Anomaly Detection in Bosch CNC Milling Machinery using Ensemble Deep Learning

by

Kinnary Raval

Student Id : **2355359**

A thesis submitted to the
University of Birmingham for the degree of
MASTER OF SCIENCE IN ARTIFICIAL
INTELLIGENCE AND MACHINE LEARNING

Supervisor : **Professor Peter Tino**

UNIVERSITY OF
BIRMINGHAM

School of Computer Science
University of Birmingham

September, 2022

Acknowledgements

I would like to thank my Almighty, my family for their constant support and Professor Peter Tino who offered me guidance in this research and encouraged to reach out for better.

Abstract

Computer Numerical Control (CNC) machines are the pioneer of assembly lines for any manufacturing industry. Self-regulated fabrication plants which use CNC for milling, demand the equipment to operate at high-speed resisting unfavourable environmental factors despite task complexities. However, the breakdown of such equipment can cause a domino effect in delivering the targets and unplanned maintenance costs. It becomes crucial to identify the reasons causing the machine failure and recognize the peculiar traits before the actual malfunction. A model is proposed using a stacking ensemble of neural networks known as Convolutional Autoencoders which competently detects an abnormality in the series of vibrations. Using the reconstructive property of AutoEncoder, multiple AutoEncoders are tuned individually as sub-models on normal and anomalous data and feature space is regenerated. Feature predictions from each estimator are fed to a supervised meta-learner build using a dense neural network which classifies the signal as faulty and not faulty. The CNC Milling data which is used in the experiment is very recent and no benchmark models have been published. Thus, in order to prepare some baseline, two more models are build using classic machine learning algorithm Fast Fourier Transform and decision-tree based approach Isolation Forrest. We have compared the proposed model with other two using standard performance evaluation metrics and successfully achieved better results for anomaly detection.

Contents

List of Figures

List of Tables

1 Introduction

CNC milling machines are mechanically used to pulverize any material block such as a piece of aluminium, steel, brass, plastic, wood etc. into a custom-designed structure. It requires maintaining a high level of detailing and finishing. Heavy industries use it for mass production as these computerised panels provide uniform, consistent and accurate outcomes based on the CAD drawings provided as input. CNC milling machines use rotary tools which have 3,4 or 5 axes, to cut away the material and create the part.

1.1 Problem Statement

This equipment faces environmental and industrial challenges resulting in wear and tear during the production cycle. In addition to the execution of high-velocity operations, there is the frequent need of mounting and un-mounting of tools on the spindle depending on the shape, material, geometries, and coatings of the final product. This engenders issues such as tool misalignment, tool breakage, chip in chuck, chip clamping [1] etc and eventually leading to the breakdown of the machine.

The deformity in the final product caused by the defective machine, may get noticed by the quality assurance team at the ground level. However, due to manual checking and intervention, there are chances of missing out on intrinsic details or correct annotations. Also, the product experts cannot see-through the arising concerns within the confines of the machinery. This is just the tip of the iceberg of the difficulties faced by the manufacturing team as well as the consumer. Creating erroneous products, repairing and maintenance costs, production delays, commercial consequences etc. causes a ripple effect in this entire product lifecycle. Thus, it becomes vital to recognize and fix the issue beforehand.

1.2 Motivation and Proposed Solution

Similar distress has been raised by many elite organizations. Putting their faith in the advancements of machine learning, they have released classified datasets where the community has helped to solutionize this problem [2]. Research has shown that statistically,

data-driven models can be built using industrial data which can maintain the robustness and more importantly, generalization for condition monitoring systems[3].

In this project, we aim to study one such dataset [4] and propose a model to identify faulty vibrations in all the machines executing different types of milling processes(OP). Using ensembles of deep neural networks known as AutoEncoders(AE) built on Convolutional layers, it is intended to reconstruct the same signal and identify peculiar ones. In the first part of the study, the data is downsampled and preprocessed in order to make it trainable for AEs. Next, the refined data is divided into 2 sets of training and testing pairs: one set for training all level-0 models or sub-models which are convolutional AE and the second set for training level-1 or meta-learner which is a dense neural network. After the data split, level-0 AE models are trained with different hyperparameters on the first set and trained models are 'saved'. For modelling the level-1 classifier, the second set of data is used which first gets trained by pre-trained 'saved' sub-models and the output of these sub-models is fed as input on level-1. With the help of ensembling of AE, the variance in the prediction is reduced making the model consistent and stable.

In the second part of the study, two baseline models are created since the data we are using, does not have any published reference models.
The first baseline model is built using the Fast Fourier Transform (FFT) where due to computation limitations, the model is trained and tested on the data of one machine instead of three. In FFT based model, the frequencies of all three vibrations are filtered using a Bandpass filter. Using the concept of the confidence interval, the frequencies that do not fall under the standard normal distribution of 3 vibrations are removed. In the next step, Inverse Fourier Transform (IFT) is used and using a defined statistical error threshold (section 3.14), anomalies are detected.

For the second baseline model, a decision tree based algorithm known as Isolation Forrest is used for fault detection. Here, we have divided data into one set of training and testing. Using Principal Component Analysis (PCA), the dimensions are reduced first. Due to a random and recursive split of features in the algorithm separating anomaly early, an anomaly score is obtained which determines faulty

and normal accelerations.

Going further, the second section describes the research done tool condition monitoring and the state-of-the-art models proposed along with our idea. Third section elaborates the technical concepts that are needed to understand experiment, algorithms used and metrics used to evaluate the performance. Fourth section explores about the dataset used and the preprocessing techniques applied on it before modelling. Fifth section depicts the methodology used to implement all three models; prososed and two baseline models. Sixth section portrays the evaluations done on the models and how the performance has evolved.

2 Literature Review

In the territory of industrialization, the role of Machine Learning is paramount which has guided businesses to resolve the problems such as Condition Monitoring, Predictive Maintenance, Sales Forecasting, Anomaly Detection etc. It has gained popularity in the enterprise world due to its reliability and consistency in predictions. Especially due to state-of-art unsupervised algorithms for regression, clustering, image segmentation, and deep neural networks, prognostic results have increased financial stability and growth.

Detecting oddity in an operating machine happening due to constant wear and tear such as Milling machines, is one of the wide and prevailing use-cases in this domain. Various industry tools have been established [5] for the prevention of breakdowns causing a decrease in operational and maintenance costs. Nonetheless, more analysis and real-time data can help to keep a constant check on tool conditioning and upcoming challenges for machines such as CNC.

Due to advancements in Industry 4.0, the Internet of Things (IoT) and Cyber-Physical Systems, it has become easier to obtain the autogenous data of any machinery at work which can be used for analysis. Often the data received for modelling is in small batches and does not represent the entire system. It becomes crucial to offer a generalized model which does not compromise on efficiency for new unseen data. In this section, we discuss the state-of-art solutions provided for condition monitoring and anomaly detection in production lines.

