizes the error between the actual value and the predicted output value is that for the weights to be revised, the error term is back propagated across the network.

The architecture of the DNN model is composed of one input layer with five nodes, {wavelength, light intensity, temperature, stress current, performance hour}, four fully connected hidden layers, and one output layer with one node, {lifetime} as shown in Figure 6.

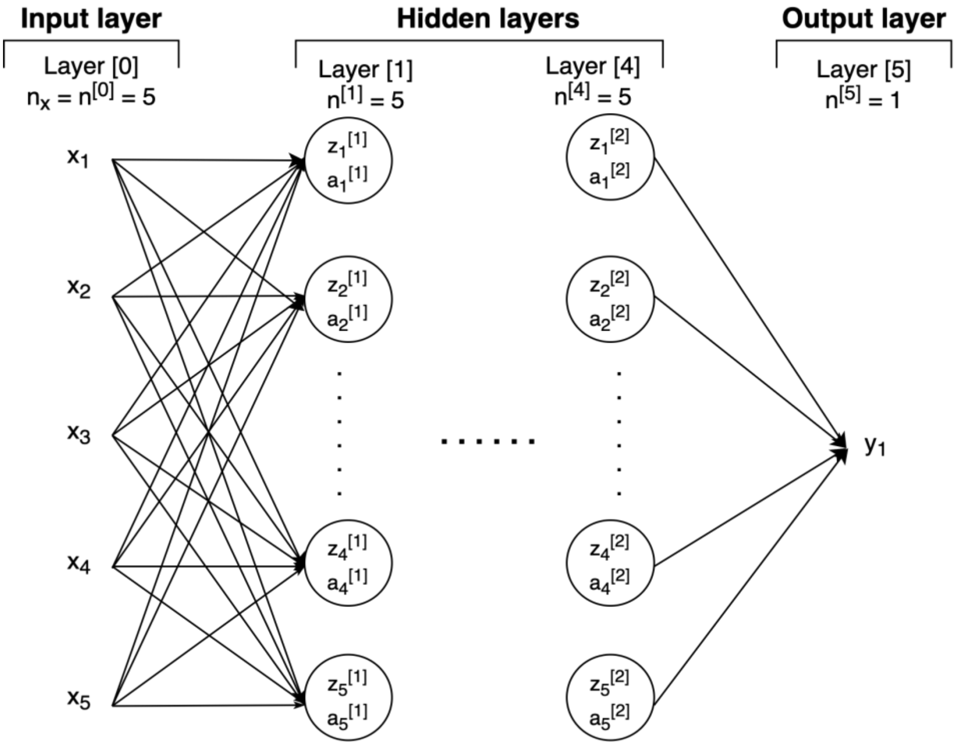

Figure 6 DNN model architecture

The fundamental equation for the DNN model is:
At the lth layer, for the ith node of it, denoting it as ${n_i}^l$, and the activation values of it as ${a_i}^l$, the transformation values of it as ${z_i}^l$, the bias of it as ${b_i}^l$. Denote the weight connecting nodes ${n_{i_1}}^l$and ${n_{i_2}}^{l+1}$ as ${w_{i_1 i_2}}^{l+1}$. The transformation value of a node is the value of it after computing the sum of the product of the weights and output value from the last layer and the bias.

$$\text{i.e. } {z_i}^l = \left(\sum_{i'=1}^{n^{[l-1]}} {w_{i'i}}^l \times {a_{i'}}^{l-1}\right) + {b_i}^l \text{ ... } \textit{equation 2,}$$

where $n^{[l-1]}$is the number of nodes in the $l-1$ th layer.

The transformation value is not the final output of the node as long as there is an activation function and a dropout step at every node in hidden layers.

The activation function proposed in this paper is ReLU, denoted by f.

$$\text{i.e. } f(x) = max(0, x) \text{ ... } \textit{equation 3}$$

For the dropout, denote the dropout rate in the nodes in the l th layer by p^l, which mean that every node the that layer have a probability p^lbecome non-activated. Dropout only applied in the training phase but not the testing phase or implement phase.
Therefore, in the training phase, only about 1-p^l out of all nodes will be activated, it will cause an increase in output value in the testing phase compared to the training phase. To offset the difference of output value in training phase and testing phase, the output value of nodes with dropout will decrease in a ratio p^l.

$$\text{i.e. } {w_{i_1 i_2}}^{l+1}_{test} = p^l \times {w_{i_1 i_2}}^{l+1}_{train} \text{ ... } \textit{equation 4}$$

With dropout, the activation value of a node is

$$r_i{}^l \sim Bernoulli(p^l) \quad \textit{... equation 5}$$

$$a_i{}^l = f(\, z_i{}^l \times r_i{}^l),$$

where $Bernoulli(p^l)$ mean that

$P(r_i{}^l = 0) \; = \; p^l$

$P(r_i{}^l = 1) \; = 1 - \; p^l$

Therefore, the output value of a node in DNN can be express as

$$a_i{}^l{}_{train} = max(0,\; (\sum_{i\prime=1}^{n^{[l-1]}} w_{i\prime i}{}^l \times a_{i\prime}{}^{l-1} + b_i{}^l) \times r_i{}^l) \quad \textit{... equation 6}$$

$$a_i{}^l{}_{test} = max(0, (\sum_{i\prime=1}^{n^{[l-1]}} p^l \times w_{i\prime i}{}^l \times a_{i\prime}{}^{l-1} + b_i{}^l)) \quad \textit{... equation 7}$$

(for $l \in$ {1, 2 . . . , L }, L is the total number of layers in the network)
When $l = 0$, it is an input node. In this network the activation function of output value in $n^{\;L}$, which is the final output of the network, is also proposed to be ReLU as the value represents a light intensity, and a light intensity should not be lower than 0.

During the model training phase, it feeds some of the training data to the model. It compares the output of the model and the actual data. Use a loss function, denoted by C, which is a function of set parameters, denote by $\theta = [\theta_1, \theta_2, \theta_3 \ldots\ldots]$, $\theta^* \; = w_{i\prime i}{}^l \,, b_i{}^l$, to quantify the difference between the output of the model and the actual data. In this paper, MSE is proposed as the loss function. Also, L1 normalization is proposed as the regularization method.

$$C(\theta) = (y - \hat{y})^2 - \sum \; |w| \quad \textit{... equation 8}$$

$$C(\theta) = (y - a^L)^2 - \sum \; |w| \quad \textit{... equation 9}$$

The goal of backpropagation is to decrease and minimize the output of loss function. Once a loss in a training step is being calculated, use gradient descent to minimize it by tuning the parameters.

$$\frac{\partial C}{\partial a^L} = 2(y - a^L)(-1) = -2(y - a^L) \text{ ... } \textit{equation 10}$$

$$\frac{\partial C}{\partial a^L} = 2(y - f(\,(\sum_{i\prime=1}^{n^{[l-1]}} w_{i\prime i}{}^{l} \times a_{i\prime}{}^{L-1} + b_i{}^{l}) \times r_i{}^{l}))$$

$$\frac{\partial C}{\partial w^L} = \frac{\partial C}{\partial a^L} \times \frac{d\ a^L}{d\ w^L}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \times \frac{d\ a^L}{d\ b^L}$$

In this paper, ReLU as an activation function and Dropout as a regularization method for hidden layers in the feedforward process are being proposed, while Root Mean Square Propagation (RMSProp) as a optimization method and Lasso normalization (L1 norm) as a regularization method for hidden layers in the backpropagation process.

2.5 Activation functions

Activation function is used in forward propagation and its derivative in backpropagation. Sigmoid, tanh, and ReLU are commonly used activation functions. [6]

Sigmoid activation: $s(x) = \frac{1}{1+e^{-x}}$ ... *equation 11*

Derivative of sigmoid: $s(x)' = \frac{e^{-x}}{(1+e^{-x})^2}$ ... *equation 12*

$$= \frac{1 + e^{-x} - 1}{(1 + e^{-x})^2}$$
$$= \frac{1}{1 + e^{-x}} - (\frac{1}{1 + e^{-x}})^2$$
$$= x - x^2 = x(1 - x)$$

Tanh activation: $t(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ ... *equation 13*

Derivative of tanh: $t(x) = \frac{(e^x - e^{-x})(e^x - e^{-x})\prime - (e^x - e^{-x})(e^x + e^{-x})\prime}{(e^x + e^{-x})^2}$ ... *equation 14*

$$= \frac{(e^x - e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2}$$
$$= \frac{(e^x + e^{-x})^2}{(e^x + e^{-x})^2} - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$
$$= 1 - x^2$$

ReLU activation: r(z) = max (0, x) ... *equation 15*

Derivative of ReLU: r(z)′ = 0, x < 0 ; 1, x > 0 ... *equation 16*

Function	Activation	Derivative
sigmoid	$\frac{1}{1+e^{-x}}$	$x(1-x)$
tanh	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1-x^2$
ReLU	max (0, x)	$0, if\ x < 0$ $1, if\ x > 0$

Table 3 Formulas of the activation functions sigmoid, tanh and ReLU

From the table 3 showing final result of the formula for each activation function, the following conclusions can be derived.

For binary problems, Sigmoid performs better at the output layer. It is not recommended for use in hidden layers. Tanh performs better in hidden layers than sigmoid because the average of the outputs of tanh is close to zero, which helps to center the data for the next layer. However, it prones to vanishing gradient problems. From the formulas written above, sigmoid and tanh functions are close to the saturation region, and after the derivation, it approaches 0, which is the so-called vanishing gradient problem, which makes the updated information unable to be transmitted through back propagation. Since the gradient of ReLU is always 1 for positive values, it is a nice choice. On the other hand, gradient of ReLU is 0 for negative values. ReLU effectively overcomes the vanishing gradient problem.

