Chapter 4: Methodology

4.1 Chapter Introduction

This chapter is focused on much more exploration on the methodology of this research by explaining how the research data is processed, prepared and how the ANN allows achieving the research goal. This methodology includes different areas of the research like the method of collecting datasets from the web, data pre-processing, approaches and methods of the analysis, data visualization, applying machine learning, and deep learning model.

The research methodology is divided into three parts for better understanding,

\* Data collection and pre-processing

\* Data analysis and visualization

\* Model building and evaluation

4.2 Data Collection and Pre-processing

### 4.2.1 Data Processing Frameworks and Libraries

#### Pandas

Pandas is an embedded analytic package for the Python. It supports multiple operation for changing needs data, with a focus on expanded linear (McKinney et al.,2011). Pandas is an embedded analytical package for the Python programming language. It supports multiple techniques for changing needs data, with a focus on expanded linear.

Pandas is a Python library that provides a set of tools to do data analysis. Pandas is used for loading dataset, preparing, manipulating, reshaping and analyzing data. It can join and merge datasets to features. It has also statistical properties that allow gaining insight into the dataset. It has a data type structure called a data frame. Pandas have been strongly used for this project for processing and analyzing the bearing RUL dataset.

NumPy

NumPy, which represents Numerical Python, is a package that contains multidimensional array structures and also a set of methods for manipulating such arrays.

Numpy deals with the multidimensional array, for its multidimensional properties the computation time takes less time than the list or dictionary. It also can take less space in memory for computation. It can execute the statistical operation, boolean, arithmetical computation. The whole dataset in this project has transformed into a NumPy multi-dimensional array for feeding the Artificial Neural Network.

Matplotlib

Matplotlib is a plotting library and object-oriented API for embedding plots in python language. It is widely used in graphical content for analysis data. It has statistical plotting propertied like histogram, barplot, box plot, etc.

Matplotlib can take one or two-dimensional arrays for plotting. It has lots of parameters to enhancing the graph quality.

Seaborn

Seaborn is a plotting and graphical package built on Matplotlib. It has some attributes and graphical properties that make it a more powerful plotting package. It provides a high-level interface for drawing attractive and informative statistical graphics. It has statistical plotting propertied like histogram, barplot, distribution plot, box plot, violin plot, line plot, pair grid, heatmap, etc. All the plotting of this project has been used in seaborn for better visualization.

Tensorflow

TensorFlow is an open-source end-to-end platform for machine learning and deep learning. It provides a comprehensive ecosystem of tools for building scalable machine learning powered applications. TensorFlow was designed to build models easily with an intuitive easy to use set of APIs that makes it simple for implementing machine learning, deep learning, and scientific computing. It takes data as a multidimensional array and computes it with multidimensional matrix form. It provides a rich collection of tools for building models including data pre-processing, data ingestion, model evaluation, visualization, and serving. But it's not just for building models. It's designed to be highly portable, running on a variety of devices and platforms. It can scale from a single CPU to a GPU or cluster of GPUs, all the way up to a multi-node TPU infrastructure. So it's simple to build models that work on any platform taking advantage of the power of each. TensorFlow also allows for powerful experimentation with the flexibility to quickly implement machine learning and deep learning models. In the project, TensorFlow has been used for implementing the Artificial Neural Network and its evaluation.

Keras

Keras is an open-source software library that acts as an interface for the TensorFlow library. Keras is an interface that can easily access and customize the machine learning frameworks including TensorFlow, Microsoft Cognitive Toolkit or CNTK, and Theano. These frameworks, also known as backends. It is a high-level API application programming interface implementing machine learning, and specifically, deep neural networks. Keras allows for quick experimentation with deep neural networks and its computational speed is very fast.

4.2.2 Data Collection

The dataset for this project has been downloaded from the FEMTO Bearing RUL dataset. The model has been built on this dataset. But before this work, the dataset has been modified for the project because the dataset was built under many experiments on the rolling bearing remaining useful lifecycle. It has 6 features which are hour, minute, second, micro-second, horizontal acceleration, and vertical acceleration. Horizontal acceleration and vertical acceleration are a feature of the vibration signal. The prediction of the remaining useful life cycle of bearing is based on vibration signal. Some operations have been applied with Microsoft excel for merging the datasets; transforming hour, minute, the second feature in Seconds; adding all times in Total\_Sec feature. The Total\_Sec columns have 12 unique values. The dataset contains 30719 rows and 3 columns after merging the sample datasets with Microsoft Excel.