For the purpose of achieving robust and generalized model, research has been conducted for tool health monitoring and in process quality [4]. Classical approaches such as K-Nearest Neighbours (KNN) with dynamic time warping [7] is performed on real-time data acquired from a turbine factory and have shown significant improvement in anomaly detection and correction than the Support Vector Machine (SVM) models. However, clustering based models such as KNN struggles with increasing number of features as computational performance is impacted quickly. Also, there is a risk of high model variance due to incorrect tuning of hyper-parameters.

One of the open source database known as SMART Lab Milling [6] Dataset hosted on Kaggle, aims to explore the tool health detection as well as detection of inadequate clamping. Algorithms involving the decision trees such as XG-Boost and Random Forrest resulted high accuracies for this data. Although CART based approaches are implemented quicker and more random variables or features can be presented for complex models, they ignore the time distributed nature of the data which is an essential factor to determine contextual

and collective anomalies (section 3.2) .

Another data source that is widely investigated and used for introducing novel methods is NASA Milling Dataset which represents data for tool wear in terms accoustic emission, current and vibration [2]. Authors using NASA Dataset in paper [8], used least squares support vector machine (LSSVM) as the classifier and achieved acceptable results for monitoring tool breakage. Because of having different sensors, various techniques emphasised drawing meaningful conclusions with just one sensor data. For example in the research shown by [9], vibration and force signals in time and frequency domain were studied however, it was recommended that vibration signals had stronger impact in detecting fault.

Signal processing and decomposition is one more way to proceed for tool condition monitoring. Since the faulty signals are non-stationary, distinguished faulty frequencies can be extracted such as mentioned in [10] using s-transform time-frequency transformation method. Dimensionality reduction methods such as principal component analysis (PCA) and linear discriminant analysis (LDA) are prevalent for such tasks [11] [12].

Deep Neural Networks have also showed remarkable outcomes when applied for outlier detection in sequential data. [13] proposed the use of long short-term memory networks (CBLSTM) in the milling process. Also fusion of time-frequency features created using statistical metrics and frequency band energy and then processed using Convolutional Neural Network (CNN) to detect gearbox faults [14]

AutoEnoders(AE) have picked up the reputation in anomaly detection as reconstructive and density estimation models. Estimating the comprehensive nature of data and developing a generic model for anomaly detection is the prolonged issue which community is trying to address. Using the proposed model, we intend to take a step ahead in that direction.

3 Background

3.1 Machine Learning

Machine Learning is the field where computer systems or programs automatically improves with experience. Tom Mitchell annotates Machine Learning in his book [15] as,

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Machine learning is an application of AI that allows the model to learn and improve from experience without separate programming using the data provided. It helps to generalize the results using performance measures.
There are 3 approaches to solving any machine learning problems :

1. Supervised : Supervised machine learning algorithms learn based on previously labelled data and predicts the unseen future data.
2. Unsupervised : Unsupervised machine learning algorithms works on the unlabelled data and tries to learn a pattern (linear or non-linear) or similar clusters.
3. Reinforcement : Reinforcement machine learning algorithms interacts with its environment by producing actions and discovering errors or rewards.

3.2 Anomaly Detection

As stated by [16], anomaly detection can be defined as follows

In data analysis, anomaly detection (also referred to as outlier detection and sometimes as novelty detection) is generally understood to be the identification of rare items, events or observations which deviate significantly from the majority of the data and do not conform to a well defined notion of normal behaviour.

An outlier can be termed as an observation which differs from other observations to the extent that it raises concerns that it was generated by a different working. In time series data, an anomaly is a data point which is not able to fit the common collective trend

or cyclic or seasonal pattern of the entire data and is significantly different from the rest of the data.
There are mainly three types of anomalies based on the type of data and situation to be analysed :

1. Global Anomalies : Data which completely falls outside the cluster or cloud of all other data records, stating it is a rare event, corresponds to Global Anomaly.
2. Contextual Anomalies : Data that does not relate to same distribution for similar points under the same context (mainly temporal) are considered contextual anomalies.
3. Collective Anomalies : Collection of related data instances that is anomalous with regards to the entire data set, it is considered as a collective anomaly. Here, individual values may not be anomalous, but its communal occurrence makes it collective anomaly.

3.3 Data Preprocessing

Data Preprocessing comprises of steps in order to convert the raw data into meaningful data which be parsed by the models and computation becomes faster. Data mining includes steps such are removal of missing or null values, converting all feature vectors to same scale known as standardization, normalizing values between 0-1 making predictions meaningful. Removal of inconsistent data leads to better model learning and reduces false predictions.

3.4 Fourier Transform

A Fourier transform (FT) is a mathematical form of signal decomposition of a time dependent functions into frequency dependent functions (spatial frequency or temporal frequency).The FT of a function is a complex sinusoids of actual function where real value represents amplitude of that sinusoid and imaginary values describes complex sinusoid's phase offset. For any function $f : R \to C$, its integral FT can be evaluated by :

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\xi x} \partial x, \forall \xi \epsilon R$$

The implementations of FT can be :

- Discrete Fourier Transform (DFT) : It is a discretized version of FT, mapping evenly spaced time series to discretized frequency domain
- Fast Fourier Transform : It applies algorithms such as Cooley–Tukey algorithm on DFT and makes it more efficient to implement.
- Inverse Fourier Transform : It is actual inverse of FT, transforming a a discrete or continuous frequency spectrum into a temporal function.

3.5 Bandpass Filter

In signal processing, a filter is usually used to remove unwanted or noisy frequency components of a signal. A band-pass filter when applied over a signal, it allows through only those components which are specified between a given range or band of frequencies and blocks components with frequencies above or below the given values.

3.6 Confidence Interval

Confidence interval is a statistical range defined such that there is a probability that the value of the given parameter will fall within the range around the mean. It helps to measure the degree of uncertainty in a sampling method. In our project it is used to create Bandpass filter, so that frequencies with lesser probability not falling under 95% confidence interval are removed.

$$CI = \bar{x} \pm z\frac{s}{\sqrt{n}}$$

where $\bar{x}$ is the sample mean , z is the confidence level value , s is the sample standard deviation and n is the sample size.

3.7 Isolation Forrest Algorithm

Most of anomaly detection techniques are optimized and trained on normal instances, because of which model when exposed to test data either predicts more false positives or may detect very few anomalies. Isolation Forest, a decision tree algorithm based on Random Forest overcomes the above challenge with the assumption that anomalies

are very rare, distinguishable and easier to isolate from other. In order to isolate a data point, the algorithm recursively generates partitions on the sample by randomly selecting an attribute and then randomly selecting a split value for the attribute, between the minimum and maximum values allowed for that attribute.

3.8 Artificial Neural Networks

Artificial neural networks are machine learning algorithms inspired by the working of the brain connected with interlinked nodes known as neurons. Each connection, also referred as an edge, can transmit a data to other neurons. The output of each neuron is computed by some non-linear function of the sum of its inputs recognized as activation functions. Neurons and edges have a weights and biases which gets adjusted as the learning proceeds using gradient-based optimisation algorithms.