2.6 Regularization method

Dropout

During training, units (along with their connections) are dropped at random from the neural network. From a reliable study finding, dropout greatly eliminates overfitting and provides substantial improvements over other regularization approaches. [7]

For standard neural networks without dropout, when training plural models with the same training data at the same time, usually different results from each of them would be gotten, especially in a regression task which requires numeric result because of the randomness of model training. By the thought of model ensemble, it's often to use an average of result or 'majority decision' method to decide the final result among these models. This strategy is effective in improving the accuracy and reliability of the prediction on model implementation, as long as one single model may easily be overfitted on some specific input, but plural models can complement each other to offset the overfitting.

For neural networks with dropout, because of dis-activation of nodes, the structure of the model is technically changed during the training process. It's similar to training a distinct model on every training data, and assembling them in the testing process. To a certain extent dropout can enhance the generality of a neural network just like the effect of model ensemble.

L1 Norm Regularization

It helps us lower the weighting of the model. A common way to prevent overfitting.
L1 norm regularization and L2 norm regularization are usually proposed at the same time, as they have the same idea to prevent overfitting by adding a penalty term in the loss function which is the sum of value of weights connecting the nodes in the model. [8]

As reducing the complexity of a model is usually a useful way to prevent overfitting, reduction on values of weights can reduce the complexity of a model. To a certain extent a model can fit all of the training data when train it without any limitation on it, which is overfitting. Therefore, with a penalty of sum of weights, the model is required to make a trade-off between the error and the model complexity, which represent a balance on overfitting and underfitting.

The key difference between L1 and L2 norm regularization is the penalty term, which L1 norm regularization is using the absolute value of sum of weights and L2 norm regularization is using the squared value of sum of weights. Although both L1 and L2 norm regularization help the model to reduce the weights, L1 norm regularization trend shrinks the less important node's weight to zero that help us on doing a feature selection and L2 norm regularization doesn't. Therefore, L1 norm regularization is proposed in this project.

2.7 Optimization method

RMSprop [9]

In neural network training, a gradient-based optimization technique is used. RMSprop allows the learning rate changes over time. Learning rate is a factor that should not be neglected because it determines whether the model can miss segments of the data. The model may ignore subtler aspects of the data if the learning rate is high. It is desirable for real-world applications if it is low.

3. Results

3.1 Loss function

This part introduces regression accuracy of the DNN model. Regression is an error minimization problem and the regression metrics of the model are Mean Absolute Error (MAE) and Mean Squared Error (MSE).[10] Cross validation is a method for estimating test performance by defining a collection of train or test splits. The discrepancy between the real and expected values is represented by the cross validation loss function. If the value of the loss function is high, it indicates that the model has a lot of variance in its output and should be corrected. LED lifetime can be estimated using a trained DNN model. MAE and MSE are used to test the performance of deep neural network models in terms of prediction accuracy.

MAE formula is as in,

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - y'_i| \quad \text{... } \textit{equation 18}$$

n is the test set size, y is the sample predicted value and y' is the actual value of sample i.

MAE calculates the average number of errors in a series of predictions without taking into account the direction. It is the average of the absolute differences between prediction and actual observation over the test sample, with equal weight given to all individual differences.

MSE formula is as in,

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - y'_i)^2 \quad \text{... } \textit{equation 17}$$

n is the test set size, y is the sample predicted value and y' is the actual value of sample i.

It is preferable if the MSE value is as low as possible; the lower the MSE value, the better the predictions of the model are. A higher MSE means that the predicted and actual values differ significantly.

The results of the training history of MAE and MSE loss are printed as a table as shown in Table 4. And, the results of MAE and MSE loss are plotted as graphs as shown in Figure 7 and Figure 8 respectively. Observation from the graphs is that the errors decreased dramatically in the early stage of training and the drop slowed down noticeably afterwards. As the difference of two loss function score is small, conclude that the model has no overfitting. This means that the model accuracy is high. Furthermore, during each epoch, training loss is computed, while validation loss is computed after each epoch. This means that training losses are estimated half an epoch earlier on average. Therefore, both MAE and MSE graphs show a very small gap between the line of train error of validation error. The small gap between the training and losses values would be eliminated if the training losses were moved half an epoch to the left.

	loss	mse	mae	val_loss	val_mse	val_mae	epoch
45	185287.984375	185287.984375	273.504333	189491.37500	189491.37500	273.311066	45
46	185665.968750	185665.968750	273.701294	195768.71875	195768.71875	290.036346	46
47	185389.937500	185389.937500	273.600769	230514.93750	230514.93750	302.171509	47
48	183914.937500	183914.937500	272.492950	206340.53125	206340.53125	285.707123	48
49	183646.406250	183646.406250	272.391083	253555.62500	253555.62500	313.534607	49

Table 4 Model training history

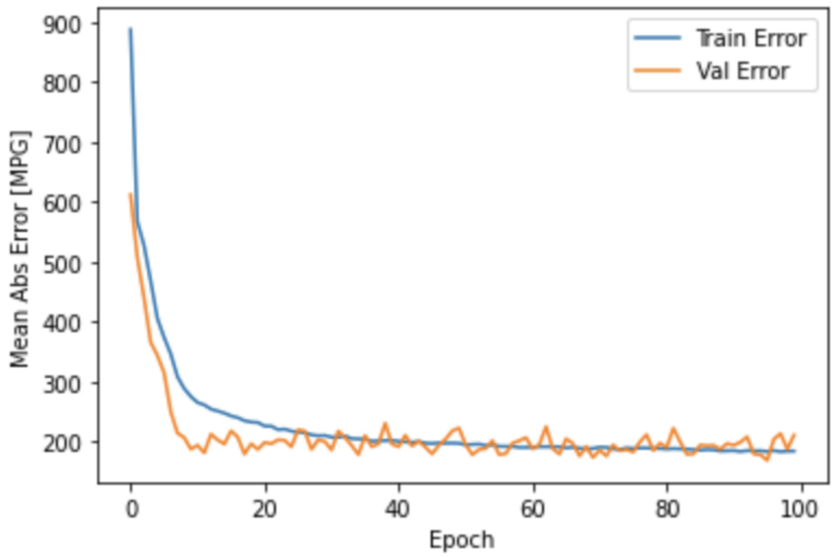

Figure 7 Plotted MAE graph of the DNN model

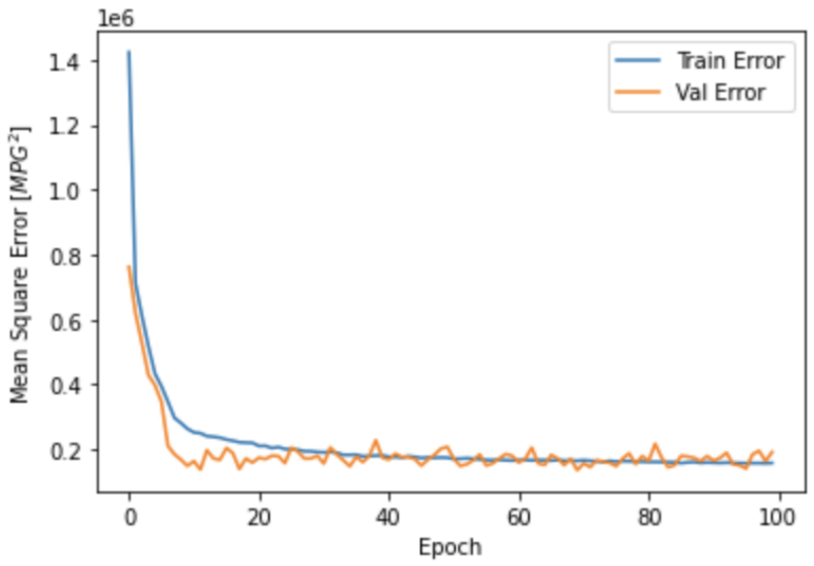

Figure 8 Plotted MSE graph of the DNN model

3.2 Comparison of Optimization and Regularization methods

In this project, SGD, Adam, Nadam and RMSprop are the methods that have been tried to apply on model training and a comparison on them of the mean absolute error of prediction on the testing set has been done with 50 epochs.

SGD stands for mini-batch gradient descent in this project, rather than stochastic gradient descent, which is more widely used nowadays. SGD updates the parameter according to the product of their gradient and the learn rate which is a hyperparameter defined by myself. The disadvantage of SGD is that the learn rate is difficult to be set as it is a constant during model training. Also, the model optimized by SGD is easily trapped in critical points. Therefore, it results in the MAE and MSE loss as plotted in Figure 9 and Figure 10.

