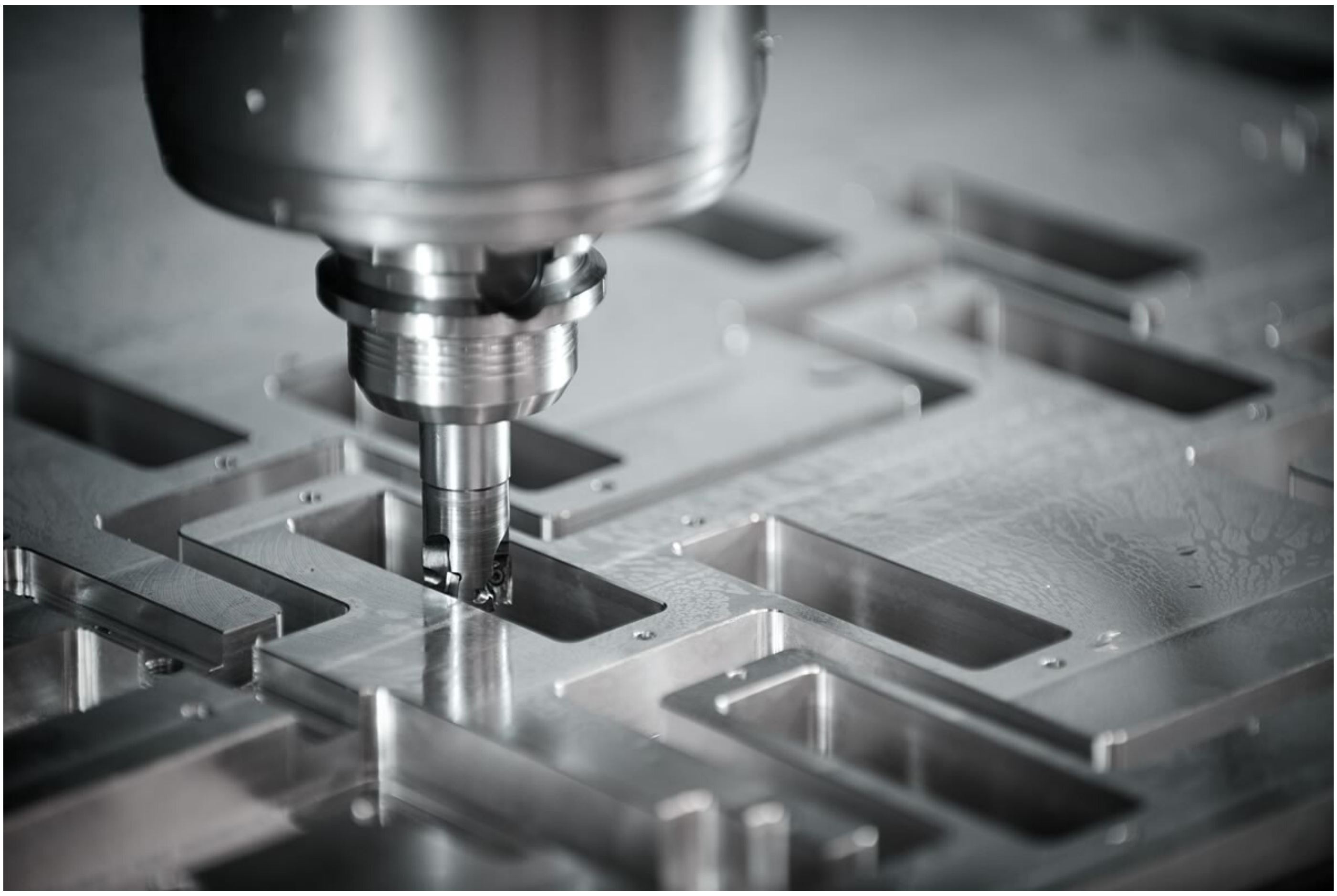
USE-CASE REPORT

Tool State Classification using Machine Learning and Deep Learning Algorithms



OBJECTIVE

The objective of this use-case is to develop an algorithm that will classify if the tool will wear out or not given raw machine inputs.