3.8.1 Activation Functions

The transfer functions used to translate the summed weighted input from the previous node to an output value which is served to the next hidden layer or as a model output. Some of the widely used activation functions are :

Rectified Linear Unit

Rectified Linear Unit (ReLU) is computationally efficient as it does not activate all neurons at once. The neurons will activated if the output of the linear transformation is more than 0.

$$f(x) = max(0, x)$$

ReLU accelerates the convergence of gradient descent towards the global minimum of the loss function due to its linear, non-saturating property. However, it faces the constraints of Dying ReLU. While considering the derivation of ReLU, the gradient values for negative values is 0, creating some dead neurons as their weights and biases are never updated. This reduces the model's ability to train or fit the data efficiently. This can be resolved by choosing variants of ReLU such as Leaky ReLU, Exponential Linear Unit (ELU) or by reducing the learning rate.

Sigmoid

This function converts the real valued input to the output valuing between 0 and 1. More positive and larger the input, output value will be more closer to 1.0 ; similarly the output will be closer to 0 if input is more negative or smaller.

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

The function is differentiable and used for probability predictions as the value ranges between 0 and 1. However, the gradient value approaches zero, the network stops to learn and suffers from the Vanishing gradient problem.

Softmax

Softmax function is outlined as a combination of multiple sigmoid functions. It calculates the relative probabilities of each class. It has been most prevalent choice for output layers.

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

where z is the input vector, K is the number of classes, $e^{z_i}$is standard exponential of input vector e^{z_j} is standard exponential of output vector.

Tanh

Being very similar to the sigmoid/logistic activation function, it differs in the output range of -1 to 1. In Tanh, larger or more positive inputs are closer to output value 1.0, whereas the smaller or more negative inputs, are closer to -1.0

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

It is more preferred than sigmoid, as unlike sigmoid function, the mean is 0 centered for the hidden layers not restricting the gradients to move along any directiond. But, it still faces the similar issue of Vanishing gradient.

3.8.2 Backpropagation

Backpropagation is a concept used to compute the gradient of the loss function with respect to the weights of the network while fitting the neural networks, and does so efficiently using the chain rule of derivation. Thus it computes the gradient of one layer at given point and then iterates backwards from last layer, avoiding redundant derivation calculation of interim terms. It can be summarised as :

3.8.3 Convolutional Neural Networks

A convolutional neural network (CNN), a regularized ANN, comprises of an input layer, intermediate hidden layers and an output layer. In a CNN, the hidden layers include layers that perform convolutions, unlike other feed-forward layers where output is masked with activation function and final convolution. Here, hidden layer performs a dot product with the layer's input matrix using predefined the convolution filter or kernel. As the convolution filter moves along the input vectors in the layer, a feature map gets generated, contributing to the input of the next layer.

The feature map produced by the convolution of a filter with any input vector can be expressed as :

$$g(t) = \int x(a)w(t-a)da$$

where x is the layer input, w is the filter and t is the amount of shift used. It is widely used for pixel data or in image based tasks such as segmentation, recognition etc. It is mainly followed by pooling layers, fully connected layers, and normalization layers.

3.9 Autoencoders

Autoencoders are the ANNs which are used to learn the useful latent information retained under the input distribution in unsupervised fashion. The latent meaningful information, also referred as encoding are refined and validated by reconstructing the input using the same encodings. The AEs learn a representation for a set of data mostly by reducing the dimensions and is trained against the noise present in the data. The Encoder takes the input vector x and converts it to latent space vector z and the Decoder is responsible to reconstruct the input, $\hat{x}$ from z portrayed as below :

$$Encoder : f_\phi : X \to Z; z = f_\phi(x)$$
$$Decoder : g_\theta : Z \to X; \hat{x} = g_\theta(f_\phi(x))$$

3.9.1 Reconstruction Loss

In order to reconstruct the data, it optimizes reconstruction loss which is mostly Mean Squared Error Loss between the actual input and regenerated input formalized as below :

$$L_{rec} = \frac{1}{d}\sum_{j=1}^{d}(x^{(j)} - g_\theta^{(j)} f_\phi(x))$$

Reconstruction loss penalizes incorrect data and thus, Encoder usually learns to encode features that explains the density estimation of most of the data giving higher level detailing of the underneath data.

3.9.2 Bottleneck Autoencoders

However, if the layer units are kept of same number as the input vector, AEs tries to learn the identity function and the model learning is futile. Thus, bottleneck layer is always there where we obtain compressed and only meaningful information. There is always a tradeoff between the compression and reconstruction as more compression leads to identifying high-level features and missing out finer details during reconstruction and vice-versa.

3.10 Ensembling

Ensembles refers to the combination of learnings from several base estimators modelled using a given algorithm to obtain better predictive modelling performance over a single estimator. The combination of the models is created by using different models as weak learners, or using same models with differnt hyperparameters or training the same model on different samples of input. This concept thrives that merging the predictions of weak learners is always better than the random choice output. It adds up the overall model complexity but becomes computationally efficient since weak learners are easier to train. Three widely used ensembling methods are described as below.

3.10.1 Bagging

For a given sample of data, several bootstrapped sub-samples are used at a time. Weak learners train the algorithm on these sub-samples and using an aggregation algorithm, the predictions are combined to create the most efficient estimator. Random Forrest is one of the algorithms using Bagging technique.

3.10.2 Boosting

Boosting uses the same weak learners which are fitted on the same set samples. Over every iteration, hyperparameters related to class weights are automatically tuned based on the incorrectly classified data. Gradient Boost, XG-Boost, Adaboost are few of algorithms using Bossting technique.

3.10.3 Stacking

An extension to the Moving Average ensemble, where multiple sub-models are considered to contribute equally for final prediction, in stacking a new model is entirely trained taking outputs from sub-models as feeding it as input. This meta-model results in better predictive performance than any single estimator.

3.11 Loss Functions

Loss functions help to evaluate how well the model has fit the data as models tend to learn based on the deviation depicted by loss functions.

3.11.1 Mean Absolute Error (MAE)

Also referred as L1 Loss, MAE is measured as the average of sum of absolute differences between predictions and real observations. It is more robust to outliers than MSE since, it does not square the error difference. However, it needs linear programming tools to compute gradients.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

3.11.2 Mean Squared Error (MSE)

Also referred as Quadratic Loss/L2 Loss, MSE is calculated as the average of squared difference between predictions and real observations. Due to squaring, the predicted values away from actual ones are penalized heavily compared to the less deviated predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

3.11.3 Cross Entropy Loss

Cross-entropy is built upon the idea of Entropy, where it calculated the difference between two probability distributions using total entropy between the distributions.It increases as the predicted probability diverges from the actual label. Cross entropy loss heavily penalizes the predicted values that are confident but wrong.

$$CEL = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

3.12 Objective Function Optimizers

An optimizer algorithm in neural networks helps to modify the attributes and hyper-parameters of the neural network, such as weights, learning rate etc. It reduces the final model loss and increase accuracy.

