POLITECNICO DI TORINO

…

CHANGEPOINT DETECTION WITH PYTHON ON SENSOR DATA IN WELDING PROCESSES

SUPERVISORS

Prof. LOMBARDI FRANCO

Prof. BRUNO GIULIA

TRAINI EMILIANO

CANDIDATE

BALAJI JAYACHANDRAN

Academic Year 2020 - 2021

ACKNOWLEDGMENTS

At the end of this long journey, I would like to spend a few words for all those who have always been there, without hesitation, both in times of difficulty and in happy and carefree ones.

First of all, a heartfelt thanks to my supervisor, **Prof. BRUNO GIULIA**, who has been an ideal teacher, mentor, and thesis supervisor, for offering her availability to discuss in all situations and for the precious advice she gave me throughout the induction and continuation of this work. Furthermore, a particular thanks to **Prof. LOMBARDI FRANCO** and for his availability. An extensive thanks to **Mr. TRAINI EMILIANO**, research fellow, for his suggestions, monitoring, and guidance throughout the work and his patience in reviewing the results and validating them. I'm proud of, and grateful for, my time working with all.

Also to my parents, for the help, encouragement, and support they gave me and for believing in me during all my years of study.

Finally, a final thought, but no less important, goes to all my friends, to the historical ones and to the new ones I met in recent years, with whom I have shared the joys and sorrows of this journey, but also many pleasant moments outside of the Politecnico.

Thank you all!

ABSTRACT

CHANGEPOINT DETECTION WITH PYTHON ON SENSOR DATA IN WELDING PROCESSES

In a machining process, there are several stages of operation involved. The sensors are used to collect the data in all the stages at the desired intervals of time of the user. Once there is a large amount of data set, finding the stages becomes complex. The stages can be identified by the Changepoint, but the identification of Changepoint is a challenge. Changepoint detection is the estimation of breakpoints at which the statistical properties of the observation change. This Changepoint gives the inference that the stage has started/ended.

Pruned Exact Linear Time (PELT) is the method adopted for detecting changepoints, which is through minimizing the cost function over possible numbers and locations of changepoints. This method performs efficiently compared to the other existing methods. To understand it better in the simulation study a comparison of PELT and dynamic programming is also made to validate the efficiency.

This thesis aims to present a framework to estimate the Changepoints and segment the process using statistical properties of the observations made. The first part of the framework identifies the changepoints and segment the processes, while the second part deals with the development of different types of algorithms and compared to find the best result. The operation is carried out in Colabatory, an IDE for Python released by Google to enable machine learning with storage on the cloud.

Ruptures, a python library for offline change point detection has been used, its algorithm exact and approximate detection for various parametric and non-parametric models. Ruptures offer ways for analysing and segmenting non-stationary data..

To validate the algorithm, the raw data set of a milling machine has been used. After the validation, it is aimed to be incorporated in the Machine monitoring of the Welding machine.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS **IV**