4.2.3 Importing the Necessary Libraries

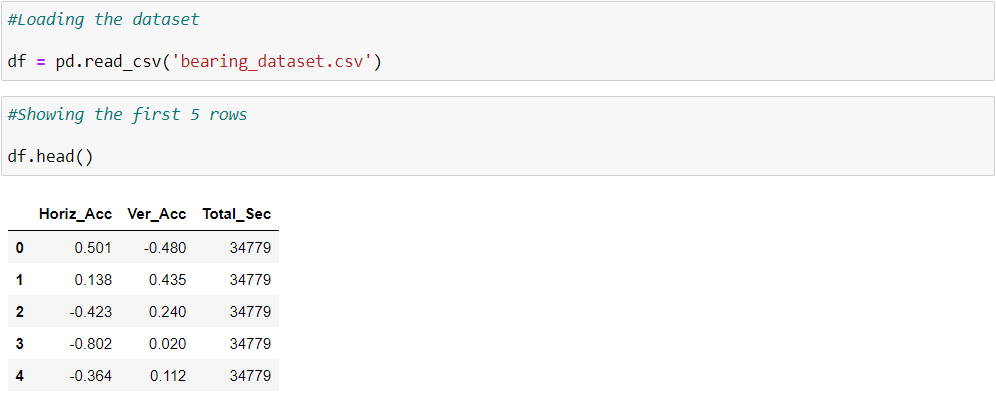
Numpy, Pandas, Matplotlib, Seaborn, TensorFlow, and Keras have been imported for processing, analyzing, visualizing, model building, and evaluating. Numpy is for mathematical computation with the multidimensional array, Pandas is for data analysis and processing. Matplotlib and Seaborn were used for data visualization. TensorFlow and Keras have been used for building ANN and evaluating the model prediction on the test dataset.



***Figure 4.1: Importing necessary libraries***

4.2.4 Loading the dataset

The dataset has been loaded with a pandas module called read\_csv(). The module takes a comma-separated values (CSV) type file. The file contains the values one after one and is divided with a comma symbol.



***Figure 4.2: Load the dataset***

4.2.5 Splitting the Dataset

The dataset has been split into two parts and it is stored in X and y variables. X contains the ‘Horiz\_Acc’ and ‘Ver\_Acc’ features, y contains the ‘Total\_Sec’ feature. Both variables are stored in the NumPy array. X is a 2-dimensional array and y is a 1-dimensional array.



***Figure 4.3: Split the dataset***

X store the value point in this way,

array([[ 0.501 , -0.48 ],

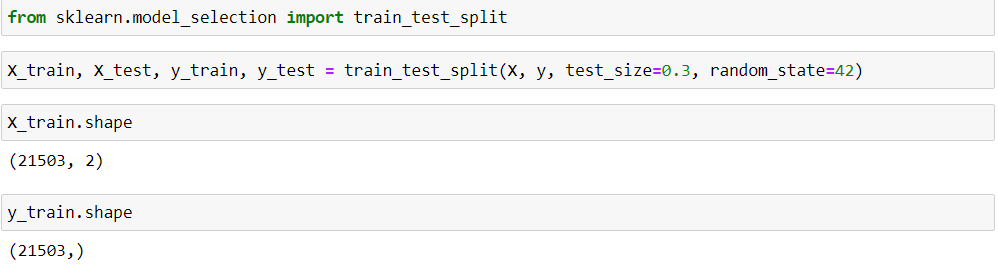
[ 0.138 , 0.435 ],

[-0.423 , 0.24 ]])

And y store the value point in this way,

array([34779, 34779, 34779, ..., 29665, 29665, 29665], dtype=int64)

A python machine learning package called Scikit-learn has been declared for importing train\_test\_split. train\_test\_split and class attribute which divides the dataset into train and test form. In this project, The test dataset has been split into 30% and the train dataset 70% of the whole dataset into X\_train and y\_train.

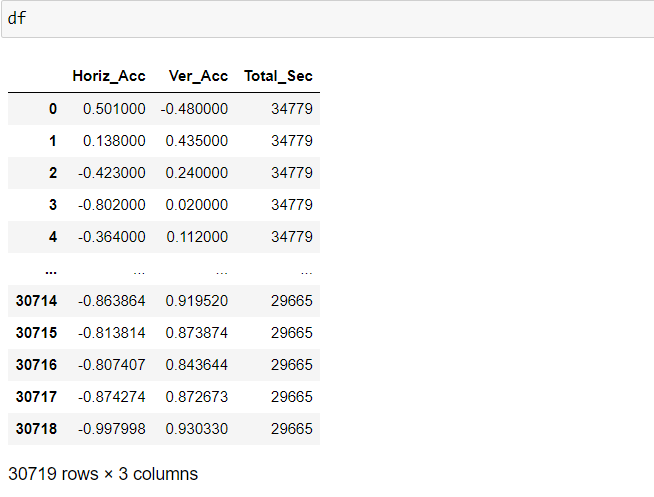


***Figure 4.4: Splitting the dataset into train and test set***

4.3 Data Analysis and Visualization

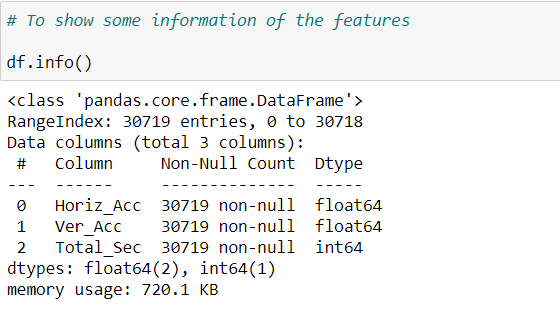
4.3.1 Exploring the dataset

After loading the dataset into a variable called ‘df’. The df variable contains the whole dataset in pandas dataframe. It shows the first and last five rows of the dataset.