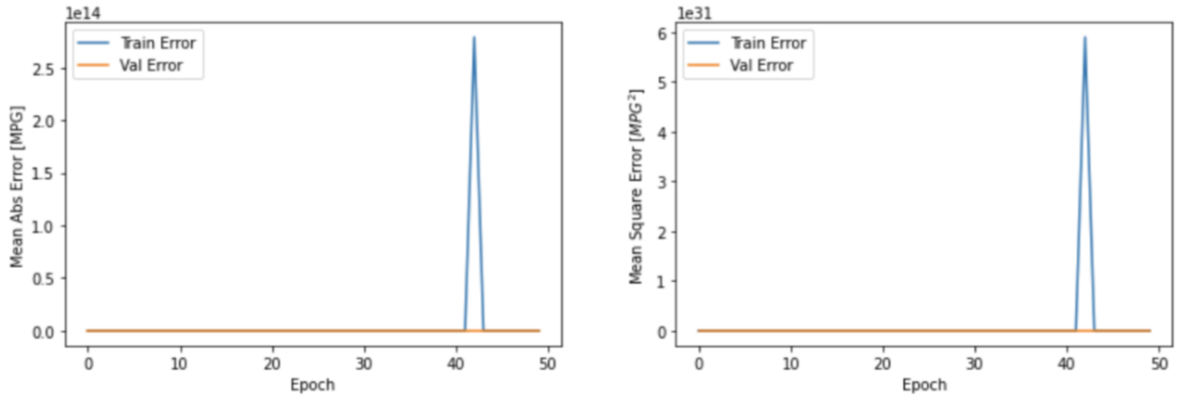

Figure 9 MAE loss (left) and MSE loss (right) of SGD L1 approach

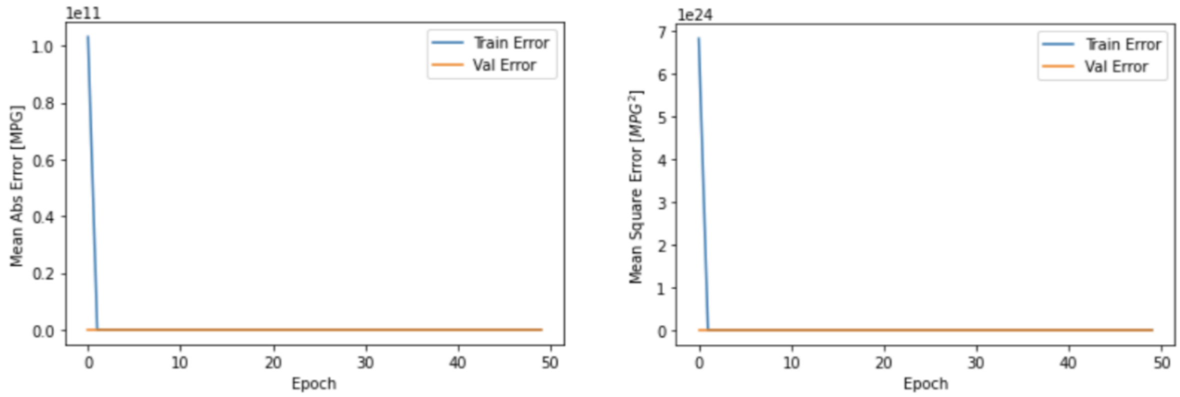

Figure 10 MAE loss (left) and MSE loss (right) of SGD L2 approach

RMSprop is a variant of Adagrad and adadelta, which have a regularization on the learning rate upon general optimizer so that the learning rate changes over time. A dynamic learning rate allows the optimizer to accurately optimize the model in the early stage of model training and slow it down at the later stage. RMSprop is good at handling non stationary targets. MAE and MSE loss of RMSprop are plotted in Figure 111 with L1 approach and Figure 12 with L2 approach.

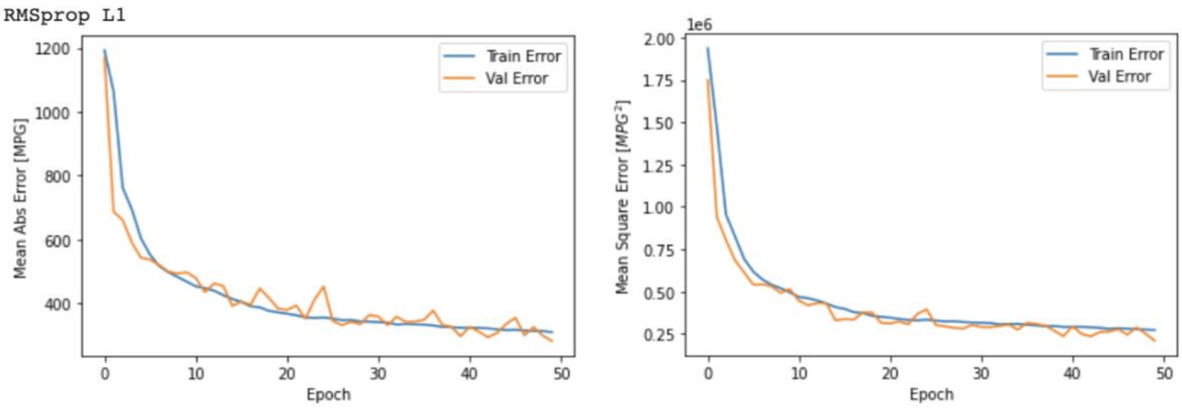

Figure 11 MAE loss (left) and MSE loss (right) of RMSprop L1 approach

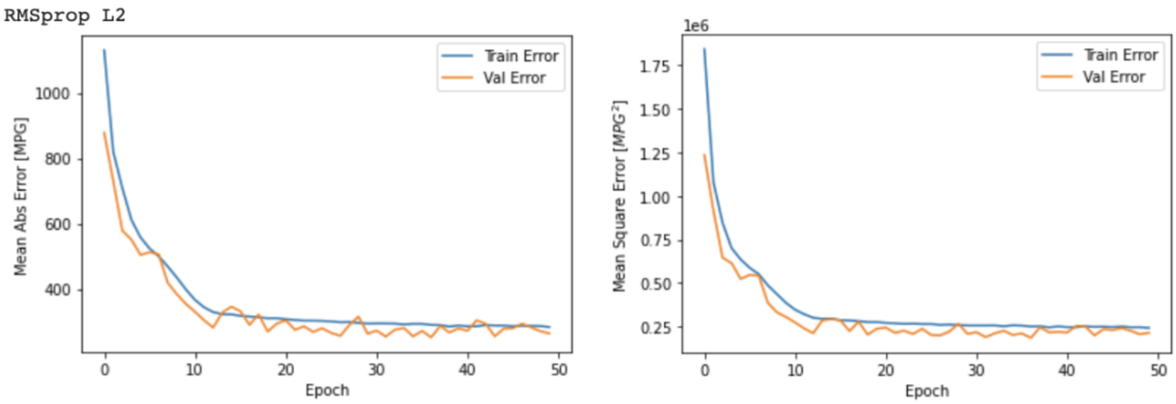

Figure 12 MAE loss (left) and MSE loss (right) of RMSprop L2 approach

Adam essentially is a RMSprop with momentum, which allows accurately optimizing when the gradient is close directed to the last gradient to simulate momentum in the physical world. MAE and MSE loss of Adam are plotted in Figure 13 with L1 approach and Figure 14 with L2 approach.

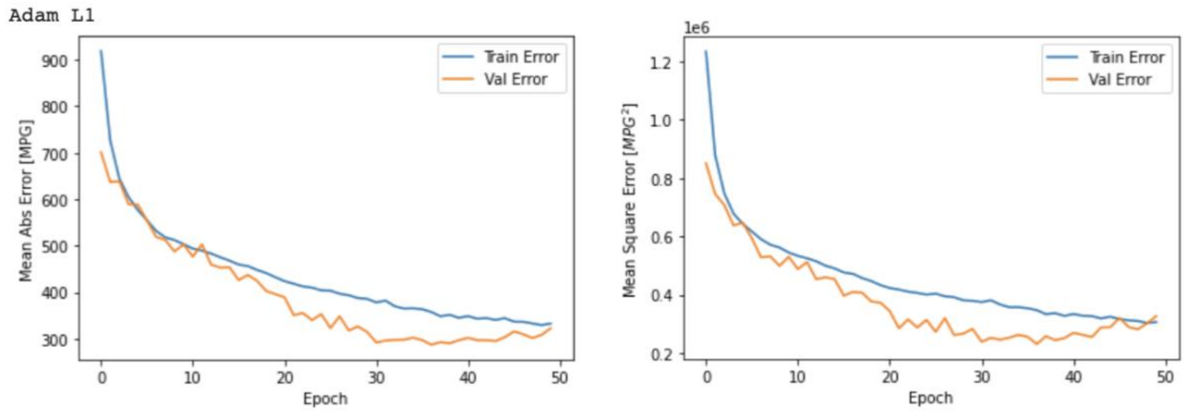

Figure 13 MAE loss (left) and MSE loss (right) of Adam L1 approach

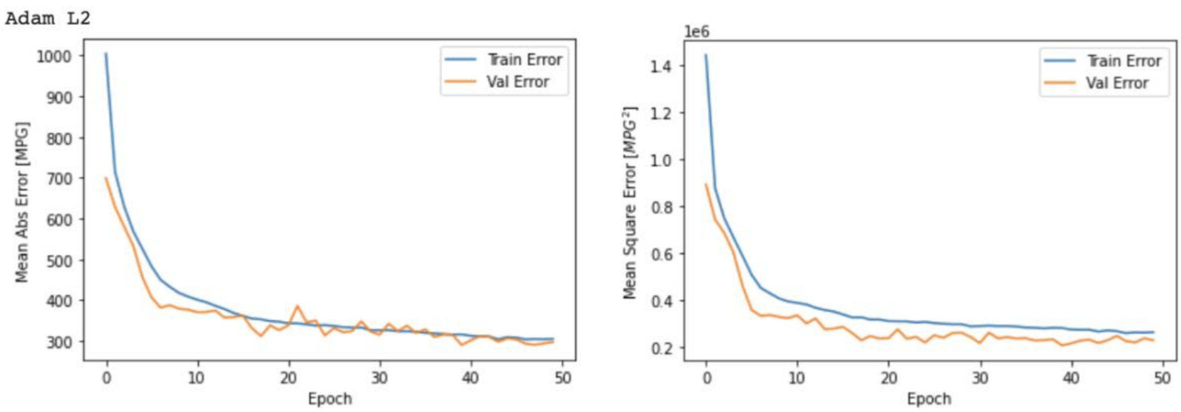

Figure 14 MAE loss (left) and MSE loss (right) of Adam L2 approach

Nadam is similar to an Adam with Nesterov, which is a variant of momentum, The difference between Nesterov and momentum is that Nesterov has an auto correction on direction of gradient by the previous gradient. MAE and MSE loss of Nadam are plotted in Figure 15 with L1 approach and Figure 16 with L2 approach.