3.12.1 Adam

In any gradient based optimizer such as Gradient Descent or Stochastic Gradient Descent, we have single learning rate for all weight updates and the it does not change during training. Where as in Adam, as stated by Diederik Kingma and Jimmy Ba [3-1], *"the method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients".*

3.13 Evaluation Metrics

Evaluation metrics explains the model performance analysed using the test data set. It helps to discriminate between the different models and their capability for problem solving.

3.13.1 Confusion Matrix

It is a contingency matrix which allow visualization of the performance of an algorithm for binary classification in supervised setting. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa – both variants are found in the literature [3-2] It has statistical metrics that gives deeper insights on types of positive rates and negative rates. There are many metrics for it, describing the two which we have used in the project. For given True Postives (TP) , False Postives (FP) and False Negatives (FN) , calculations for Recall and F1-Score is shown below.

3.13.2 Recall

Recall describes the ability of a classifier to correctly find all positive samples . For each class, it is defined as the ratio of true positives to the sum of true positives and false negatives. [3-3]

$$Recall = \frac{TP}{TP + FN}$$

3.13.3 F1-Score

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy. [3-3]

$$F_1 Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

3.13.4 ROC-AUC

The receiver operating characteristic curve determines the model performance for classification tasks where the tradeoff between true positive rate and false positive rate is depicted against all thresholds. Area Under the Receiver Operating Characteristics Curve (AUCROC) represents the degree of separability determined by classifiers. Higher the measure of AUC, the better the model is at predict-

ing positive classes (0 for 0 and 1 for 1) than the mis-classification. AUCROC is not differentiable, thus not used with gradient-based learning algorithms.

3.14 Principal Component Analysis

Principal Component Analysis (PCA), is a method for reducing dimensions of large data sets. It transforms the original feature space into kernalized feature space which is smaller than the actual one while preserving the useful information.
PCA is calculated using the covariance of the standardized data and eigenvectors and eigenvalues of covariance matrix are computed to evaluate principal components.

4 Data Extraction and Preprocessing

4.1 Data Collection

The database used in this project was first presented by Bosch officials during the 55th CIRP Conference on Manufacturing Systems in the year 2022 and soon, was published by Procedia CIRP [4]. Previously, most of the CNC related research datasets [2] [6] have been cultivated in the laboratory and mapped for a shorter time frame com- pared to our Bosch dataset. The idea behind these data records is to provide the working statistics belonging to a real-time production environment retrieved from multiple machines over a longer span of time.

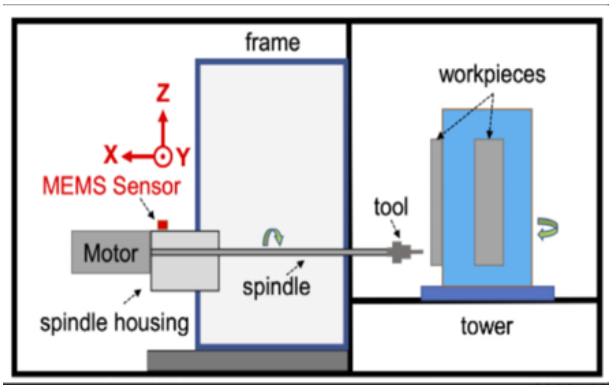

Figure 1: Schematic diagram of 4 axis machining center

Thhe data is collected from three brownfield milling CNC machines at four different time frames being February 2019, August 2019, February 2021, August 2021 within two years intervals (2019-2021). As shown in the figure 1, the tools used for milling are mounted on the rotating spindle connected to a motor.
For this experiment, the accelerations of the spindle are measured using a tri-axial CISS accelerometer which senses and records the vibrations in 3 different directions X, Y and Z. 15 types of milling activities (operation processes: OP), such as step drill, simple drill, t-slot cutter etc., have been conducted for each machine and the recordings are gathered at the sampling rate of 2 kHz i.e 2000 samples per second.
Hence, for each machine, it forms multiple instances of signal record-

ings; where each recording comprises samples corresponding to 20-150 seconds of readings of 3 different vibrations labelled OK and NOT-OK depicting the health status of machines as referred to in figure 2.

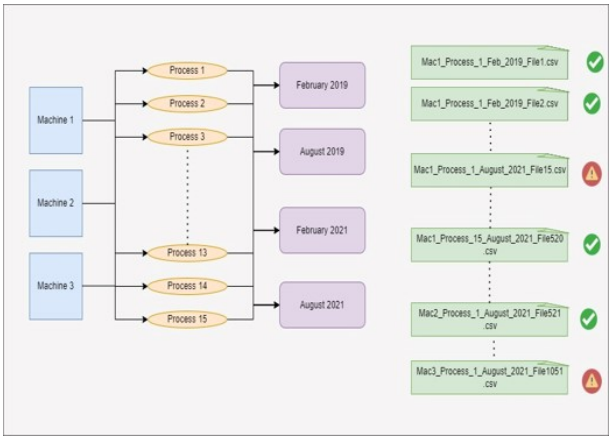

Figure 2: Data layout and record file formation

4.2 Data Extraction

The data published by formal Bosch representatives is hosted on GITHUB : https://github.com/boschresearch/CNCMachining. The individual vibrational sequences across 3 channels are stored in the format of .h5 file. Helper packages are used to load the data for all machines provided on github.

4.3 Exploratory Data Analysis

With regards to the data pulled from Github, basic data analysis is conducted for better understanding. Since there are approximately 0.1 million samples per signal, due to computing limitations EDA was done on limited set of signals.
First, the normal and abnormal accelerations across 3 axis during a single tool operation are compared visually. As shown in 3, the left signal shows normal vibrations in X,Y,Z axes for 28 second and the one on right displays the faulty signal. Clearly, we need more

insights in order to characterize abnormal data.

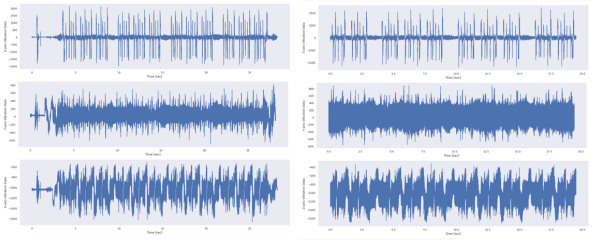

Figure 3: Vibrations recorded for single OP of normal signal (L) and vibrations recorded for same OP for faulty sequence across all three axes

Moving ahead, knowledge of class imbalance becomes quite vital because if not accounted for, model will falsely converge by classifying everything to normal as the majority class. Since the split of the data, specially for training and validation, needs to maintain the ratio of both classes, analysis is done on class distribution machine-wise and process-wise.