Abstract

Tool condition monitoring has become an essential industry phenomenon to achieve high-quality machine operation and cost management. Classification of tool state using the machine parameters will help in avoiding the failure of a machine and faulty products. In this use-case, we use the concept of machine learning and deep learning to classify whether the tool will wear out or not using the raw machine data. Using raw data as input to predict the tool state eases out any statistical calculation or filtering for the machine operators. Here we present machine learning approaches for classifying the Tool state and integrate the most successful approaches into the experimental workflow.

Contents

Introduction

What is Machine Learning?
Application of AI which provides systems an ability to automatically learn and improve from experience without being explicitly programmed.

Tool condition monitoring has become an essential industry phenomenon to achieve high-quality machine operation and cost management. This method monitors the performance and condition of equipment during normal operation to reduce the likelihood of failures

Dataset and Description

For this use-case, we used the dataset of CNC milling machine experiments. A series of 18 experiments were conducted whose time-series data was collected. The Experiments were conducted on a CNC machine to create a wax structure. These Experiments were done at the smart lab of the University of Michigan.

The dataset consisted of 25,286 instances of 51 features. The features were the measurement of velocity, acceleration, position, voltage, current, power, position taken by sensors attached to X, Y, Z, and spindle axes and also features like feed rate, pressure, material, process step, and tool condition.

Data Pre-processing

The dataset needed some cleaning and modifications. The categorical data were converted to numerical data using one-hot encoding. We considered dimensionality reduction using the PCA method for certain algorithms and data was also scaled using the standard scaling method.

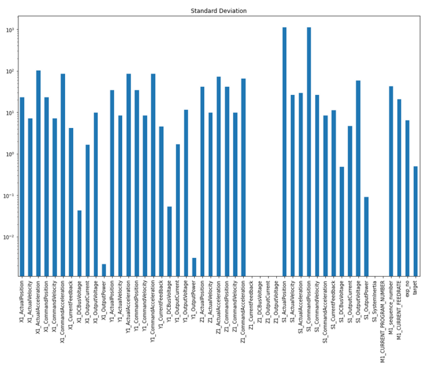
Data Pre-processing is necessary because features having null values make the model in-consistent which results in an inaccurate model. Also, categorical Data cannot be understood by the computer so it is necessary to convert it to numerical form. Data analysis tasks become significantly harder as the dimensionality of the data increases. As dimensionality increases, the number of planes occupied by the data increases thus the data becomes difficult to model and visualize.

Feature Selection

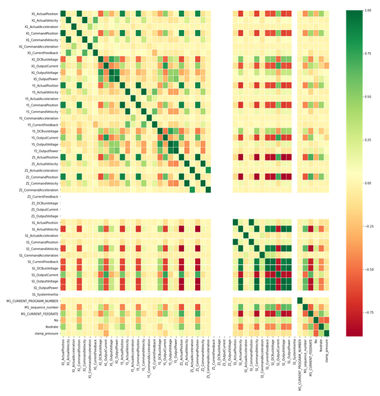
In this process we we selected those features which contribute most to the target variable and remove those features which affects negatively to the model performance.

These features do not contribute to model performance and they also increase the computation time. Feature Selection is an important process because it helps in improving the accuracy of the model, reduces overfitting because the model is not being trained on noisy data, and also training on fewer features requires less time.

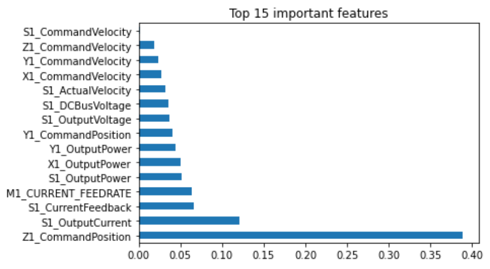
Some methods which were used for finding important features are calculation of standard deviation, correlation values and using library functions.