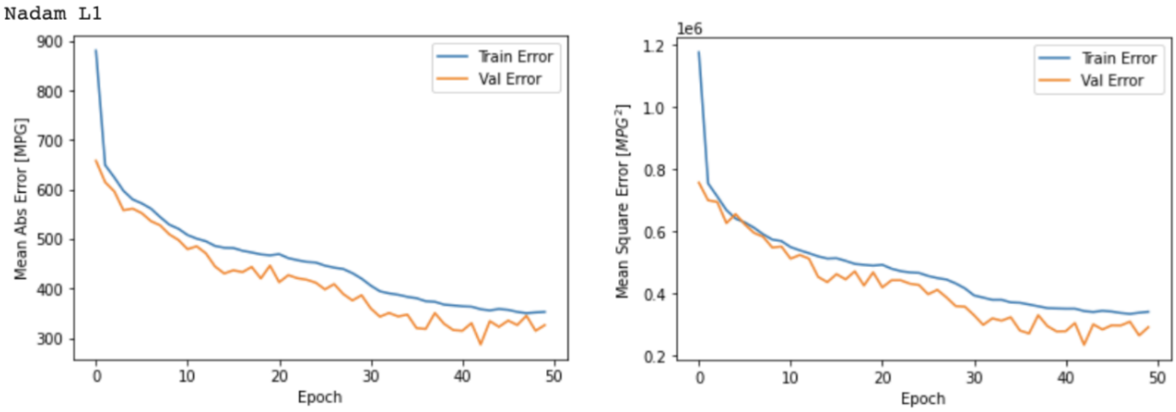

Figure 15 MAE loss (left) and MSE loss (right) of Nadam L1 approach

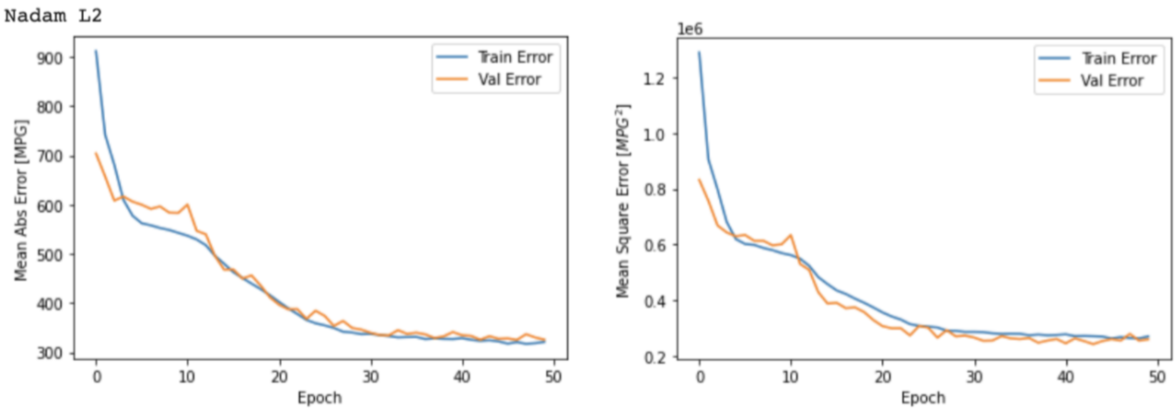

Figure 16 MAE loss (left) and MSE loss (right) of Nadam L2 approach

The four optimization methods SGD, Adam, Nadam and RMSprop have been trained with L1 and L2 norm regularization methods. The comparison of the MAE of prediction on the testing set is shown in Table 5.

On account of the fact that the MAE of prediction on the testing set has the lowest value for Adam with L1 norm regularization. Therefore, Adam is selected as optimization and L1 is selected as regularization method.

	L1 regularization	L2 regularization
SGD	1111.056274	1111.056274
Adam	297.027802	322.621033
Nadam	325.535156	326.184662
RMSprop	263.812134	281.515350

Table 5 Comparison of MAE of prediction on the testing set

3.3 Lifetime Prediction Results

Lifetime prediction results are numeric results and present by light intensity versus performance hour filtered by temperature and stressing current. The results are generated from temperature 273°K to 368°K for every 5°K visualized in 0°C to 95°C, from stressing current 15mA to 105mA for every 5mA for every 1 hour. Stressing current is started from 15mA which represents normal operation current without stress.

Since there are prediction results under numerous conditions, conditions of temperature 15°C, 45°C and 75°C would be taken to show. 15°C is chosen to represent LED runs in a normal cool weather temperature, 45°C is chosen to represents LED runs in a hot weather temperature, and 75°C is chosen to represent LED runs in high temperature environment. Selected results are shown in Figure 17. Under a normal cool weather temperature 15°C operating with no stress, an LED could last for 5000hours. By the increment of stressing current, the performance hour of an LED is shortened. Also, by the increment of temperature, the performance hour of an LED is shortened.

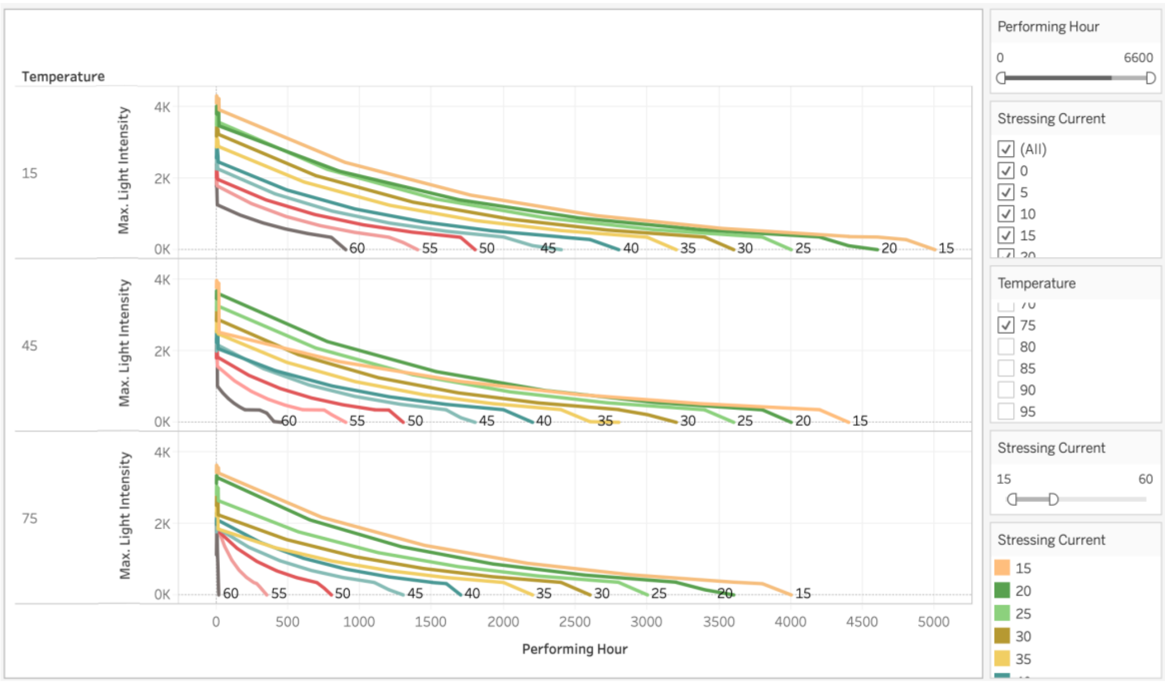

Figure 17 Maximum light intensity vs performance hour filtered by temperature and low stressing current

In Figure 18, it shows the lifetime prediction under the 3 chosen temperature environments and high stressing current between 70mA and 105mA. The lifetime prediction data visualization graphs for high stressing current are more detailed due to the short performance hour.