**Figure 4.5: Showing the dataframe**

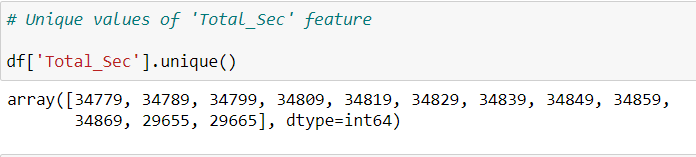
Pandas have a function called info() which shows the number of rows of each feature column. The dataset has 30719 non-null values. ‘Horiz\_Acc’ and ‘Ver\_Acc’ contain float64 type data and ‘Total\_Sec’ has int64 data type. The range index I 0 to 30718.



***Figure 4.6: To show some information of the dataframe***

The label feature of this dataset is the ‘Total\_Sec’ column. So, the ‘Total\_Sec’ has been seen deeply to know its properties. It has 12 unique values. For having 12 unique values, it would be a classification type model.

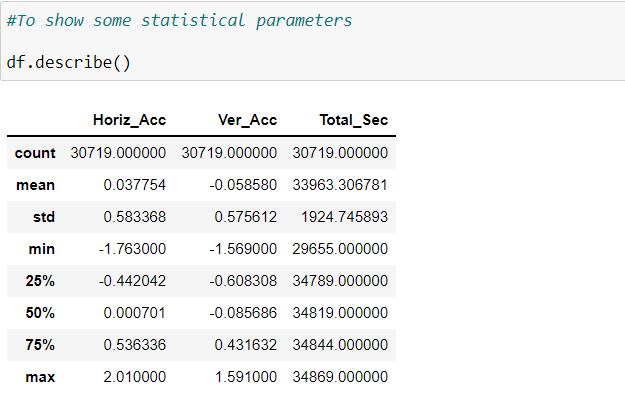
They are 34779, 34789, 34799, 34809, 34819, 34829, 34839, 34849, 34859, 34869, 29655 and 29665.



***Figure 4.7: Unique values of 'Total\_Sec' feature***

4.3.2 Statistical Data Analysis

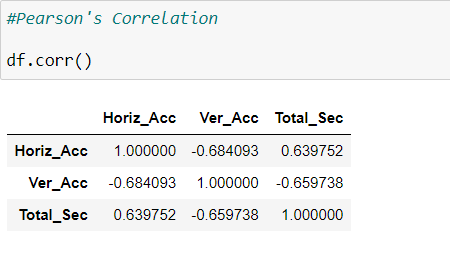
In data analysis, statistical methods and operations are used for gaining the overall aspect and knowledge of the given dataset. Pandas can execute many statistical operations. Pandas have a method called describe() that gives a brief statistical computation on some parameters. They are counts, mean, standard deviation, minimum, 25% percentile, 50% percentile, 75% percentile, and maximum. The ‘Horiz\_Acc’, ‘Ver\_Acc’, ‘Total\_Acc’ features statistical operations have been displayed below.



***Figure 4.8: To show some statistical parameters***

**Pearson’s correlation coefficient** is the statistical method that determined the statistical relationship, association, between two continuous variables.  It is the best method of measuring the relationship between variables of two features because it is based on the method of covariance.  It gives the knowledge about the magnitude of the association, correlation as well as the direction of the relationship. In this project, Pearson’s correlation is used for determining the relationships of features and how the features are correlated with each other.

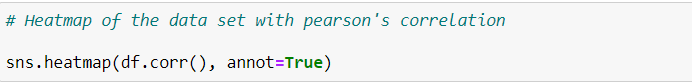
Here the Pearson’s Correlation has been shown by pandas corr() function.



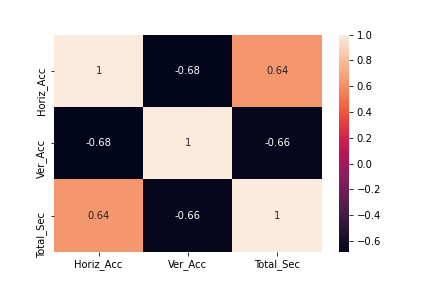
***Figure 4.9: Pearson's Correlation***

Its range is between 1 to -1. If two variables have the correction value of 0 that means those variables have no relation. If this variable value is near 0, it means the correlation between them is low. And if the value of the variable is near 1 or -1, it means the variables are highly correlated.

A better way to visualize the correlation is with the help of Seaborn’s heatmap() object function. It allows us to see the correlation with the graph. It has a parameter called annot which allows the Pearson’s correlation coefficient into the colored boxes.



***Figure 4.10: Heatmap of the data set with Pearson's correlation***



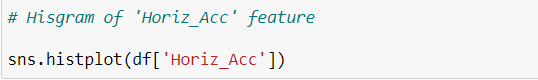
***Figure 4.11: Heat map of Horiz\_Acc, Ver\_Acc, and Total\_Sec feature***

4.3.3 Data Visualization

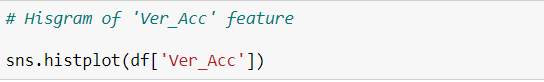
Data visualization has been performed over the dataset to gain insight into the structure of data. It was based on the two main questions: how the features are correlated and whether the features are linearly separable or not.

The histogram is a frequency distribution that shows how the data points of different value in a set of data occurs. The histogram is a very useful graph or plot to see the frequency distributions of the data points. It shows the concentration of the frequency data with counts.