Standard Deviation



Correlation Matrix

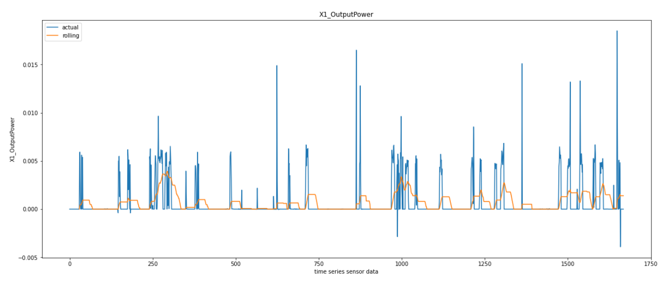


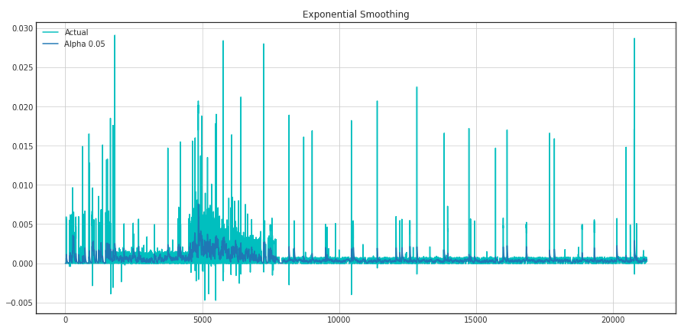
Feature Importance

Feature Creation

Feature creation is a step which is performed for a dataset or certain series if the features contain high amount of noise. Noisy data are inaccurate and are caused mainly due to human errors. So, we used used Sliding window method and Exponential Smoothing to remove the noise from the data.

Feature creation becomes an important step in case of noisy data because if the data is left untreated the model to learn the noise as a pattern and thereby decrease model accuracy.





Training and Testing data

We tried splitting the data in different ways to choose best balanced set.

- Clubbing all the experiments and choosing test and train set with 80:20 ratio. This split produced the best accuracy and auc-roc score.
- Creating all combinations (3200) of balanced experiment wise split and taking mean accuracy. This split produced less accuracy due to biased data.

Cross Validation and Hyper Parameter Tuning

We perform Cross Validation and Hyper Parameter tuning for every learning model.
We test the model on unseen data to see if the model overfits or underfits. Cross validation helps to train and test the model on entire data by splitting the data in k folds. It helps us to test the effectiveness of our machine learning model. By finding the mean of all K folds we get the best accuracy for our model.

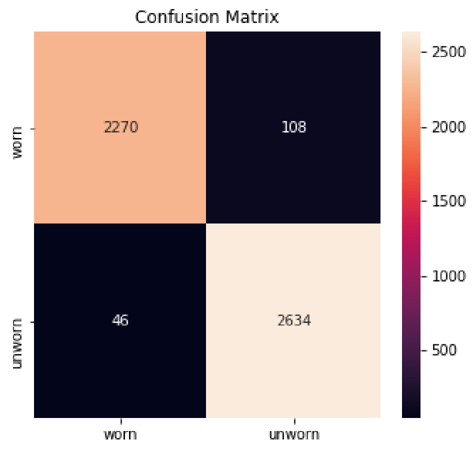
Hyperparameter tuning is a method to choose the best parameters for any model for which the performance of the model is best. The parameters like n_neighbours, max_depth, learning rate, optimizers, epochs, hidden layers etc. govern the training process of any model and produce a significant impact on the accuracy of the model. We used Grid search method for determination of best parameter for every model.

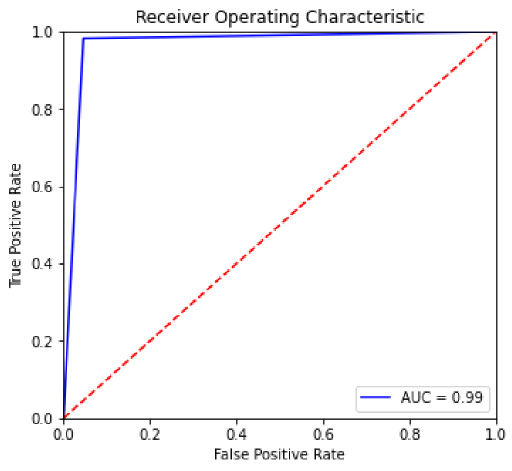
Learning Models

We trained and tested our dataset on different machine learning and deep learning models. Some Learning models like Decision tree classifier, Random Forest classifier, XG Boost, K Nearest Neighbours classifier and Neural Networks showed good accuracy and roc-auc score.