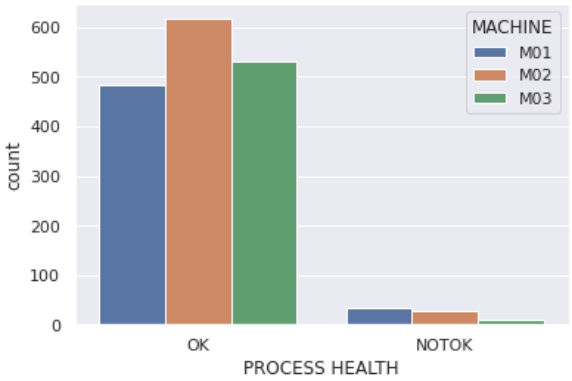

Figure 4: Class distribution per machine

It becomes necessary to check the distribution and correlation of all three features present in the data. Also, exploring the skewness in the features is helpful to define the pre-processing methods needed. The boxplots of the distribution of all features, as shown in figure 6, depicts the spread and locality of the data and how the mean of

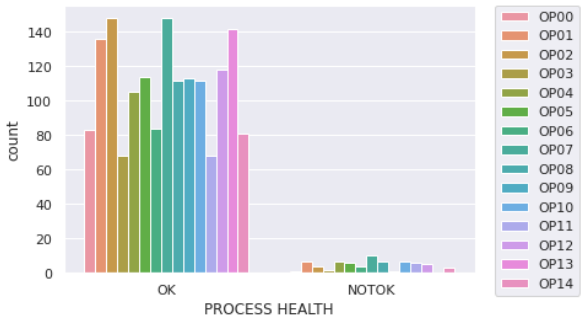

Figure 5: Class distribution per operation process

each feature varies vastly.

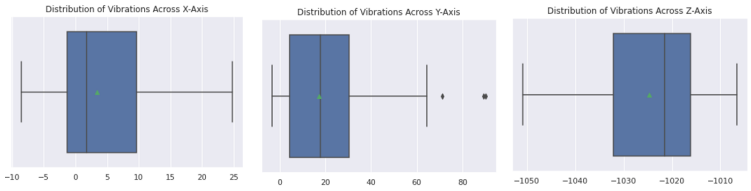

Figure 6: Boxplot of X,Y,Z Acceration data in (L), (C) and (R)

4.4 Data Preprocessing

Real-world data is raw, affected by various environmental factors and involves human intervention. All these reasons make the available data inconsistent, incomplete and erroneous. In order to fix it and mould it in model crunch-able format, data pre-processing is a vital step in data science.
Based on the EDA performed on data, series of data pre-processing steps are conducted.

4.4.1 Data Padding

As mentioned, the samples for vibrations along 3 axes are taken for different amounts of time, thus some sequences are for 20 seconds while some are for 140 seconds. This becomes an issue for neural

networks as same-sized feature vectors are required as input. In order to overcome this issue, zero padding is done i.e filling all missing values with zeros. This method reduces complexities compared to other ways to handle dynamic size time-distributed data

4.4.2 Data Downsampling

The experiment involves creating three types of models where one is based on Fast Fourier Transform (FFT) and other two are data-driven model based approaches.

As mentioned about data extraction in [4], the sampling rate of retrieving accelerations samples, which is 2 kHz, is equivalent to Nyquist frequency rate. Nyquist frequency rate (NFR) describes that in order to regenerate the original signal, the minimum sampling rate should be twice the original signal frequency. In our case, the original signal is between 75Hz - 1KHz and thus NFR should be 2 kHz minimum. Thus, if we downsample, there is no guarantee we can reconstruct the signal. However, due to resource limitations, we have to reduce the samples by a factor of 10.

For other model based approaches, based on computation limitations and time taken for model training, we downsampled the data by factor of 250 retrieving 80 samples per second by taking the mean of records within each window.

4.4.3 Data Normalization

In Data normalization we rescale the original data in such a way that the final values are between 0-1. As noticed in fig 6 , the range of X, Y and Z vibrations are varying by a greater margin, some have larger negative values and some have lesser spread or variance. In order to fix this, decrease the training time and increase model performance, data normalization is implemented. Min-max normalization (usually called feature scaling) has been used as a pre-processing technique to perform a linear transformation on the original data. Data x is normalized according to the below formula :
Min-max normalization (usually called feature scaling) has been used as pre-processing technique to perform a linear transformation on the original data. Data x is normalized according to below formula :

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

5 Data Modelling

5.1 Data Modelling using Individual AutoEncoders

5.1.1 Architecting Neural Network

In order to achieve the goal of anomaly detection using deep neural networks, reasearch has shown that AE and its variant such as Variational Autoencoders (VAE) are widely chosen [17] [18]. Building on the same ANN, we have used AE with different hyperparameters.

The general flow for simulating the model is considered as below:
Each sequential data of spindle accelerations, indicated with three features, is fed to the input layer of the AE. Using the property of AE to learn the encoding of latent space from feature space and rebuild it back to feature space, it is designed to reconstruct the same sequence based on how the model is learning and evolving. Here, it is to be noted that since the number of features are quite less, at the bottleneck stage of AE, we have to increase the units than the actual features. Upon obtaining reconstructed signals, the error between the original vibrations and predicted vibrations is calculated. Utilizing the standard error-thresholding rule, the instances surpassing the threshold value of error are considered anomalous.
Thus, accounting for the general flow, it is intended to decide the type of neural network layer, the number of layers in AE, units and optimize the architecture based on the data.

Layers

Since, our data is a time-series data, experiments started with Recurrent layers as they are specifically used for temporal use cases. For recurrent layers, LSTMs (Long-Short Term Memory) were chosen as units but computationally it turned out very heavy and the performance was also the issue. Also, it was then noted that LSTMs are more beneficial to determine next state but here, we are looking for determining whole sequence. Next we tried convolution neural networks (CNN) which have also proved substantial to perform better for chronological data [19]. Based upon the time taken for training and performance, Autoencoder with Convolutional Layers are finalized.

Number of Layers, Units and Kernal Size

Initially, the investigation about the number of layers starts in accordance with the minimum layers needed for encoding and decoding of AE. The layers are embedded with standard 2^n units which are then added gradually as the model performance is significantly increased.

Similarly, in order to fit an efficient kernel which can be used for low-pass and high-pass filtering in our case, different sized kernels such as 3*3 , 5*5 and 7*7 are evaluated against the model's efficiency to determine anomaly. It is proved that smaller kernel sizes provided better useful feature extraction and that increased model performance. Based on the community preference for activation functions and the use case, trials are conducted with Leaky ReLU and ReLu from which ReLU is chosen for intermediate layers.