ABSTRACT **V**

TABLE OF CONTENTS **VI**

LIST OF TABLES **VIII**

LIST OF FIGURES **IX**

1. INTRODUCTION **1**

1.1 INTERNET OF THINGS 1

1.2 WELDING 1

1.2.1 Monitoring the welding process *2*

1.2.2 Measuring Voltage and Current. *3*

1.2.3 Wire feed rate *3*

1.3 SETUP 3

1.4 STRUCTURE OF THE THESIS 4

2. LITERATURE REVIEW **7**

2.1 HISTORY OF CHANGEPOINT DETECTION AND ITS APPLICATION 7

2.1.1 Edge Computing *8*

3. MATHEMATICAL MODELLING AND PYTHON PACKAGE **10**

3.1 PELT 10

3.1.1 Pelt method *10*

3.1.2 POWER OF PELT ALGORITHM *12*

3.1.3 PELT ALGORITHM *14*

3.2 RUPTURES 14

4. CASE STUDY **15**

4.1 FLOW CHART OF THE STUDY 15

4.1.1 Monitoring *16*

4.1.2 Changepoint detection *16*

4.1.3 Feature Extraction *17*

4.2 DATASET 17

4.3 METHODS FOR DETECTING CHANGEPOINT 19

4.3.1 PELT *19*

4.3.2 Dynamic programming *20*

4.4 SEGMENTATION OF THE PROCESSES 21

4.5 FETCHING THE PROCESS 22

4.6 IMPLEMENTATION 23

4.6.1 Interpreting each row *23*
4.6.2 Compiling whole data *25*

5. RESULTS AND CONCLUSION **26**

5.1 DISCUSSION FROM THE LITERATURE 26
5.2 RESULT FROM THE CASE STUDY 27
5.2.1 Comparisons between PELT and Dynamic processing *27*
5.3 THE RESULT FROM THE OTHER SENSORS DATA 37
5.3.1 Direct Current *37*
5.3.2 Alternate current *39*
5.3.3 Vibration of table *41*
5.3.4 Inference *42*
5.4 IMPLICATIONS AND RECOMMENDATIONS 42
5.4.1 Verdict between the PELT and Dynamic processing *42*
5.4.2 Verdict between the Interpretation and compiling *43*
5.5 LIMITATIONS AND FUTURE RESEARCH 44
5.5.1 Cleaning the data *44*
5.5.2 Using the historical values *44*
5.5.3 Using sampling *45*
5.6 CONCLUSION 45

BIBLIOGRAPHY **46**

LIST OF TABLES

Table Page

TABLE 4. 1 SAMPLE OF THE MILLING MACHINE DATA 17

LIST OF FIGURES

Figure Page

FIGURE 1. 1 SETUP OF THE SENSOR-BASED MONITORING SYSTEM 4

FIGURE 3. 1 CHANGEPOINT DETECTION METHOD 10

FIGURE 4. 1 FLOWCHART OF THE CASE STUDY 15
FIGURE 4. 2 GENERAL EXAMPLE OF CHANGEPOINT DETECTIONS 16
FIGURE 4. 3 RAW INPUT OF ROW16 FIGURE 4. 4 RAW INPUT OF ROW17 18
FIGURE 4. 5 RAW INPUT OF ROW 35 FIGURE 4. 6 RAW INPUT OF ROW108 18
FIGURE 4. 7 RAW INPUT OF ROW 18 FIGURE 4. 8 RAW INPUT OF ROW 25 18
FIGURE 4. 9 RAW INPUT OF ROW 5 19
FIGURE 4. 10 CHANGEPOINT DETECTIONS OF ROW 5 BY PELT 20
FIGURE 4. 11 CHANGEPOINT DETECTIONS OF ROW 5 BY DYP 21

FIGURE 5. 1 RAW INPUT OF ROW 2 27
FIGURE 5. 2 CHANGEPOINT DETECTIONS ON ROW 2 BY PELT 28
FIGURE 5. 3 STATIONARY PROCESS ON ROW 2 BY PELT 28
FIGURE 5. 4 CHANGEPOINT DETECTIONS ON ROW 2 BY DYP 29
FIGURE 5. 5 STATIONARY PROCESS ON ROW 2 BY DYP 29
FIGURE 5. 6 RAW INPUT OF ROW 9 30
FIGURE 5. 7 CHANGEPOINT DETECTIONS ON ROW 9 BY PELT 30
FIGURE 5. 8 STATIONARY PROCESS ON ROW 9 BY PELT 31
FIGURE 5. 9 CHANGEPOINT DETECTIONS ON ROW 9 BY DYP 31
FIGURE 5. 10 STATIONARY PROCESS ON ROW 9 BY DYP 31
FIGURE 5. 11 RAW INPUT OF ROW 33 32
FIGURE 5. 12 CHANGEPOINT DETECTIONS ON ROW 33 BY PELT 32
FIGURE 5. 13 STATIONARY PROCESS ON ROW 33 BY PELT 32
FIGURE 5. 14 CHANGEPOINT DETECTIONS ON ROW 33 BY DYP 33
FIGURE 5. 15 STATIONARY PROCESS ON ROW 33 BY DYP 33
FIGURE 5. 16 RAW INPUT OF ROW 117 34
FIGURE 5. 17 CHANGEPOINT DETECTIONS ON ROW 117 BY PELT 34
FIGURE 5. 18 STATIONARY PROCESS ON ROW 117 BY PELT 34
FIGURE 5. 19 CHANGEPOINT DETECTIONS ON ROW 117 BY DYP 35
FIGURE 5. 20 STATIONARY PROCESS ON ROW 117 BY DYP 35
FIGURE 5. 21 RAW INPUT OF ROW 118 35
FIGURE 5. 22 CHANGEPOINT DETECTIONS ON ROW 118 BY PELT 36
FIGURE 5. 23 STATIONARY PROCESS OF ROW 118 BY PELT 36
FIGURE 5. 24 CHANGEPOINT DETECTIONS OF ROW 118 BY DYP 36