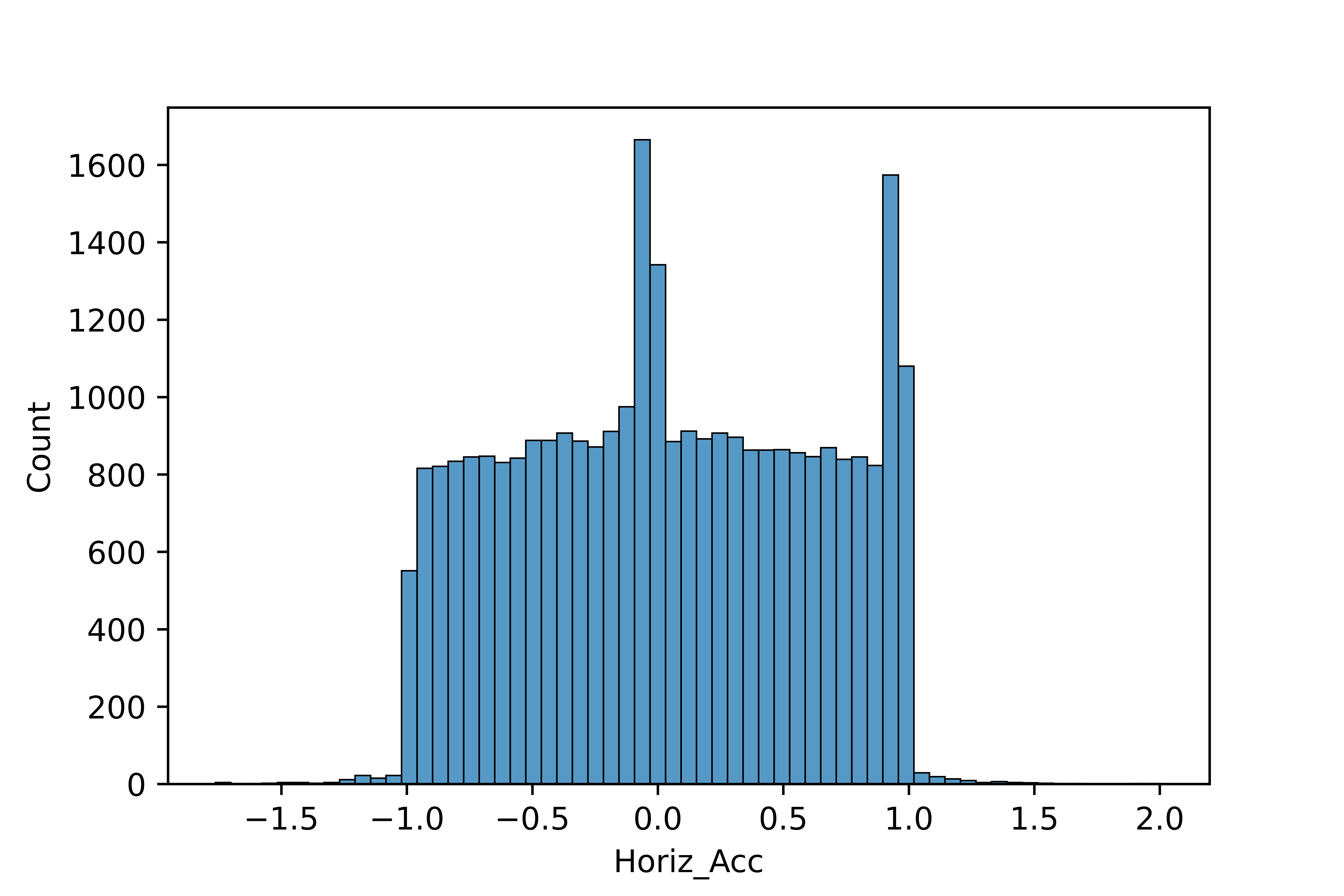
The first histogram refers to the ‘Horize\_Acc’ feature. The maximum value of this feature is between -1 to 1. It has a high density near 0. It has the highest count nearly 1600 and the average count between 800 to 1000.