Decision Tree Classifier

Decision Tree is like a flowchart like tree structure where every node has a decision function and the leaves are the outcomes. Decision Tree takes very less computation time and is faster than Neural Networks. For noisy and unbalanced data the algorithm does not work well so we need to remove noises to improve the model in such cases. The hierarchy of the tree is decided on the basis of entropy.



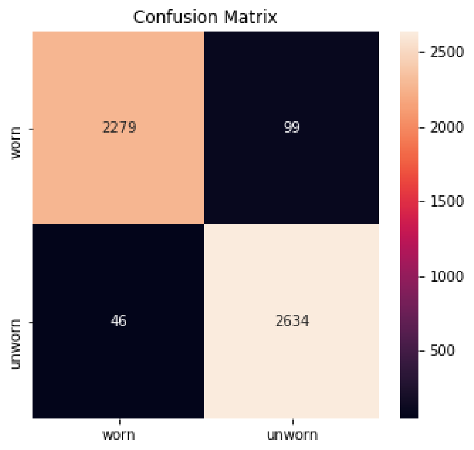


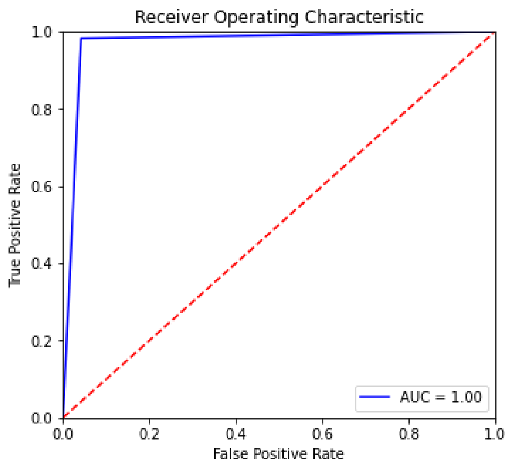
Training Accuracy: 0.986
Testing Accuracy: 0.969
ROC AUC SCORE: 0.985

Random Forest Classifier

It is an ensemble tree based algorithm. Random Forest classifier is a set of decision trees from randomly selected subset of training set. The trees are voted and the majority voting class decides the final class for the test set. Random forest provides the best accuracy but it takes longer computation time than decision tree, Artificial neural networks and gradient boosting algorithms.

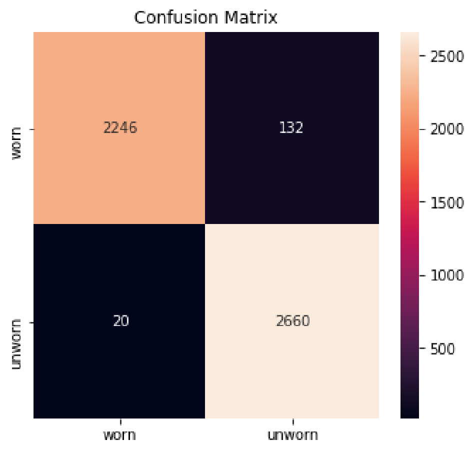
Training Accuracy: 1.0
Testing Accuracy: 0.97
ROC AUC SCORE: 0.99

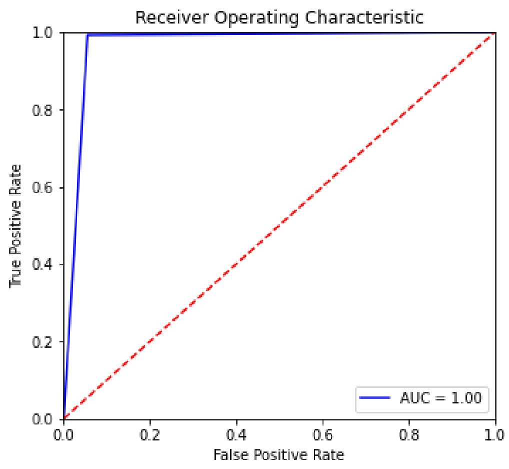




XG Boost Classifier

XG Boost Classifiers use optimized gradient boosting algorithm through parallel processing, tree pruning, handling missing values and regularization to avoid overfitting. It works best for small to medium structured data however it is outperformed by neural nets when large datasets are used.