Although the layers of AE and intrinsic details are decided by systematic experimentation, there are many other techniques to determine the same such as the Pruning and Heuristic approach [20]. In pruning, the idea is to remove nodes during training in the network by recognizing the ones which would not affect model performance significantly if removed. In Heuristic one, a directed search across formations such as an evolutionary algorithm or Bayesian optimization is conducted. However, since these methods require more time, resources and knowledge, we could not implement it.

5.1.2 Data Splitting

The project aims to provide a generalized model which is robust against any kind of milling operation conducted by a machine. Thus, it is decided to analyse the data considering machines as the context and not operational processes. The data of Machine 1 and Machine 3 are used for training and validation whereas Machine 2 data is used for testing.

5.1.3 Cost Function

With regards to cost function, it is again based on the type of data we are handling as well as model evaluation. As the data is of numerical type, it is decided to choose between Mean Absolute Error (MAE) or Mean Squared Error (MSE). Trials conducted on both the

objective functions with Adam as optimizer are shown in the figure 7. It is concluded that since MSE rate is lesser and the function is easily differentiable than MAE, it is a better choice.

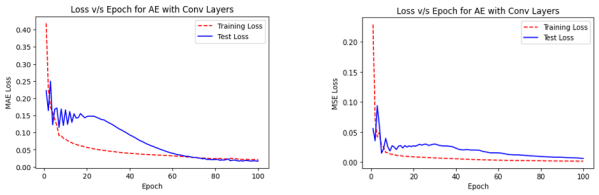

Figure 7: Comparison of Loss along with training and validation set

5.1.4 Error Optimizer

After determining cost function for the model's error rate, few optimizer are investigated considering the state-of-art performers. As shown in the fig 8, it is evident that Adam provided better outcomes and the default hyper-parameters as : learning-rate $\eta = 0.001, \epsilon = 1e^8$, initial decay rate of 1^{st} and 2^{nd} moment of gradient $\beta_1 = 0.9, \beta_2 = 0.999$ which are adaptively changing the learning rate met the required criteria.

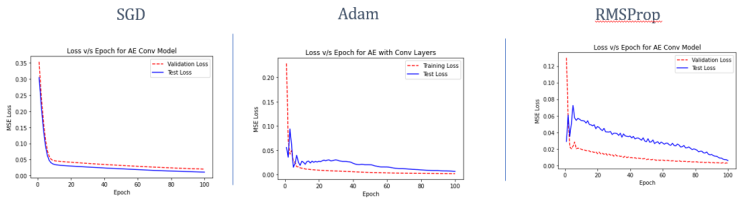

Figure 8: Comparison of 3 optimizers based on error loss along with training and validation set

5.1.5 Error Threshold for Anomaly

After training the data using the above parameters, we need to identify the sequences which have failed to reconstruct. This is specified as those instances whose error rate is more than certain limit making it fall as 'NOTOK' or faulty sequence. This threshold limit for the error rate is determined using standard statistical threshold estimation of 2 sigma rule. Thus, if the error value is more than 2

standard deviation off from the mean, it is considered anomaly.

$$threshold_{error} = \bar{error} + 2 * \sigma_{error}$$

where $\bar{error}$ and σ_{error} are the mean and the standard deviation of MSE distribution respectively.

5.1.6 Evaluation Metric

AUCROC has been used as the metric to calculate the model accuracy. Also, a confusion matrix is created to check the contingency and classification measurement, specifically Recall. In our current scenario, it is easier to get good precision since there is a class imbalance toward normal signals. However, to improve predictions of actual anomalies as anomalies and not as normal instance, we have accounted for Recall of more importance than Precision.
Data has been trained for 100 epochs with the batchsize of 100 instances; parameters which are concluded after certain number of trails.

5.2 Data Modelling using Stacking Generalization

Although our previous experiment with individual AE has shown good accuracy, there is a key issue with it. AE are very stochastic in nature, which is proved with every iteration, and the set of detected anomalies is different with every run. A few of the reasons for that happening are random weight initialization for AE or both classes have equal weights whereas the dataset is imbalanced.

One way to overcome this is by evaluating the output of the multiple AE and feeding it to another model which uses supervised learning to perform prediction. As discussed in (section 3.10.3), stacking is one of the ensemble techniques where the meta-learner model, also known as level-1 is used to combine predictions of various weak-learners, known as level-0 models and thus integrates the best of each model giving optimum results.

5.2.1 Base Learners

As level-0 models, we are using the same AE trained as the individual ones with the addition of a dropout layer after the first layer. Also, the data splitting is modified since the training dataset for

level-0 and level-1 models should be different. The instances are divided into 2 sets of training and testing pairs. All sub-models are trained and tested with the first set of data-split, whereas the second set of data-split is used to retrieve outputs from trained level-0 models and these outputs, are fed in for level-1 training. We have used 3 base learners AE which differs from each other in terms of dropout rate (0.5, 0.2, 0.1 after first layer) and the no of convolutional units.

$Model1 : 128 \Rightarrow 64 :: 64 \Rightarrow 128 \Rightarrow 3; Dropout : 0.5$
$Model2 : 64 \Rightarrow 32 \Rightarrow 16 :: 16 \Rightarrow 32 \Rightarrow 64 \Rightarrow 3; Dropout : 0.2$
$Model3 : 64 \Rightarrow 32 \Rightarrow 16 :: 16 \Rightarrow 32 \Rightarrow 64 \Rightarrow 3; Dropout : 0.1$

We train the 5 instances using 2 model instances twice. These models are decided after evaluating the impact of dropout and how to avoid AE overfitting the data.

5.2.2 Meta Learners

As a part of model blending i.e stacking generalization, a level-1 model is built using a neural network with dense layers. Using similar strategies for determining layers and units (section 5.2.1), we create a 3-layered network with one input layer, one 20 neurons hidden layer and one 2 units output layer. As an activation function, we have used ReLU at the intermediate layer and Softmax for the final output layer since it is a binary classification problem and we need probabilities for each class. The model is trained for 20 epochs, with Adam optimizer and MSE as loss function.

Models for performance comparison

With the purpose of building generalized model for heavy-duty processes, we also need other models to compare the efficiency of proposed learner. As mentioned earlier, this data is very new till the time of this report and no benchmark models have been published yet. For the sole purpose, two more models have been implemented using Fast Fourier Transform and Isolation Forrest Algorithm.