***Figure 4.12: Histogram of 'Horiz\_Acc' feature***

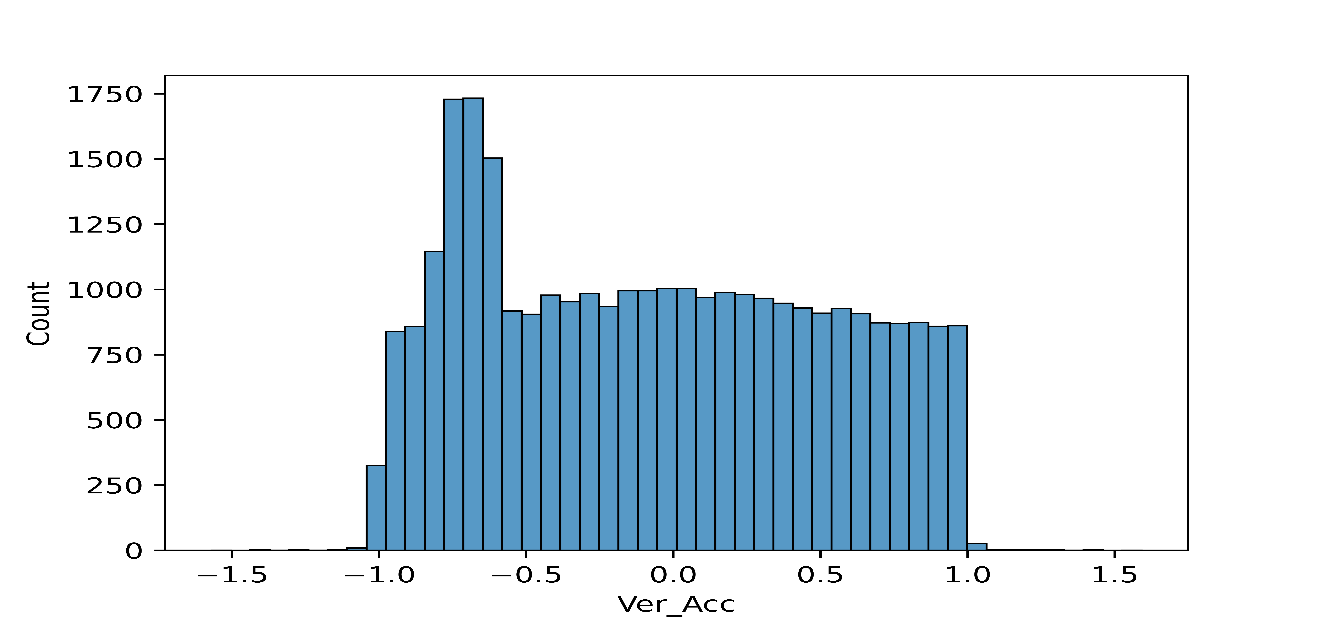


***Figure 4.13: Histogram of 'Ver\_Acc' feature***



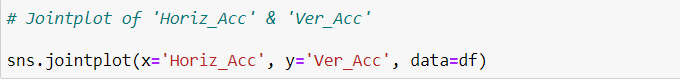
***Figure 4.14: Histogram of Horiz\_Acc feature***

The second graph is the histogram of the ‘Ver\_Acc’ feature. The maximum value of this feature is between -1 to 1. It has a high density near 0. It has the highest count nearly 1750 and the average counts between 750 to 1000.