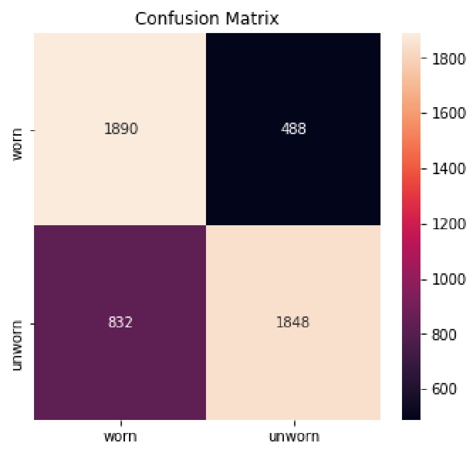


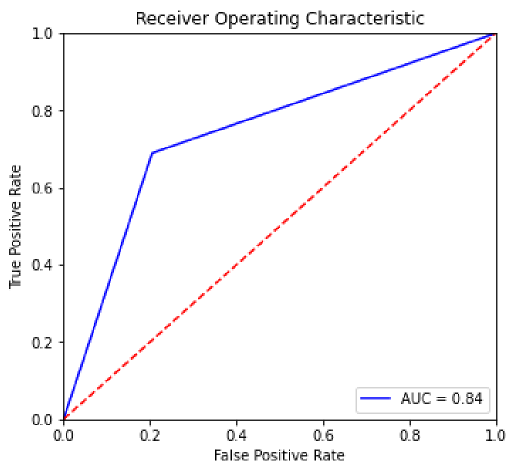


Training Accuracy: 0.972
Testing Accuracy: 0.969
ROC AUC SCORE: 0.997

K Nearest Neighbours

KNN is a non-parametric and lazy learning algorithm i.e; model structure determined by the dataset and KNN does not needs to trained for model generation so it makes testing phase slower and consumes high amount of memory. KNN model performs poor at high dimensional dataset as is the case. We normalize the dataset in this case and also tune this algorithm for different hyperparameters to achieve best estimator.





Training Accuracy: 0.807
Testing Accuracy: 0.735
ROC AUC SCORE: 0.831

Artificial Neural Networks

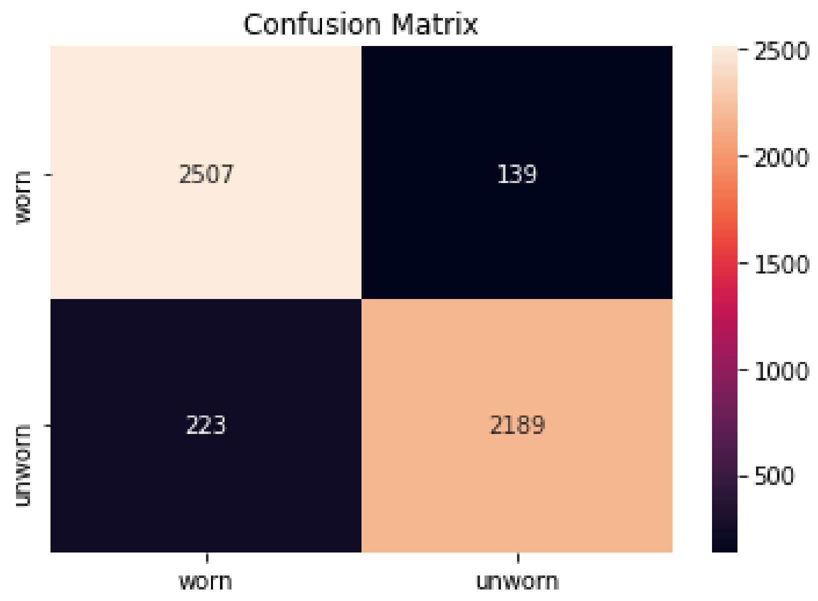
Artificial Neural Network consists of an artificial network of functions, called parameters, which allows the computer to learn, and to fine tune itself by analyzing new data. For this model the number of input parameters (34) were more than that of ML models (16).