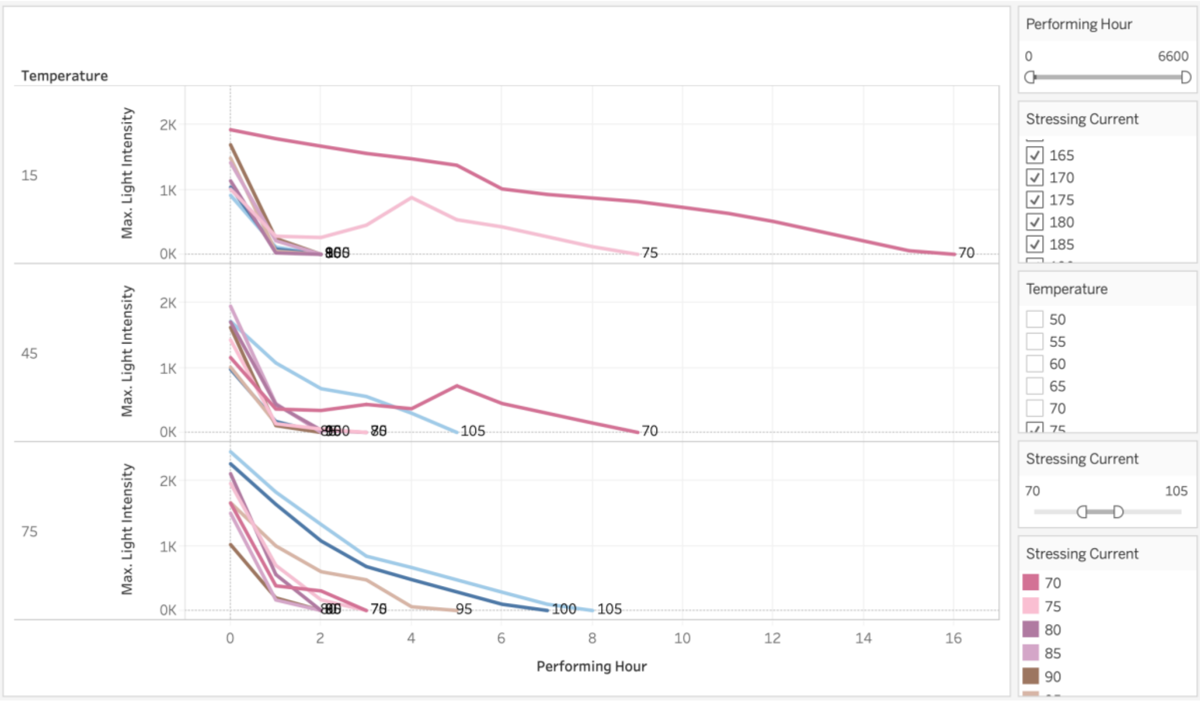

Figure 18 Maximum light intensity vs performance hour filtered by temperature and high stressing current

3.4 Color shift degradation

Figure 19 shows the overview of LED color shift under the 3 chosen temperature environments, 25°C, 45°C and 65°C and filtered by stressing current 15mA, which is the normal operation current without stress, 45mA and 75mA respectively.

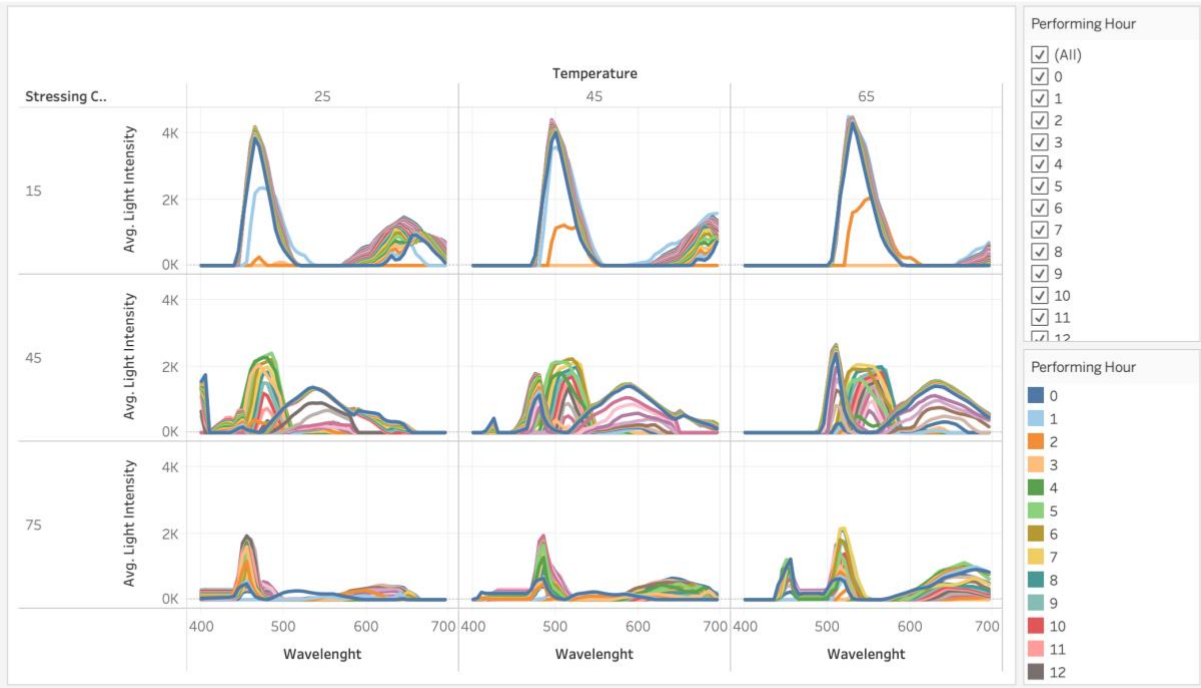

Figure 19 Overview of the color shift degradation

4. Discussion

Overviews of the lifetime prediction results and color shift degradation are demonstrated in the previous part. In this part, the results will be analyzed and discussed.

Due to the limited temperature samples of the original dataset, 25°C and 60°C only, the median of temperature which is 45°C is considered to be relatively more accurate. Therefore, lifetime predictions at 45°C are taken to discuss for both low and high stress current.

In Figure 20, it shows the lifetime prediction 45°C for low stress current. In the beginning, all the light intensity experienced a steep downward drop and followed by a modest decrease steadily. The graph tells that the lower the stress current, the longer the performance hour an LED has. At 45°C, for every 5mA stress current increment, the performance hour is around 500hours shorter.

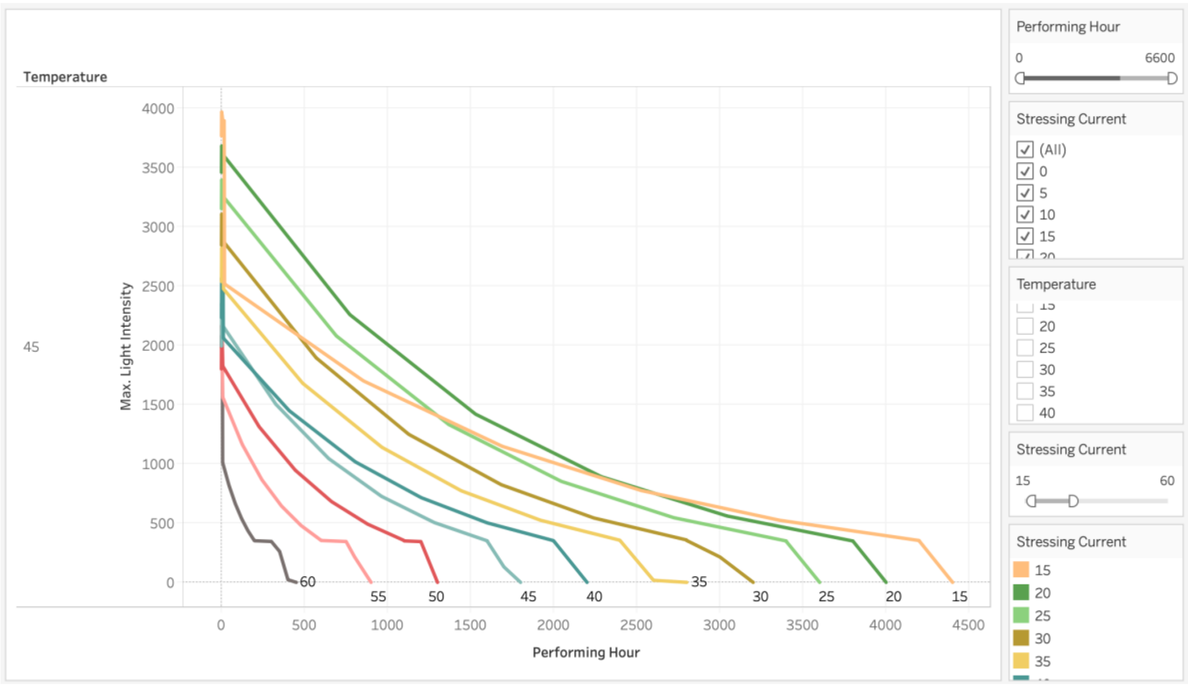

Figure 20 Maximum light intensity vs performance hour at 45°C filtered by low stressing current

In Figure 21, due to the overlap in the graph, only critical lines are being shown. Line of stress current at 65mA starts off with a slight increase in light intensity which is a normal phenomenon for the first operating hour of LED. Stress current at 65mA and 70mA have longer performance hours compared to the others. Lifetime at stress current between 80 and 100 are theoretically reasonable. The worth noticing lines are the higher stress current lines. Theoretically, the higher the stress current, the shorter a LED lifetime should be. However, lifetime at stress current 105mA is longer than stress current between 80 and 100.