FIGURE 5. 25 STATIONARY PROCESS OF ROW 118 BY DYP 37
FIGURE 5. 26 RAW INPUT OF ROW 36 OF DC SENSOR DATA 38
FIGURE 5. 27 CHANGEPOINT DETECTIONS OF ROW 36 BY PELT 38
FIGURE 5. 28 STATIONARY PROCESSES OF ROW 36 BY PELT 38
FIGURE 5. 29 CHANGEPOINT DETECTIONS OF ROW 36 BY DYP 39
FIGURE 5. 30 STATIONARY PROCESSES OF ROW 36 BY DYP 39
FIGURE 5. 31 RAW INPUT OF ROW 18 OF DC SENSOR DATA 40
FIGURE 5. 32 CHANGEPOINT DETECTIONS OF ROW 18 BY PELT 40
FIGURE 5. 33 STATIONARY PROCESS OF ROW 18 BY PELT 40
FIGURE 5. 34 RAW INPUT OF ROW 158 OF SENSOR DATA OF VIBRATION OF TABLE 41
FIGURE 5. 35 CHANGEPOINT DETECTIONS OF ROW 158 BY PELT 41
FIGURE 5. 36 STATIONARY PROCESS OF ROW 158 BY PELT 41
FIGURE 5. 37 RAW INPUT OF ROW 18 44

1. INTRODUCTION

1.1 Internet of Things

Internet of things is an interaction between the physical and digital world using sensors and actuators. The devices and appliances are connected to a network and are used collaboratively to achieve complex tasks that require a high degree of intelligence. IoT got inducted into our daily lives in many ways like smart homes, smart transportation, and smart city. The development of IoT has enabled the appliances to sense, interconnect and robotically communicate and adaptively.

In industrial processes, monitoring has great importance to reduce downtime, increase efficiency, reduce waste and make informed decisions. The advancement of the Internet of things on industrial machines helps to collect real-time data from various types of sensors (e.g., Potentiometer, Vibration, Position, Pressure) during the process execution. The information collected from these sensors is still a big challenge to handle and process.

1.2 Welding

Welding is the technique of connecting metal components together by melting the surfaces of the parts[1]. Arc welding, friction stir welding, gas welding, energy beam welding, and solid-state welding are some of the different forms of welding available[2].

In the arc welding process, an arc is produced by passing electricity between an electrode and metal parts to be welded. The heat produced by the arc melts the metal to form molten metal. This molted metal is referred as weld pool. This pool of molten metal forms the joint of the two metal parts upon cooling.

The weld quality is critical in many applications such as aerospace where it ensures the safety of people. Establishing continual monitoring of the physical phenomena occurring inside the various areas of the arc, and controlling parameters like heat input, penetration, weld form, and heat-affected zone, is an efficient means of measuring the quality of welded components (HAZ).

1.2.1 Monitoring the welding process

For real-time monitoring of welding signatures, an arcwelding test-bed was created. A collection of hardware and sensors were chosen and incorporated into the welding system using multi-criteria decision-making (MCDM) methods [3] to build this test-bed.

The different parameters which influence the quality of arc weld are

- Current (A) [4],
- Voltage (V) [5,6],
- Speed of torch (in/min or m/min) [7],
- gas flow rate (L/min) [8],
- wire feed rate (in/min or m/min) [9,10]

Sensors are used to measure different performance characteristics such as pressure, temperature, vibration, power consumption, and so on. A sensor is a device that receives a signal or stimulus and responds to it. A sensor's job is to transform a physical phenomenon into a quantifiable electrical signal, such as current, voltage, or resistance, that fluctuates over time.[11].

Different sensors are used to obtain data on weld quality. Sensors are mostly utilized in welding applications to verify process parameters at the joint and component location. Sensors can be utilized to capture needed signals for quality control during welding, pre-welding, and post-welding situations.[12]

When it comes to electrical data, the arc voltage and welding current are the two most important metrics for detecting welding flaws. These factors can give you a decent idea of how well the welding is going and whether or not there are any defects.[13]. However, just analysing the electrical signals from the welding power source is insufficient to accurately determine the factors that impact the welding. As a result, additional sensors are necessary for weld condition monitoring.[14]

1.2.2 Measuring Voltage and Current.

The voltage measurement in arc welding should be done as near to the welding arc as feasible. At the contact tube, the current is delivered to the wire, and this position is believed to be a good measurement point. Between the contact tube and the wire tip, however, there is a voltage drop.