5.3 Data Modelling using Fourier Transform

Fourier transformation can help to decompose the signal from the time domain to the frequency domain. In our vibrational data, converting the data into the frequency domain helps us to understand which frequency has more power and which frequencies are present in what proportions. We then cut off the frequencies which do not fall in the normal distribution of acceleration data across 3 variated data.
Using Inverse Fourier Transform (IFT), filtered data is converted back to the time domain and based on the difference between the primary data and predicted data, the error threshold is calculated. Error threshold helps to determine the anomalies same as in the case of the AE model. Above shown figure 9 shows the spectogram

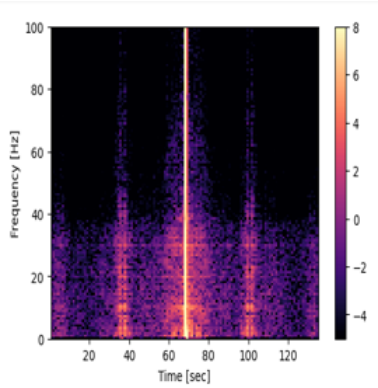

Figure 9: Spectogram of an analogous signal

of one of the faulty signal. Below mentioned steps are performed to build FFT based model:

1. Sequential data of 3 features is converted to frequency-based data using FFT Algorithm.
2. Bandpass filter is created where the frequencies under 97% confidence interval range of all 3 features, separately, are weighted as 1 and remaining are weighted as 0.
3. Bandpass filter is applied to each signal instance and the unwanted frequency which failed to fall under normal distribution are removed.

4. IFT (section 3.4) is applied on the filtered data converting the data back to time domain.
5. Error is calculated using MSE and anomalies are detected using error-thresholding.

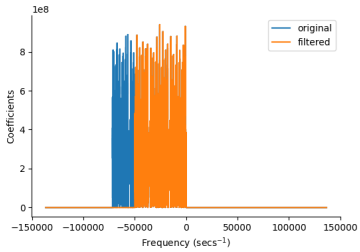

Figure 10: Signal after the removal of unwanted frequencies

Figure 10 shows one of the signal after getting the frequencies removed due to Bandpass filter.

5.4 Data Modelling using Isolation Forrest

Isolation Forrest, based on decision trees, offers the unsupervised way of determining outliers on the assumption that these outliers are very few in number and vary from each other. As mentioned in (section 3.7), this algorithm moves the anomalous data points away from the normal observations in the feature space creating shorter average depths. Until all records are isolated, they are partitioned repeatedly in a recursive manner. There are reasons that random partitioning evolves shorter paths for outliers such as lesser instances of outliers produce fewer partitions and thus shorter paths. Also, they are detected in early partitioning due to distinct feature-value.

Following steps are conducted in order to build this model :

1. Principal Component Analysis (PCA) is performed on the three features for every recording of the signal where each signal has approx. 2000 downsampled records. Thus, it gets converted to multiple recording instances of single featured sequences .
2. Second level PCA is performed on these approx 2000 samples for each recording and reduced to 50 features which would act as input for the Isolation Forrest classifier.

3. Hyperparameters are tuned for the model as follow :
   (a) contamination factor i.e proportion of the outliers in data is set to 0.1 based on our EDA,
   (b) maximum features to be extracted by estimators as 15 i.e maximum features to consider while implementing for a random split
   (c) bootstrap as False
4. After training, the model is used to predict on the test data set and anomaly score is obtained.

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

   where h(x) is the average search height or path length for 'x' from the isolation trees constructed
   E(h(x)) is the expected value of path length from collection of isolation trees(cite-isolation-paper)
   c(n) is the average search height (or depth) of any node in isolation trees used to normalize E(h(x))
   n is the number of external nodes in the Binary Search Tree.
   Average anomaly score helps to determine anomalies with negative values.

6 Model Evaluation

6.1 Analysis of Iterative AE Results

During our experiment it was noted that AE predicted different subsets of anomalies inconsistently when it was run repeatedly for few times. Increasing epochs or batch size proved insignificant on the outlier detection. As shown in the table 1, the false positives and false negative rates are used to determine some inconsistency. Also, upon verifying manually, the anomalies were precarious as their predictions were volatile. Some reasons for this are : no weight allocation to the classes initially or due to the AE neuron weight which get initialized to random. AE also tends to overfit the data which results into low bias but high variance in the predictions causing the instability in the predictions.

Iteration No.	False Positive	False Negative	ROC-AUC	F1-Score
1	3	12	0.70	0.64
2	4	17	0.68	0.63
3	7	8	0.84	0.63
4	3	9	0.83	0.63
5	6	13	0.75	0.63

Table 1: False Positives and False Negatives while running Individual AE for multiple times

As noticed in 1, the F1-Score is constant and ROC-AUC is getting better however, difference between False Positive and False Negative in each iteration changes massively.

6.2 Analysis of Ensemble of models with other models

It was noted while testing, that the F1 Score, which describes the trade-off between precision and recall is constant for most of the measurements. However, in our case errors caused by False Negatives are more undesirable, so we need to keep a look on Sensitivity and ROC-AUC for in-depth analysis.

ROC curve is used to find an optimal threshold for classification based on the use case. The predictive performance is measured between the various types of models using ROC. In our case, missing out on anomaly is more expensive than predicting normal signal as

anomaly.So predicting more false positives is better and effective.

Based on the fig 11 and 12, it can be concluded that between Fourier Transform based model and decision-tree based model, Isolation forrest performs much better to predict false positives as the area under the curve is more.

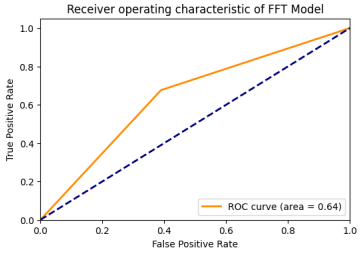

Figure 11: ROC-AUC for Fourier Transform Model

Also, isolation forrest are more robust to the outliers than FFT. In fact, Isolation Forrest at times have performed better than Individual AutoEncoder as seen in fig 13.

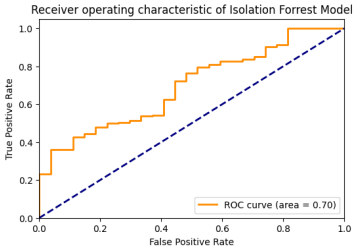

Figure 12: ROC-AUC for Isolation Forrest Model

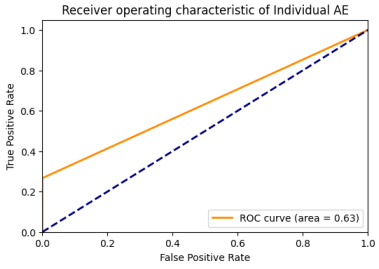

Figure 13: ROC-AUC for Individual AE

Comparing all the ROC curves, it is quite evident that area under

the ROC is better in case of Ensembled Autoencoders. It goes till 0.80 when 5 Autoencoders with differnt hyperparameters are ensembled and combined to given common prediction.