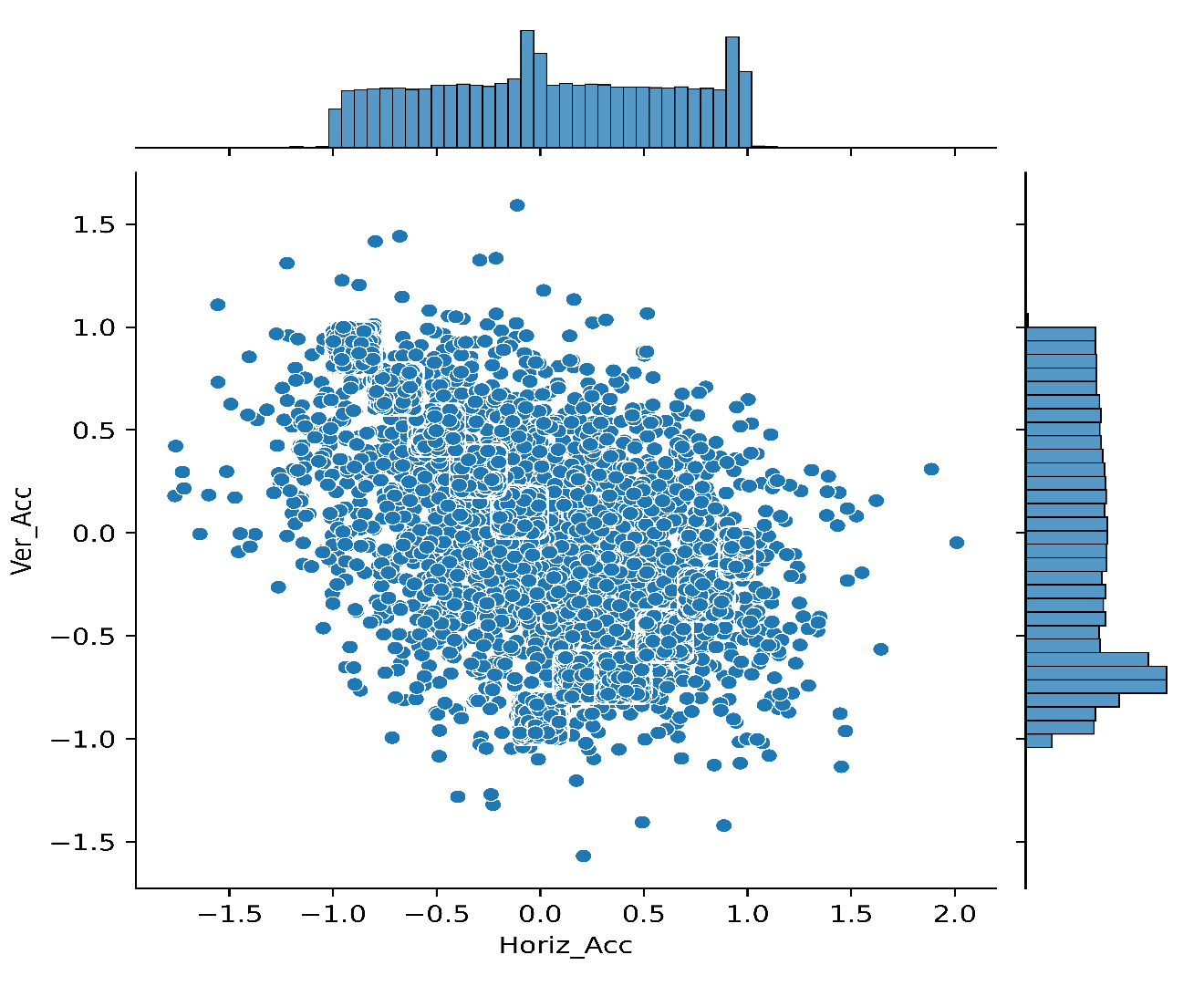


***Figure 4.15: Histogram of Ver\_Acc feature***

The most important data visualization of this project is the relation between the ‘Horiz\_Acc’ and ‘Ver\_Acc’ features. The prediction of the remaining useful life of bearing was based on vibration signals which are horizontal acceleration and vertical acceleration. The graph of the ‘Horiz\_Acc’ and ‘Ver\_Acc’ shows the correlation and the density of the value points. The average, standard deviation, kernel density are visible in the following graph. The graph has been plotted with seaborn’s plotting function called jointplot().



***Figure 4.16: Jointplot of 'Horiz\_Acc' & 'Ver\_Acc'***



***Figure 4.17: Relation of Horiz\_Acc and Ver\_Acc feature***

4.3 Model Building and Evaluation

In this research, a simple Artificial Neural Network was used through a Keras Sequential Model, which ran on top of the Keras API which was imported from TensorFlow. The model was used loss as Sparse Categorical Crossentropy, optimizer as Stochastic Gradient Descent, metrics as Accuracy

4.3.1 Artificial Neural Network

Artificial Neural Networks (ANN) is one of the sub-branches of Artificial Intelligence (AI). As IBM defines, ANN is a set of supervised algorithms that build the framework of Deep Learning (DL), a subset of Machine Learning (ML), which is again a subset of Artificial Intelligence (AI) (IBM, 2020).

The Artificial Neural Network was built on the base of the human brain. The computer of the machine learns from experience by mimicking the human brain. The Neural word came from the human brain’s neurons. Human brain neurons connect and pass the signal from one neuron to the next neuron. The computer does such things as passing the electric signal from each layer to other layers.



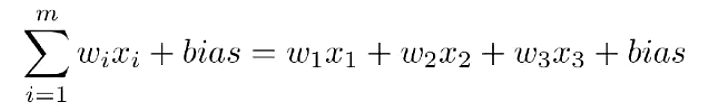
*Figure 4.18: Rudimentary ANN model structure*

An Artificial Neural Network has three factors for processing the overall system, that are-

* Input(s)
* Weights, and
* Output(s)

The input is like human eyes, ears, tongue, etc. organs, that sense something from the environment and pass the signal to the brain. The computer does the same as this. Electric signals go into the computer’s processor as input. Weights are like a human brain processing system where the brain decides and process what to do and how to do it. The computer's processor calculates the weights with input, changes the weights according to input and desire output. The weight changes based on accuracy which means how much the computer detects accurately. The last factor is output. In humans, the output is the act of doing something with the body according to the brain’s decision. The computer’s output is the desired result, prediction, and learning from experience.

Equation (2) implies the mathematical relation between these parameters.



*Equation1: Mathematical representation of the relations between the inputs, the weights, and the outputs*

According to how these layers function, they can be categorized into three distinct layers, namely:

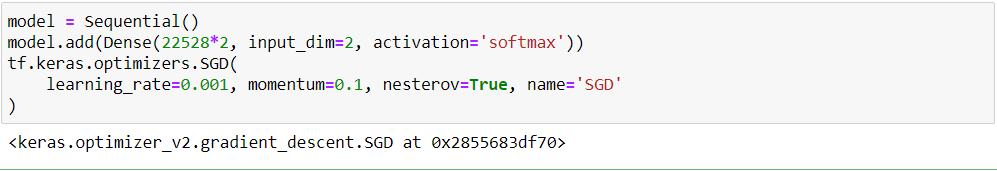
* The Input layer
* The Hidden Layer(s)
* The Output Layer

The Input layer grabs the input signal or data, It can be used in various shapes. Most of the time, the input layer takes a 1-dimensional array. The Hidden layer is the most important part of the Artificial Neural Network. It calculates the input data according to the desired output. It calculates with weights, bias, and some mathematical operation line chain rule of derivative. The hidden layer can have many layers inside it. Some layers process some activation functions line relu, sigmoid, liner, gelu, etc. The output layer shows the predictions in probability form.

### 4.3.2 Keras Sequential Model

Keras, a Sequential model is an ideal implementation for cases where every layer has a unit input or output tensor (Keras, 2020).

Keras sequential model is used for building the model with a sequence form. This means each layer in a sequential model contains a vector or a matrix of n-dimension comprising input or output data. It has a layer with input features and input shape, dense, and dropout. A Keras sequential model has been used for this research.



***Figure 4.19: Keras Model***

4.3.3 Stochastic Gradient Descent (SGD)

Stochastic gradient descent is an optimizing function with the iterative property of [differentiable](https://en.wikipedia.org/wiki/Differentiable_function) or [sub-differentiable](https://en.wikipedia.org/wiki/Subgradient_method) objects. It is one kind of gradient descent but it only takes one data randomly and then calculating the data with weight and bias. After calculating with weight and bias, it finds the cost function and error rate. Then it does forward pass and adjust weight and bias.