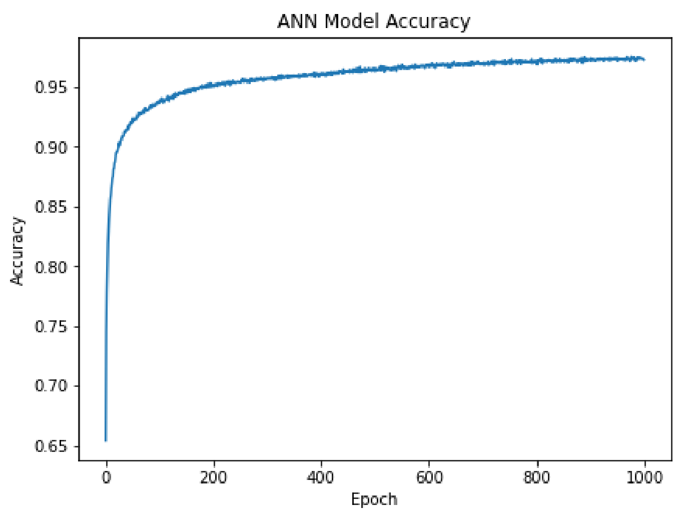
We created feed forward sequential model using keras library. The network consisted of three dense layers. It took a 2D input of dimension (1,34) and passed it to next dense layer of 17 nodes through a Relu activation function. The second dense layer passes its input through a function and Relu activation function to next layer of nine nodes which then passes it to the last layer of single node which uses sigmoid function to determine the truth value of the input.

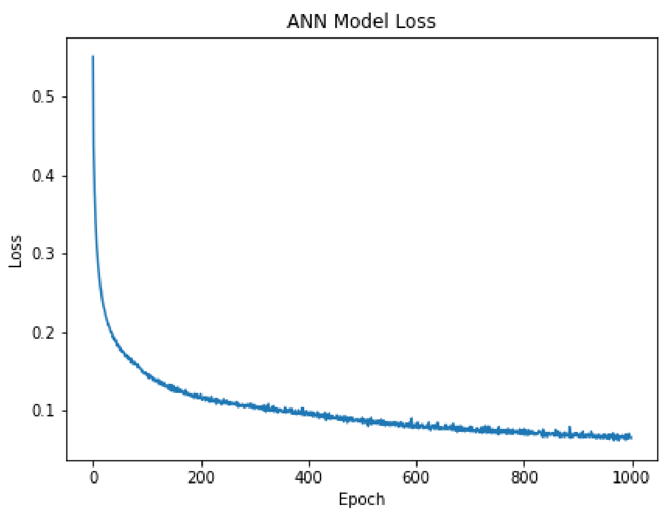
The train and test sets were scaled using standard scaler function so that input feature values implicitly weights all features equally in their representation.

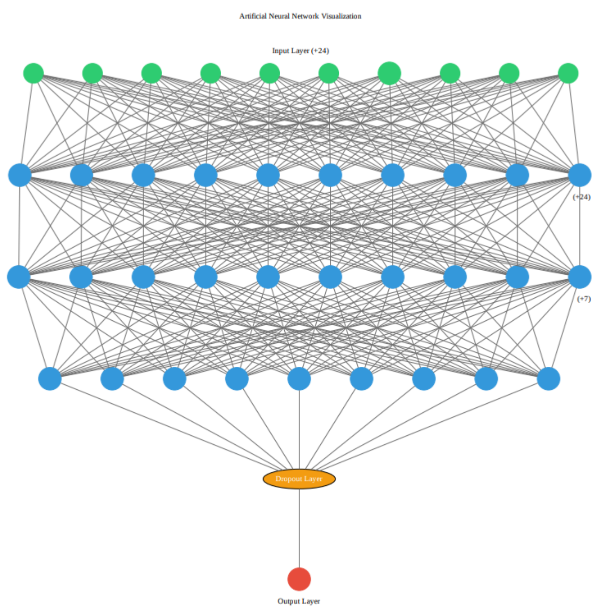
Accuracy and the confusion matrix generated from the Artificial Neural Network