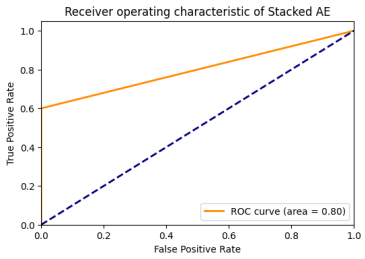

Figure 14: ROC-AUC for Stacked AE

We have also analysed the confusion matrix in order to see the detection of more false positives rather than true negatives. As

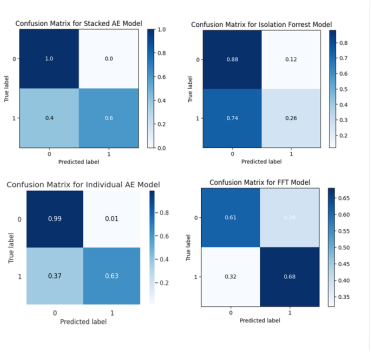

Figure 15: Comparision of Confusion Matrix between all models

shown in the 15, we get minimum recall in the stacked autoencoder which shows that model has performed much better compared to other. Also, the accuray F1-Score achieved is 0.75.
Below table compares the capabilities of all models to determine anomaly based on the latest code uploaded:

Model	ROC-AUC	F1-Score	Test Accuracy
FFT	0.64	0.19	60%
Isolation Forrest	0.65	0.15	87%
Invididual AE	0.82	0.72	96%
Stacked AE	0.86	0.85	0.97%

7 Achievements and Future Aspects

In this project we proposed a model using the stacking ensemble of Autoencoders with varying hyperparameters to detect the anomalies in the data of real-time heavy machinery. This model provided better results than the other classic ML models such as Fast Fourier Transform Model and Isolation Forrest Model, built as a part of this project with the rate of 80% to determine anomaly.

Based on experimentation, preprocessing techniques were finalized and hyperparameters were in on accordance to the accuracy of model. During the development of FFT and Isolation Forrest models, various feature engineering aspects were explored and evaluated such as Nyquist Rate, downsampling, PCA. As a part of project, it can be also concluded that CART-based algorithms also provide efficient solution if tuned properly.

Our solution is purely based on the vibrational data received by one sensor. In tool condition monitoring more than one sensors are used which can provide the ML model more insights and can predict better. However, this model does not consume memory and RAM space as much FFT Model, we could not consider it as time efficient as Isolation Forrest Model. More work can be done on optimizing the Autoencoders and reducing the training time. CNN faces the issue of overfitting and exploding gradient impact of which was seen during the training of individual AE. Upon getting 75% accuracy to determine correct faulty signals, experiments can be conducted using stacking of Isolation Tree and AE.

Conclusion

The core aim for this project was to propose an algorithm that would predict the faulty signals generated in CNC Milling machine using the acceleration data. For the given data, which does not have reference model, the proposed algorithm performed better than the other two popular and classic models. Although, the accuracy achieved is good, but can be increased more using more computational resources and time as not all faulty data is getting predicted. Since the end-to-end business could rely on this analysis, more models and types of ensembles should be experimented with. We would not call this model as benchmark, still the insights obtained during research can be used further.

References

[1] Chandra Nath. Integrated tool condition monitoring systems and their ap- plications: a comprehensive review. Procedia Manufacturing, 48:852–863, 2020

[2] A. Agogino and K. Goebel. Best lab, uc berkeley. milling data set, nasaames prognostics data repository. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/, 2007.

[3] Daniel Frank Hesser and Bernd Markert. Tool wear monitoring of a retrofitted cnc milling machine using artificial neural networks. Manu- facturing letters, 19:1–4, 2019.

[4] Mohamed-Ali Tnani, Michael Feil, Klaus Diepold (2021) Smart Data Collection System for Brownfield CNC Milling Machines: A New Benchmark Dataset for Data-Driven Machine Monitoring , Procedia CIRP 131-136

[5] P. Tavner and J. Penman, Condition Monitoring of Electrical Machines. England: John Wiley & Sons Inc, 1987

[6] System level Manufacturing and Automation Research Testbed (SMART) at the University of Michigan. Cnc milling dataset. https://www. kaggle.com/shasun/tool-wear-detection-in-cnc-mill, 2018.

[7] Liu, Y., Zhang, J., Hu, X. et al. (2022) Sensor data anomaly detection and correction for improving the life prediction of cutting tools in the slot milling process. Int J Adv Manuf Technol 119, 463–475

[8] Lin X, Bo Z, Zhu L (2017) Sequential spindle current-based tool condition monitoring with support vector classifier for milling process. Int J Adv Manuf Technol, 1–10

[9] Harun MHS, Ghazali MF, Yusoff AR (2017) Analysis of tri-axial force and vibration sensors for detection of failure criterion in deep twist drilling process. Int J Adv Manuf Technol 89(9–12):3535– 3545

[10] Feng Z, Liang M, Chu F (2013) Recent advances in time–frequency analysis methods for machinery fault diagnosis: a

review with application examples. Mech Syst Signal Process 38(1):165–205

[11] Elgargni M, Al-Habaibeh A, Lotfi A (2015) Cutting tool tracking and recognition based on infrared and visual imaging systems using principal component analysis (PCA) and discrete wavelet transform (DWT) combined with neural networks. Int J Adv Manuf Technol 77(9–12):1965–1978

[12] Jin X, Zhao M, Chow TWS, Pecht M (2014) Motor bearing fault diagnosis using trace ratio linear discriminant analysis. IEEE Trans Ind Electron 61(5):2441–2451

[13] Zhao R, Yan R, Wang J, Mao K (2017) Learning to monitor machine health with convolutional bi-directional LSTM networks. Sensors 17(2):273

[14] Chen ZQ, Li C, Sanchez R-V (2015) Gearbox fault identification and classification with convolutional neural networks. Shock Vib, 2015

[15] Book : Machine Learning , (McGraw-Hill International Editions Computer Science Series) by Tom Mitchell

[16] Chandola, V. Banerjee, A. Kumar, V. (2009) Anomaly detection: A survey. ACM Computing Surveys. 41 (3): 1–58.

[17] Mayu Sakurada,Takehisa Yairi (2014) Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction.MLSDA'14: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis :4-11

[18] Jinwon An, Sungzoon Cho (2015) Variational Autoencoder based Anomaly Detection using Reconstruction Probability. SNU Data Mining

[19] B. Zhao, H. Lu, S. Chen, J. Liu and D. Wu, "Convolutional neural networks for time series classification," in Journal of Systems Engineering and Electronics, vol. 28, no. 1, pp. 162-169, Feb. 2017, doi: 10.21629/JSEE.2017.01.18.

[20] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," in IEEE Potentials, vol. 13, no. 4, pp. 27-31, Oct.-Nov. 1994, doi: 10.1109/45.329294.

APPENDICES

Code related files for the project are hosted on the github server of the University of Birmingham at the following URL: https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2021/kxr160. Most of the code is self-written except the code for Stacking generalization which is heavily inspired from the code written Json Brownlee on the website : https://machinelearningmastery.com/stacking-ensemble-for-deep-learning-neural-networks/ .

There are three Jupyter notebook referring to three models build and elaborated in this report. One can direclty run any three of the notebooks :
Model_1_Stacking_Ensemble_AE.ipynb.ipynb
Model_2_Fast_Fourier_Transform.ipynb
Model_3_Isolation_Forrest.ipynb .