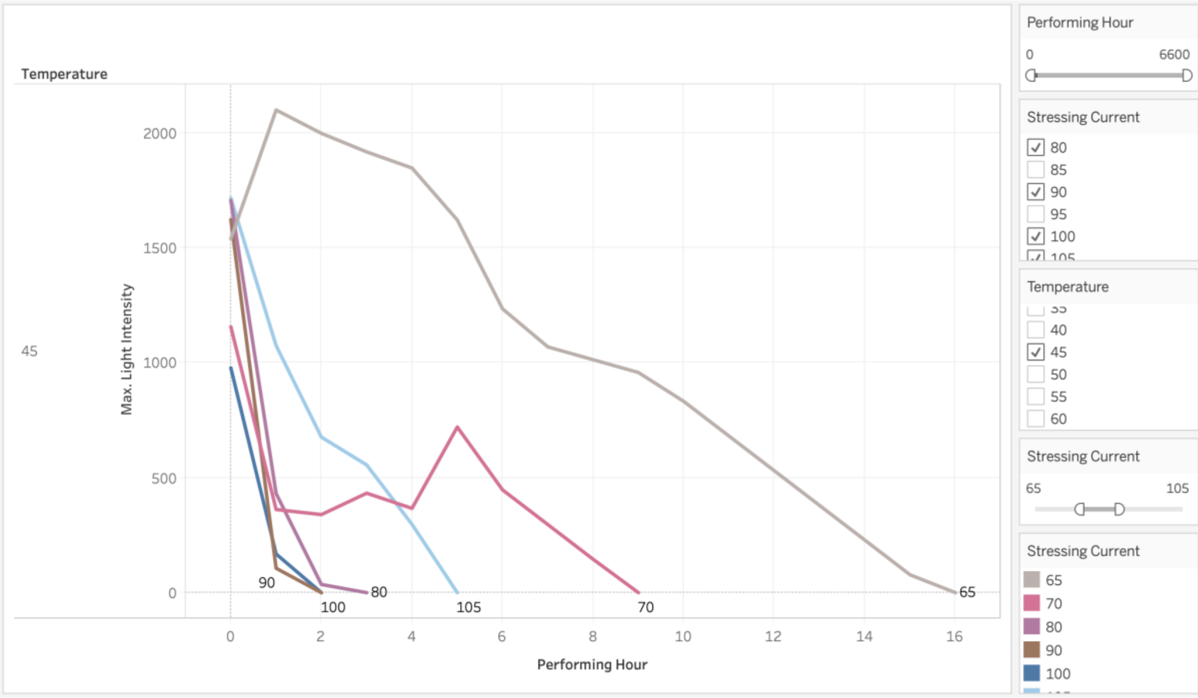

Figure 21 Maximum light intensity vs performance hour at 45°C filtered by high stressing current

The reason for having the above prediction result is explained in the following.

Take the original data as shown in Figure 22 as reference. Firstly, due to the fact that lines of low stress current 40mA and 60mA at both temperature situations last for the longest, this leads to prediction results at stress current 65mA and 70mA last for the longest. And, for lines of stress current 40mA at 25°C and stress current at 60mA at 60°C, both of them experience a slight increase in light intensity. The prediction results also show the pattern of this light intensity increment in the beginning.

Secondly, stress current 100mA at temperature 25°C has higher light intensity compared to stress current 80mA for the whole trajectory. This possibly leads to the phenomenon in the above prediction result that lifetime at stress current 105mA is longer than stress current between 80 and 100.

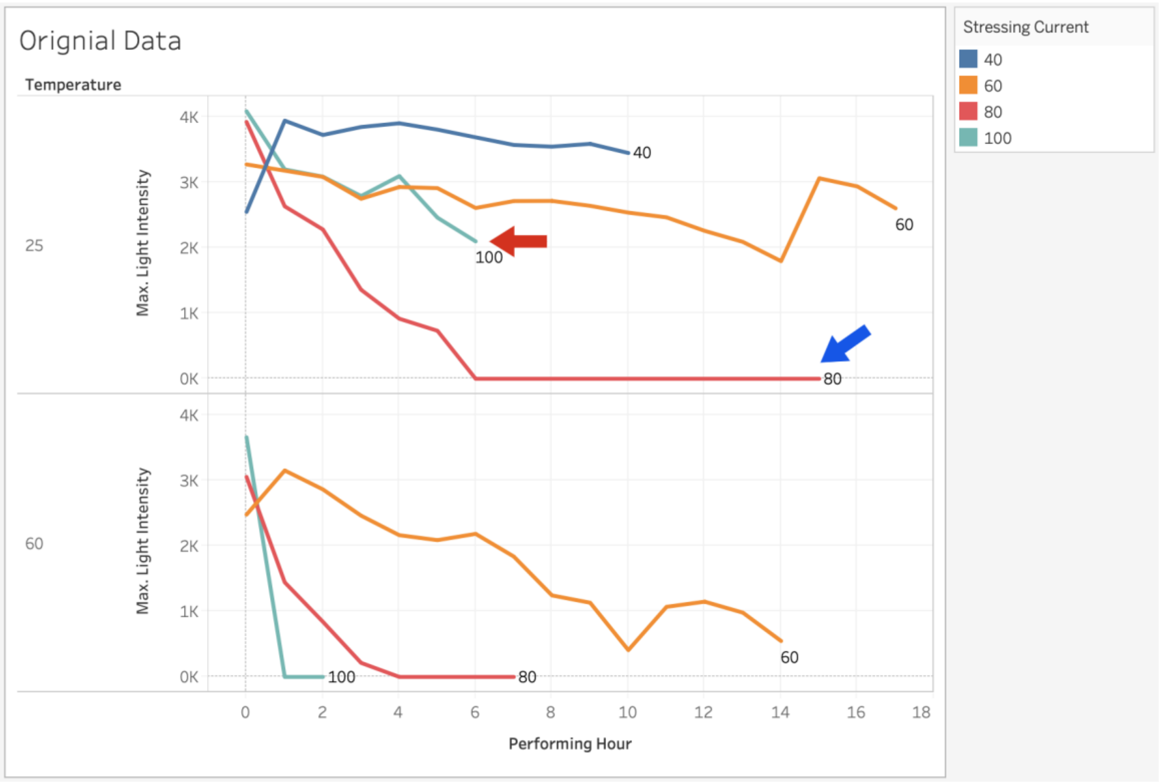

Figure 22 Original dataset visualization for maximum light intensity vs performance hour with arrow marks

In figure 23, lifetime prediction of high stress current between 65mA to 105mA at high temperature 85°C is shown. For the lines of higher stress current, their light intensity dramatically decrease and reach near 0 in the first hour and gradually decrease in the following hour(s).

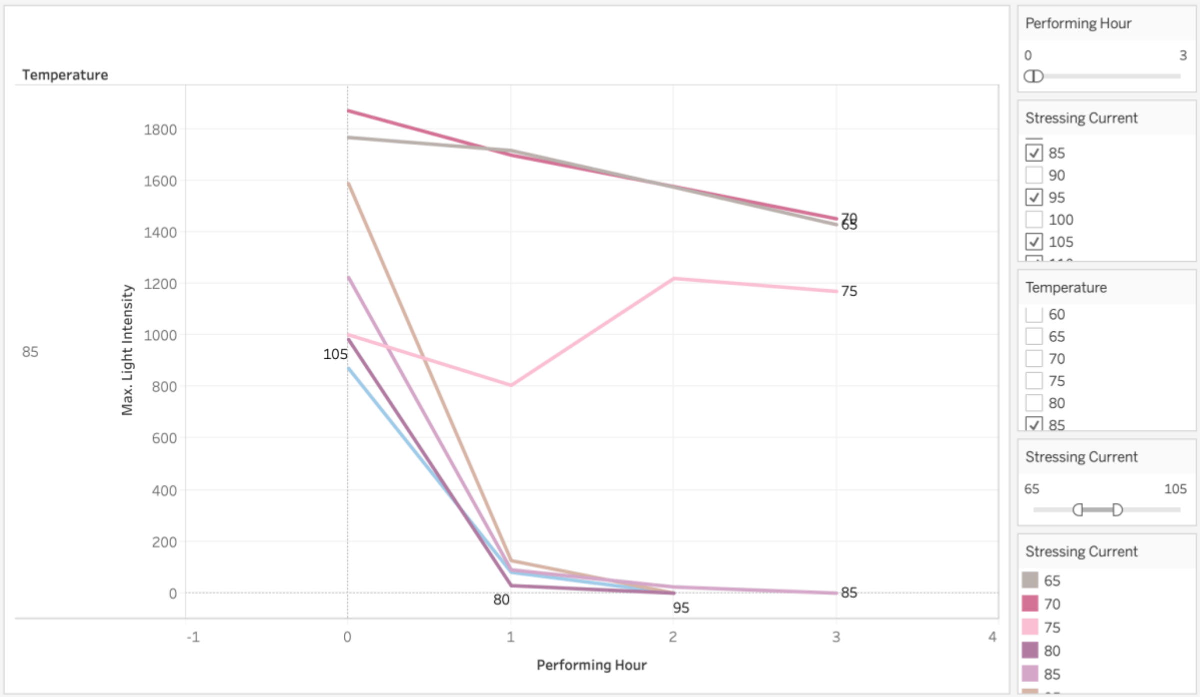

Figure 23 Maximum light intensity vs performance hour at 85°C filtered by high stressing current

In Figure 24, lifetime prediction of high stress current between 65mA to 105mA at high temperature 95°C is shown. The result is very similar to the 85°C mentioned in the previous page. From the observation of both 85°C and 95°C lifetime prediction results, an assumption could be made that this range of temperature is about the maximum temperature that the set of LEDs used in experiments could live with.