In this research, the Stochastic gradient descent has been used for getting a good prediction accuracy. Sparse categorical crossentropy has been used as a loss because the label feature contains 12 categorical values. Softmax activation has been used because of its smoothness in calculating speed with SGD.

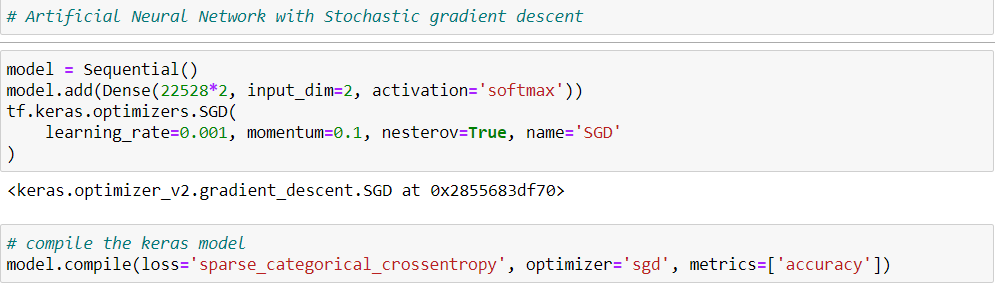
The model.add() is a keras module for adding a layer. The input layer has taken 22528\*2 = 45056 data as input. Because the X\_train variable has 22528 rows, which contains a total of 45056 values. Input dimension is 2 because of the shape of X\_train.

tf.keras.optimizers.SGD(

learning\_rate=0.001, momentum=0.1, nesterov=True, name='SGD'

)

The learning rate of the gradient descent optimizer is set at 0.001. the momentum is 0.1



***Figure 4.20: Artificial Neural Network with Stochastic gradient descent***

4.3.4 Model Evaluation

After building and training the model, it was evaluated based on several crucial parameters, which included:

* **The Accuracy of the model**
* **Loss**
* **Precision**
* **Recall**
* **F1- Score**
* **Confusion Matrix**
* **Accuracy Vs. Epoch Chart**
* **Loss Vs. Epoch Chart**

### 4.3.4.1 Model Accuracy

Model accuracy is the measure of how well the model predicted the target labels, usually expressed as a percentage. More precisely, it is the ratio of correct predictions to total predictions and can be mathematically expressed as below:

**Accuracy =**

*Equation2: Model Accuracy Formula*

### 4.3.4.2 Loss

Loss is the measure of the model's prediction error, i.e., a quantitative value of the difference between the predicted output and the actual output.

### 4.3.4.3 Precision

Precision is the ratio of the number of true positives to the sum of the number of true and false positives (Koehrsen, 2021).

**Precision =**

*Equation 3: Model Precision Formula*

### 4.3.4.4 Recall

The recall is the ratio of the number of true positives to the sum of the number of true positives and false negatives (Koehrsen, 2021).

**Recall =**

*Equation 4: Model Recall Formula*

### 4.3.4.5 F1-Score

F1-Score, also known as F – Score, is a collective measure of the model's performance that considers model precision and recall. The value can be expressed by the mathematical formula below (Wood, n.d.):

**F-1 Score =**

*Equation 6: F1-Score Formula*

### 4.3.4.6 Confusion Matrix

Confusion Matrix is a type of matrix of 2d array which shows a summary of how well the model performed in terms of accuracy, number of current predictions, and the number of incorrect predictions.

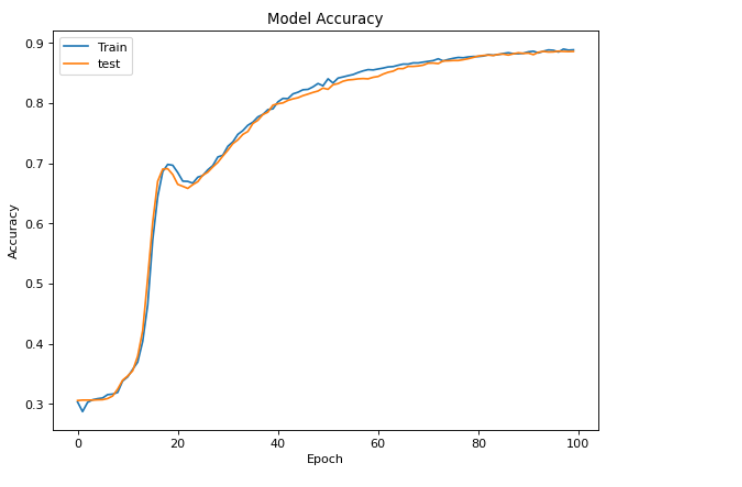
It has four factors: True Positives (TP), True Negatives (TN), False Positive (FP), and False Negative (FN) (Mohajon, 2020).

Chapter 5: Result Analysis

The previous chapter was covered the research methodology about how the data processing, analyzing, visualizing, building, and evaluating model was executed. In this chapter, the model evaluation method results have been described.