The voltage between the workpiece and the wire at the feed rollers is measured for more accurate and dependable monitoring of arc voltage, especially in a production scenario. The wire acts as a perfect conductor, preventing voltage drop across the welding cables. The welding current is constant throughout the welding current circuit and is quite high, preventing it from passing through the instrument directly. Other methods for measuring welding current include:
*Hall Effect sensors, which work by measuring the magnetic field around a conductor, are employed.

1.2.3 Wire feed rate

In order to produce a reliable welding process, the wire feed rate is a critical parameter that must be monitored and managed [15]. Because an increase in wire feed rate can lead to an increase in current, differences in welding wire feed rate also correspond to variations in current.

Welding power sources are often adjusted to produce a steady voltage and current. In welding, however, the most typical strategy is to keep the voltage and wire feed rate constant while altering the current. As a result, by regulating the wire feed rate at a nominal value, a stable welding process can be obtained. The wire feeder is installed on the robot arm in robot welding systems. Custom-made solutions or managing the wire feeder unit's drive wheel can be used to determine the wire feed rate. Sensors with idler rollers that give a signal proportional to the wire feed rate are also available [16].

1.3 Setup

The data from the machine are recorded and used to process to monitor the machine. The sensors record the data in certain desired intervals. The data gathered are high in

number to handle and are very raw to handle. So these data must be processed and sent for data collection. It is very important to get the data of the machine when it is running stationarily or working process. So, finding this process is critically important so that the data collection is not bombarded with a high amount of data and avoid wastage of time. After the prerequisite of cleaning the data, the process has to be segmented using the changepoint detection using a mini-computer closer to the machine and then transmit the relevant process to the cloud or desired location for further investigation. This can be possible through the advancement of the edge computing technologies

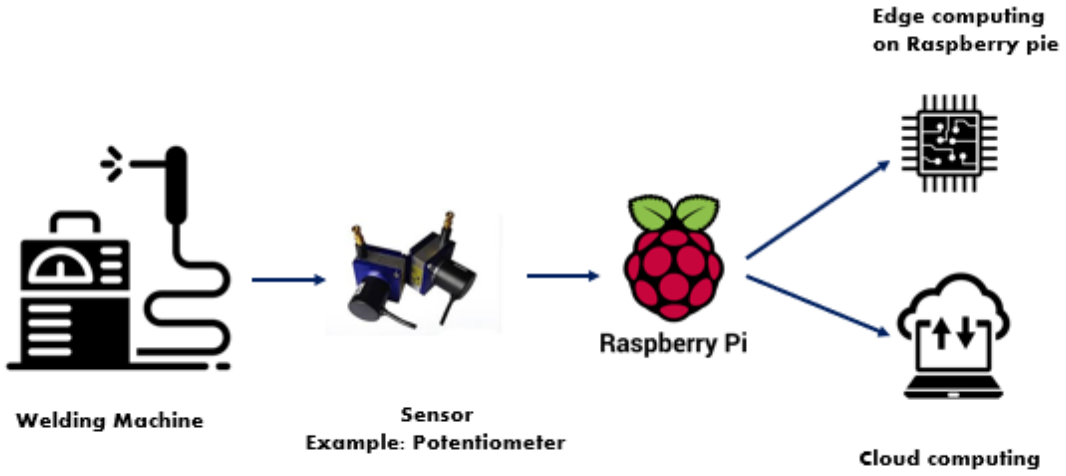

Figure 1. 1 Setup of the Sensor-based Monitoring system

1.4 Structure of the thesis

This research consists of two parts, a theoretical and an experimental one. The theoretical part firstly introduces the relevant concepts to understand the Internet of things, Welding processes, Changepoint detection, Edge computing, Mathematical modelling.

The topics that will be analysed are machine monitoring and their trends, the structure of the modelling methods, the characteristics of the changepoint detection methods, the difference of the changepoint detection, and finally a case study with a dataset of a

milling machine. Subsequently, a literature review of changepoint detection in every field, edge computing and their and welding methods and relevant parameters to be monitored will be introduced. This chapter aims to get a sufficient understanding and overview of the current status of the changepoint detection methods implemented. Subsequently, a short paragraph focuses on packages available for programming in python. The findings and conclusion of this theoretical part are used as a starting point of the subsequent analysis.

The third chapter is about mathematical modelling. Firstly, some existing research will be used to identify and know more about the statistical analysis done. It will completely speak about the PELT method which is very relevant for this thesis.