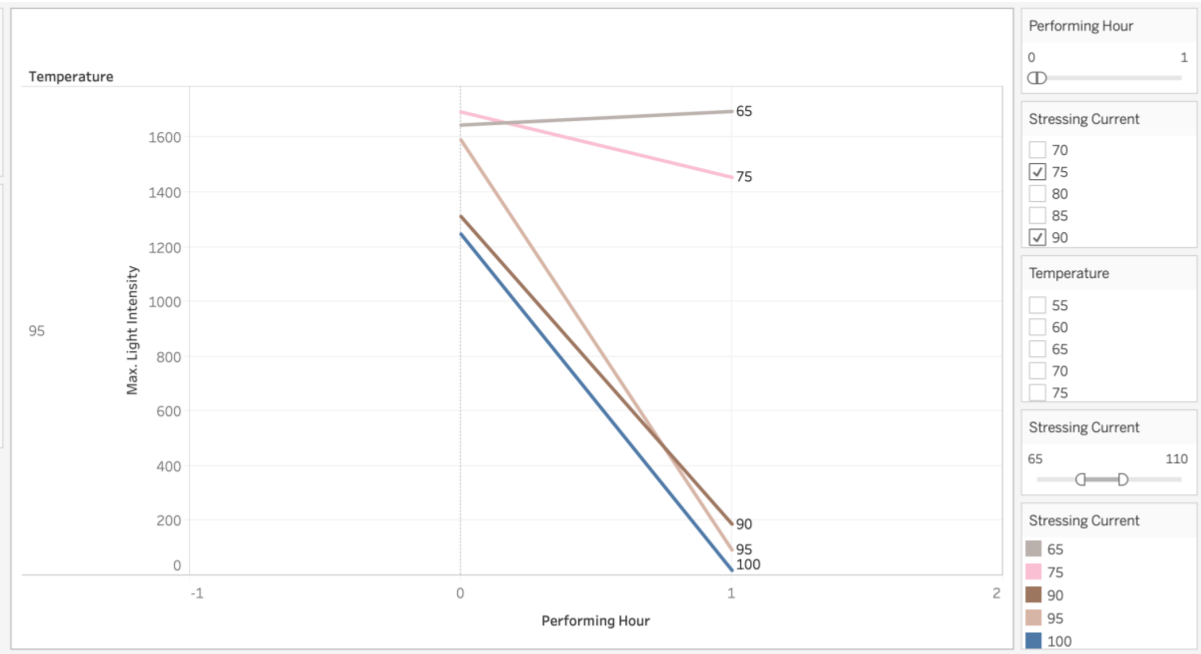

Figure 24 Maximum light intensity vs performance hour at 95°C filtered by high stressing current

For the color shift degradation prediction results to be analyzed and discussed, the original data visualized as spectrum should be discussed first.

In Figure 25, it shows the original data of average light intensity group by LEDs versus wavelength. From what data this project has in original to feed to the deep learning model, is 7 samples in total as shown in Figure 25. The dataset of stress current 40mA at 60°C is missed can be observed. Dataset of stress current 60mA in both temperature situations is most detailed. Dataset starting from stress current 80mA has no much data being recorded.

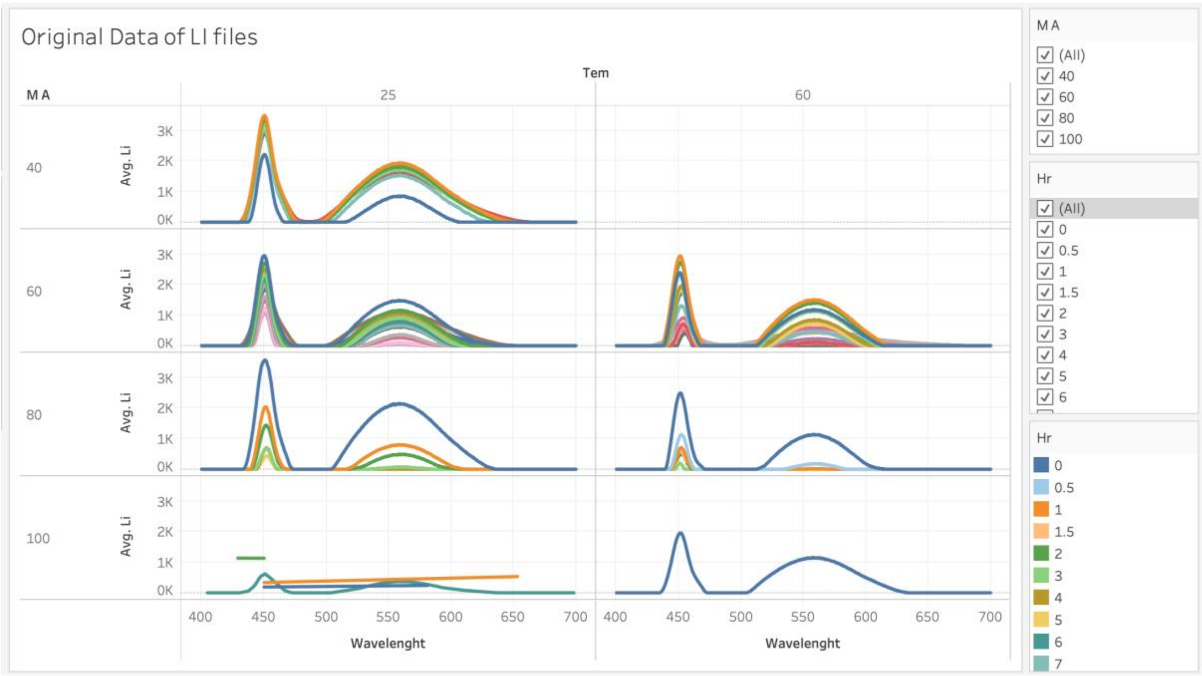

Figure 25 Original data of light intensity vs wavelength

As the reasons mentioned above, the prediction results of color shift would be considerably more accurate and reliable in medium stress current at all temperature situations and low stress current at low temperature situations.

From the results of the color shift spectrum graph, the lifetime of the LED other than the color shift as each line represents the performance hour can be observed. An overview of the color shift degradation is shown in Figure 26. Next, take a closer look into a spectrum in the following page.

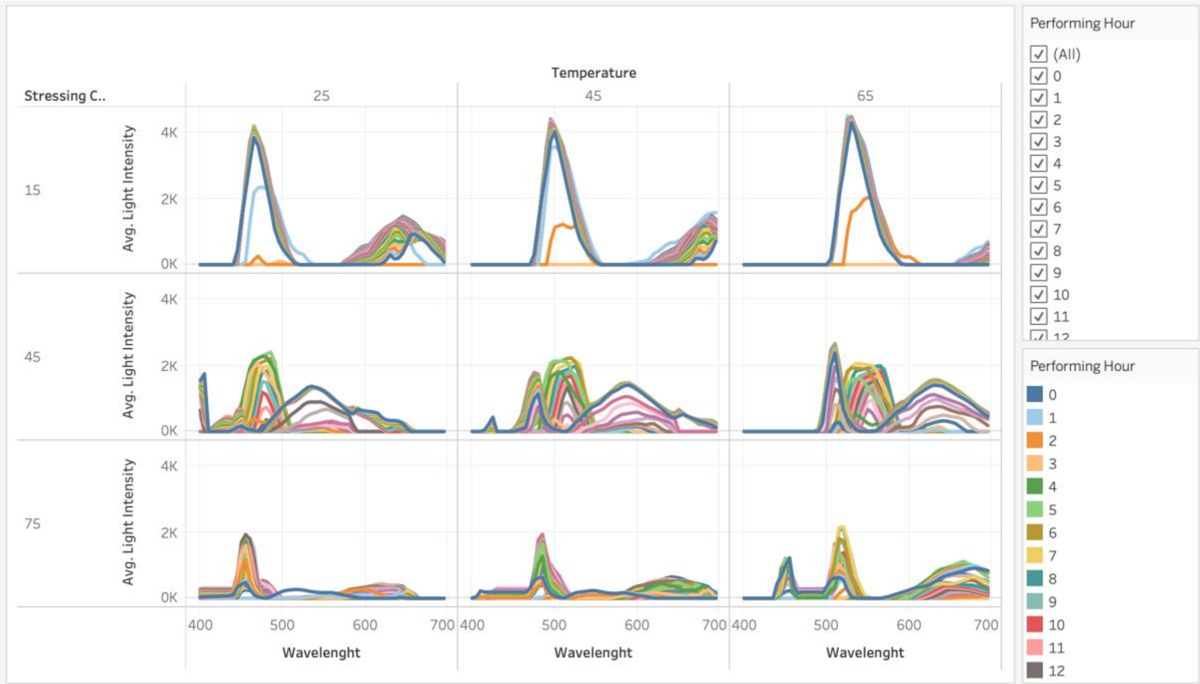

Figure 26 Overview of the color shift degradation

In Figure 27, it shows the color shift prediction in a light spectrum graph of stress current 75mA at 45°C. Each line represents performance hour and the performance hour is listed at the peak value of each line. Considering that showing all lines would result in overlapping, only portions of the lines are selected to show and discuss.

According to figure, maximum light intensity of LED under 75mA stress current at 45°C starts around 480nm at 5th hour and moves to around 515nm at 35th hour. Obeserve that the peak value of each line follows a rightward trajectory according to the increment of performance hour.