Training Accuracy: 0.97
Testing Accuracy: 0.93







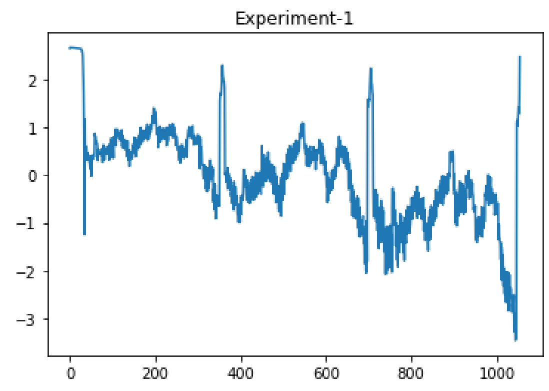


Visualization of the Artificial Neural Network which consisted of three hidden layers, input and output layers. The hidden dropout layers prevents the model from overfitting.

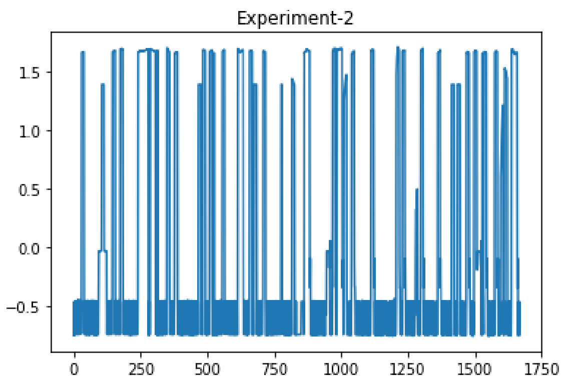
Long Short Term Memory Networks

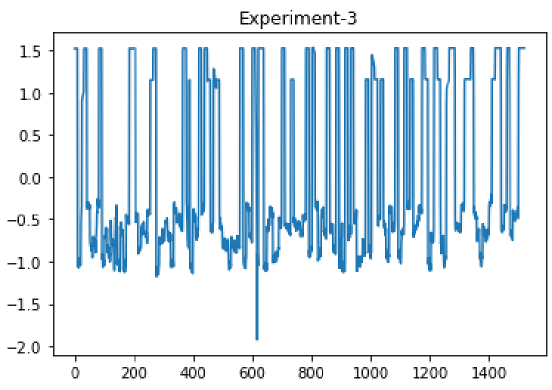
LSTM networks are special kind of Recurrent neural nets which are capable of long term dependencies. LSTM acts like regressor and forecasts future values.

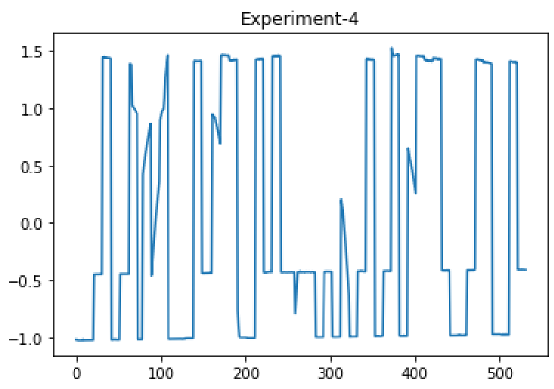
Here in this case we don't have the target feature as a time series decay value so, to apply LSTM we need to create a target feature that shows a trend with time for every experiment. To create such a feature we reduced all the features to one dimension feature using PCA to observe how the data behaves with time.

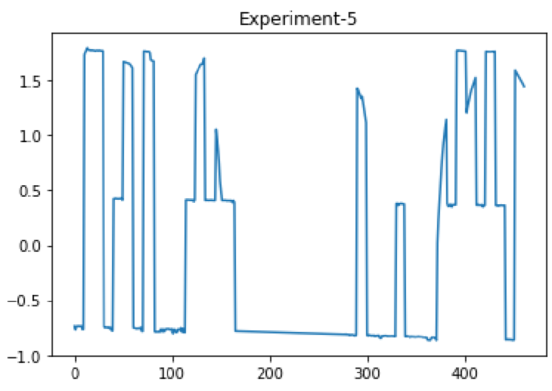


We see that the experiments do not show trend with each other and so using LSTM for this data will give absurd results. However, we created a LSTM Network and found the accuracy to be 51.3% which is very poor for binary classification use-cases.