5.1 Accuracy

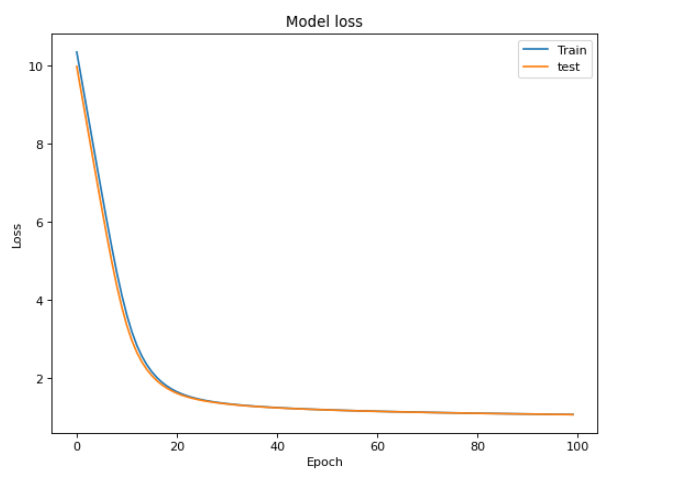
Accuracy is the simplest way for scoring a model. It is the ratio of the number of correct predictions and the total number of predictions. In this project, the ANN model has achieved 89.88% accuracy on the train set and 89.68 on the test set.



***Figure 5.1: Accuracy vs Epoch graph***

5.2 Loss

Sparse categorical crossentropy has been used in this model as loss. Here is the loss vs. epoch graph.



***Figure 5.2: Loss vs Epoch graph***

5.3 Classification Report

Classification report is a chart that contains precision, recall, f1-score, and support. The classification report has been imported from the scikit-learn package.

precision recall f1-score support

29655 0.95 0.76 0.84 792

29665 0.77 1.00 0.87 755

34779 0.85 0.22 0.35 755

34789 0.89 1.00 0.94 782

34799 0.94 0.90 0.92 757

34809 0.93 0.98 0.95 779

34819 0.95 0.99 0.97 797

34829 0.95 0.90 0.92 802

34839 0.83 1.00 0.90 744

34849 0.82 0.96 0.88 717

34859 0.89 0.96 0.92 749

34869 0.93 0.99 0.96 787

accuracy 0.89 9216

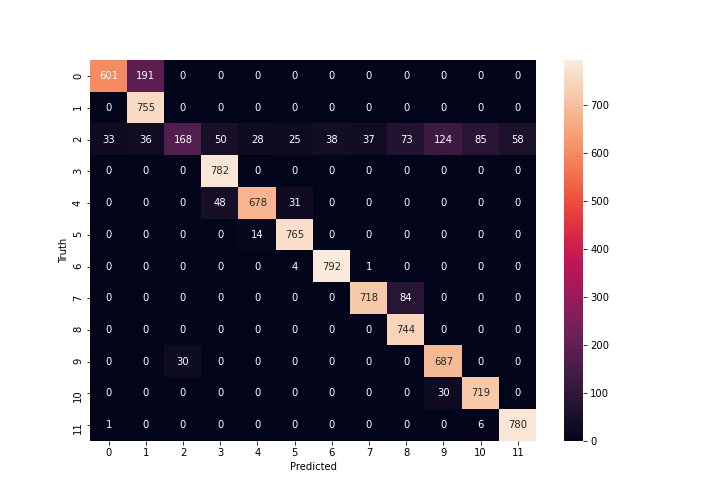
macro avg 0.89 0.89 0.87 9216

weighted avg 0.89 0.89 0.87 9216

***Table 5.1: Classification report***

5.4 Confusion Matrix

The confusion matrix of this model has been imported here,



***Figure 5.3: confusion matrix***

Chapter 6: Conclusion

Industrial and manufacturing machinery devices like the rolling bearing are needed to be predicted for remaining useful life for the maintenance of the industrial system. It is important for determining maintenance decisions, prolong critical equipment life, reduce maintenance costs, and plan to replace defective equipment at an early stage to minimize cost. This project attempted to predict the remaining useful life of bearing with a deep learning algorithm. The deep learning model ANN has performed well for predicting the remaining useful life. The accuracy was 89.88% which is good enough for the prediction. The goal of this research was to build and validate the Artificial Neural Network for predicting the RUL of bearing. The model could predict more effectively if the dataset was large enough. The overall project has been done with a good prediction over the test dataset.

# References

McKinney, W., 2011. pandas: a foundational Python library for data analysis and statistics. *Python for high performance and scientific computing*, *14*(9), pp.1-9.

Keras, 2020. *Keras documentation: Layer activation functions*. [online] Keras.io. Available at: <https://keras.io/api/layers/activations/> [Accessed 25 August 2021].

IBM, 2020. *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What’s the Difference?*. [online] Ibm.com. Available at: <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks> [Accessed 29 April 2021].

Koehrsen, W., 2021. *When Accuracy Isn’t Enough, Use Precision and Recall to Evaluate Your Classification Model*. [online] Built In. Available at: <https://builtin.com/data-science/precision-and-recall> [Accessed 25 August 2021].

Wood, T., n.d. *F-Score*. [online] DeepAI. Available at: <https://deepai.org/machine-learning-glossary-and-terms/f-score> [Accessed 25 August 2021].

Mohajon, J., 2020. *Confusion Matrix for Your Multi-Class Machine Learning Model*. [online] Medium. Available at: <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826> [Accessed 25 August 2021].