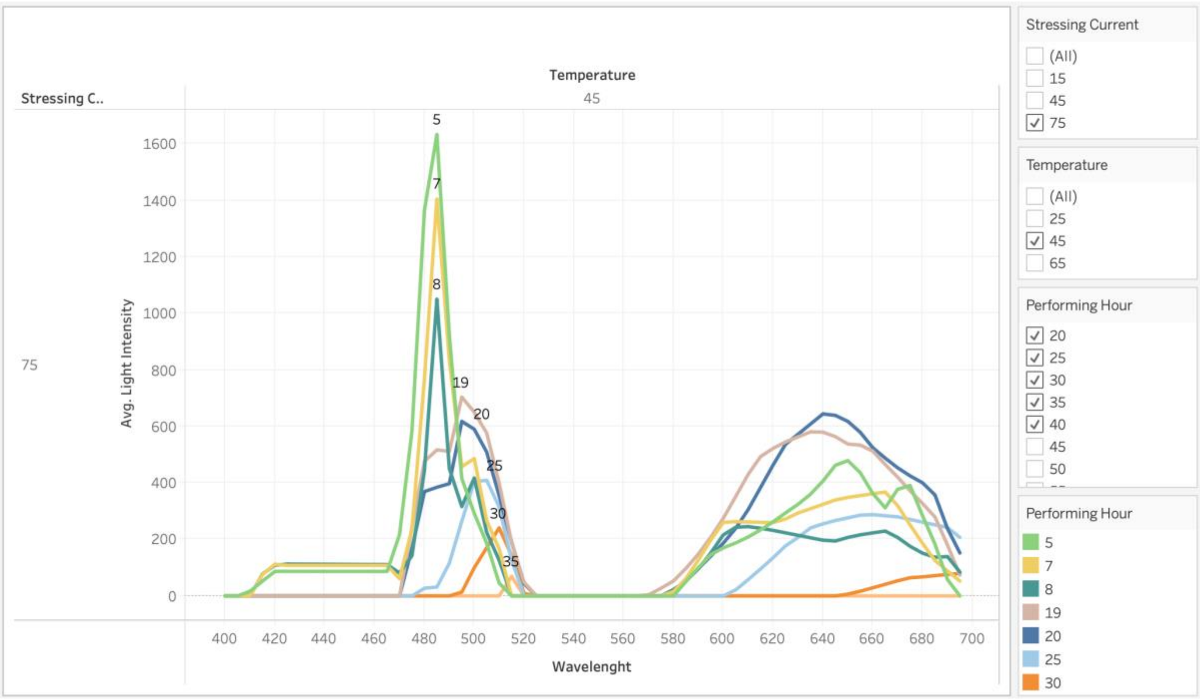

Figure 27 Color shift prediction presented in spectrum of stress current 75mA at 45°C

Other than observing the visible spectrum following the rightward trajectory according to the increment of performance hour, the lifetime in performance hour can also be found in a spectrum.

In Figure 28, it shows lifetime prediction in a light spectrum graph of stress current 75mA at 45°C. Overall maximum light intensity at operation 5th hour is around 1600 and drops below 100 at 35th hour. On the right hand side, performance hour 40, 45 and 50 are being selected, and no data could be visualized. The lifetime of the LED under this situation is around 35hour.

For analysing spectrum graphs, two observations can be found, they are color shift and lifetime prediction.

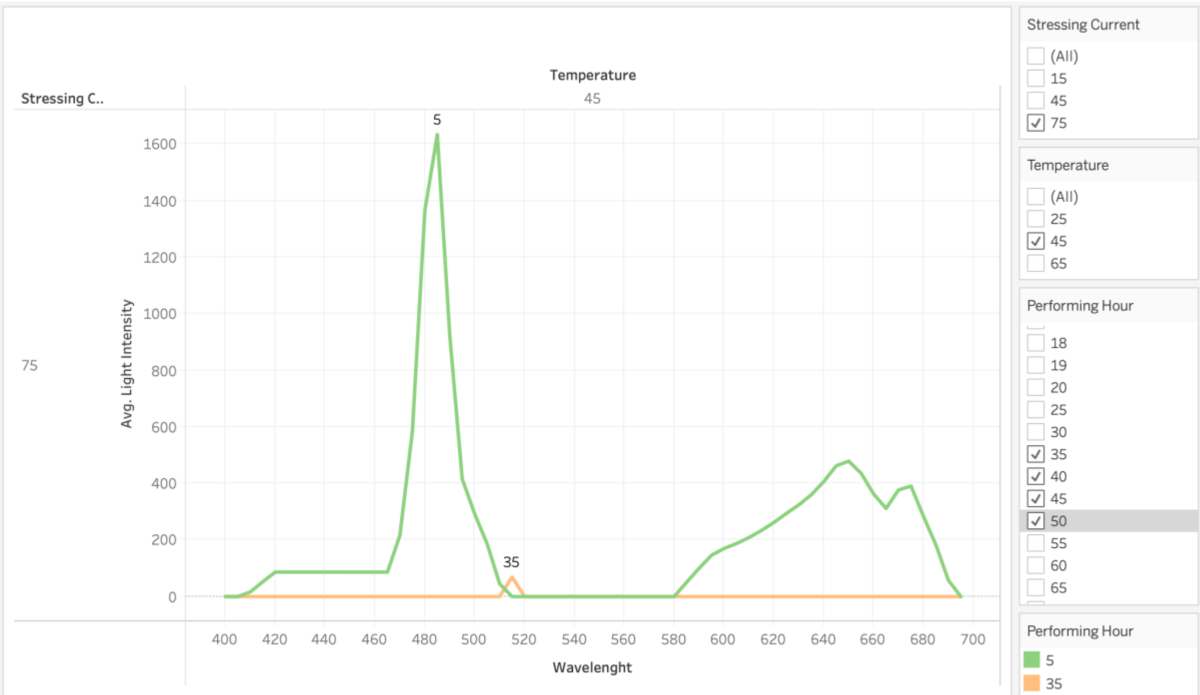

Figure 28 Lifetime prediction presented in spectrum of stress current 75mA at 45°C

5. Conclusion

The developments of AI and deep learning areas provide a promising method for LED lifetime prediction. The main contributions can be summarized as follows: (1) a data-driven DNN approach with RMSprop optimization and L1 norm regularization; (2) lifetime prediction with color shift. The proposed approach is applied to the real dataset of gallium nitride based LEDs life.

For the feedforward process of the DNN model, ReLU is used as activation function and dropout is used as regularization method. For the backpropagation, the deep learning final evaluation result shows the comparison in MAE loss of RMSprop, SGD, Adam, Nadam and compared with L1 and L2 norm regularization. The approach of RMSprop with L1 norm regularization has the lowest MAE loss. As a result, the model is built using RMSprop as optimization and L1 norm as regularization for backpropagation.

The deep learning result shows the lifetime prediction with color shift of LED under different working condition. The prediction result is filtered by working conditions which are temperature and stress current. Other than successfully finding this project main goal which is the lifetime prediction, insights of tolerance in temperature and a rightward color shift in spectrum were also found.

In further improvements, suggestions would be made on trying to work more on feature engineering. Due to the data size is small and some data seem to be with bias in the logging data process in the experiment, it is suggested to manually select and neglect those data which might have bias on affecting output. In this way of improvement, it may be a selection bias. However, it is always a trade-off between bias and accuracy in the deep learning area. The suggested idea is a workable way for further improvement.

6. Appendix

6.1 Table of Abbreviations

Abbreviation	Meaning
DNN	Deep Neural Networks
GBDT	Gradient Boosting Decision Tree
Hr	Hour
LED	Light-Emitting Diode
L1	Lasso Regression
L2	Ridge Regression
mA	milliampere
MAE	Mean Absolute Error
MSE	Mean-Square Error
nm	Newton metre
ReLU	Rectified Linear Unit
RMSprop	Root Mean Square Propagation
SGD	Stochastic Gradient Descent
Tanh	Hyperbolic Tangent

7. Reference

[1] Kübler, R. (2019, December 27). *Introducing Hoeffding's Inequality for creating Storage-less Decision Trees*. Towards Data Science. https://towardsdatascience.com/introducing-hoeffdings-inequality-for-creating-storage-less-decision-trees-b5135e65e51e

[2] Peng, J., Zheng, Z., Zhang, X., Deng, K., Gao, K., Li, H., Chen, B., Yang, Y., & Huang, Z. (2020). A Data-Driven Method with Feature Enhancement and Adaptive Optimization for Lithium-Ion Battery Remaining Useful Life Prediction. *Energies*, *13*(3), 752. https://doi.org/10.3390/en13030752

[3] Hung, C. Y., Chen, W. C., Lai, P. T., Lin, C. H., & Lee, C. C. (2017). Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database. *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 1. https://doi.org/10.1109/embc.2017.8037515

[4] *LED lifetime and the factors that affect it*. (n.d.). Glamox. Retrieved January 15, 2021, from https://glamox.com/gsx/led-lifetime-and-the-factors-that-affect-it

[5] U.S. Department of Energy. (2017, April). *LED Luminaire Reliability: Impact of Color Shift*. Next Generation Lighting Industry Alliance LED Systems Reliability Consortium. https://www.energy.gov/sites/prod/files/2017/04/f34/lsrc_colorshift_apr2017.pdf

[6] Rahman, M. S. (2019). Computations, optimization and tuning of deep feedforward neural networks. https://doi.org/10.1101/2019.12.27.889311

[7] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting Journal of Machine Learning Research. 15: 1929-1958.

[8] Karim, R. (2020, October 5). *Intuitions on L1 and L2 Regularisation - Towards Data Science*. Medium. https://towardsdatascience.com/intuitions-on-l1-and-l2-regularisation-235f2db4c261

[9] Zou, F., Shen, L., Jie, Z., Zhang, W., & Liu, W. (2019). A Sufficient Condition for Convergences of Adam and RMSProp. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1. https://doi.org/10.1109/cvpr.2019.01138

[10] Brownlee, J. (2021, February 15). *Regression Metrics for Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/regression-metrics-for-machine-learning/