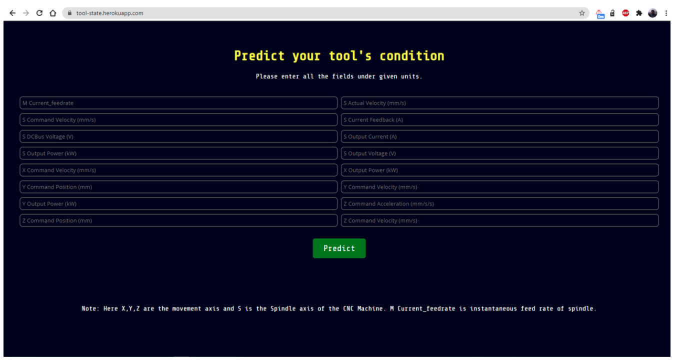








Building API and Web-Application Deployment


tool-state.herokuapp.com

Predict your tool's condition

Please enter all the fields under given units.

M Current_feedrate	S Actual Velocity (mm/s)
S Command Velocity (mm/s)	S Current Feedback (A)
S DCBus Voltage (V)	S Output Current (A)
S Output Power (kW)	S Output Voltage (V)
X Command Velocity (mm/s)	X Output Power (kW)
Y Command Position (mm)	Y Command Velocity (mm/s)
Y Output Power (kW)	Z Command Acceleration (mm/s/s)
Z Command Position (mm)	Z Command Velocity (mm/s)

Predict

Note: Here X,Y,Z are the movement axis and S is the Spindle axis of the CNC Machine. M Current_feedrate is instantaneous feed rate of spindle.

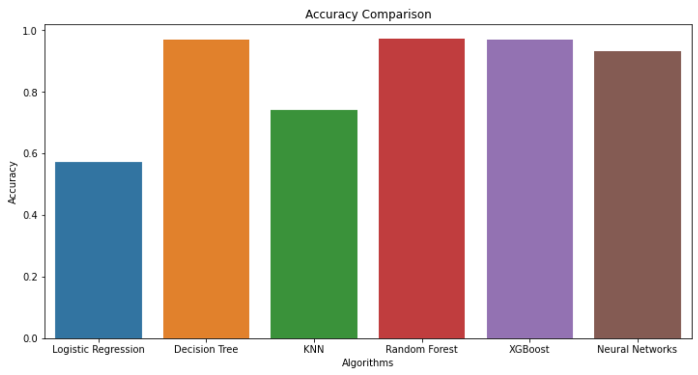
We choose Random Forest classifier as the prediction model and build an API using Flask. We created a Web-app using HTML and CSS and deployed the website using Heroku web services. The Web app takes 16 raw numerical machine features and predicts the tool state. The Web-App is live at **https://tool-state.herokuapp.com/**

Technology and Framework

- I used python language it is a powerful high-level language and I am well versed with it.
- I used scikit learn library for the development of Machine Learning models as scikit learn provides many useful libraries and has very good documentation of its tools.
- I used Keras library on TensorFlow as Keras is easy to learn and is a very powerful library used for deep learning purposes.
- For the data handling, computation, and visualization purposes Pandas, numpy, seaborn, and matplotlib was used.
- For the development of API, we used the Flask library and wrote the code in python. Flask is a well-known technology and helps in the creation of robust applications very easily. It also provides good documentation.
- The website was written using HTML and styled with CSS. The website is hosted on Heroku because it provides a sufficient amount of memory for free and easy hosting.
- I used Google colaboratory as our platform so that we can use it's fast and free TPU accelerators which decreases model training and testing time to a great extent.

Comparision and Conclusion

From the above use-case, we determined the important features which affect the tool state. The models were trained on a sufficient amount of data and so the results are reliable and can be used during CNC experiments to protect the cutting tool. The cutting tool is made of high-speed steel or carbides and is therefore very costly. We also determined the accuracy of algorithms and which of them will be most efficient for Web-app creation.



Mentors:
Ajesh Rajan
Rahmatullah Zacki

Performed By:
Nikhil Kumar
LTI Campus intern
IIT(BHU), varanasi