In the fourth chapter, a case study is to be introduced to check the effectiveness of the algorithm developed. The dataset of a milling machine is extrapolated from ", NASA Ames Prognostics Data Repository[34]. Then, a meticulous collection of data will be carried out on the different types of sensors and the data from the relevant sensor which has been chosen. Following this, the algorithm developed for PELT and Dynamic programming will be analyzed intensively on every data available for the particular sensor. The changepoint detections, segmentation of the processes, and feature extraction of the processes are to be vividly explained.

Then, the subsequent chapter shows some descriptive results which will allow describing and understanding the features of the dataset and algorithm. The analyses that will be carried out, firstly by making some explorative findings and then with a more precise approach by comparing the results from both the algorithm. Additionally, the algorithm will be checked with the other sensor data as well to find the performance relevant to different kinds of data. while a deeper study will focus only on the vibration of spindle sensor data.

However, as with any research, this thesis has limitations, but these main shortcomings could be used as inputs for future research. In the next section, the limitations from the thesis will be discussed and the relevant future works to be done are also recommended concerning the results.

Finally, the last section points out the conclusion of the work. The first paragraph will discuss the findings concerning the issues arising from the literature review. Then, based on previous research and through the outcomes of this thesis are to be discussed, a few actions that could help to this work's relevance in other machines and monitoring methods.

2. LITERATURE REVIEW

2.1 History of Changepoint detection and its application

In recent years, changepoint detection has received a lot of attention and studied widely in different streams of studies. Changepoint detection has become a key setup in machine monitoring for the real-time analysis of the machines. The initial research on change-point detection dates from the 1950s [17]: the problem addressed in this paper is concerned with the identifications of the subsamples and the detection of changes in the parameter value. The initial research was focused on detecting the change in mean. This issue has been intensively researched ever since, and in-depth monographs have been published on it consistently.

There are two main branches of changepoint detection methods offline and online. The offline method is functional when all the samples are collected. It detects the Changepoints offline as the name suggests. It is also called signal segmentation. The online method is used to detect the changepoints in real-time when they occur. It is also called event detection [27].
Also, there are two kinds of change point detection, single changepoint method, and multiple change point detection. Single changepoint is useful on many occasions, even if many changepoints exist in the sample, single changepoint detection can be used. For example: to find the anomaly and highlight the flaw in a sample even if there is a single changepoint found [24]. Multiple changepoints are used for different other requirements such as segmentation, anomaly detection.

Changepoint detection has many different uses and has been adopted in many real-world scenarios. It has been studied in the financial analysis [18]: this paper discusses the Adaptive detection of multiple change-points in asset price volatility, [19]: examines the multiscale change point estimator, [20]: also deals the similar multiple structural changes.

Bioinformatics has much research extensively on changepoint detection widely. The papers [21]: advocate the uses of different kinds of mathematical modeling such as FPOP, dynamic programming,[22]: talk about the Gaussian model with piecewise

constant variance or global variance,[23]: a new method based on Seasonal-Trend decomposition with locally weighted regression (Loess) and probability ratio statistics to sense the changes.
Climatology also has some intensive research on the same topic of changepoint detection [24]: gives the analysis of complex changepoint problems and methods. [25]: A generic change detection approach is proposed for time series by detecting trend and seasonal changes. [26] In climate data series this work specifies, classifies, and equates different methods to detect undocumented changepoints.

2.1.1 Edge Computing

P. Varshney and Y. Simmhan [28] mentions that there is an increasing requirement for sensing a large amount of data. This increase is due to the Internet of Things (IoT), commodity sensors, accessible communications, and the need for intelligent infrastructure management. These IoT data are processed in the cloud in a large data center. The problem with cloud computing is round trip latency time. The IoT applications like voice response in Siri and demand prediction of power grids are latency-sensitive and their performance can be compromised by cloud decision making. This increasing amount of data from the Edge led to the supplement or replace the cloud computing. Some of the processing and analytics are performed in Edge devices while the cloud is used for data archival and coordination.

L. F. Bittencourt [29] suggests that there is increased efficiency in moving data storage and computing to the edges of a network, but the central cloud is necessary to coordinate all the edge activities. M. S. D. Brito, S. Hoque, R. Steinke, and A. Willner [30] mentions that the fourth industrial revolution is the Industrial Internet. Brito describes those developments in the semiconductor industry that led to a reduction in the cost, size, and energy consumption of the chipset. These developments make the embedment of more sensors cheaper and easier.

Lin and J. Yang [31] mentions that intelligent logistics centers have clouds that are located and managed in a centralized manner. Whereas a factory is very large so there are a large number of requests from the IoT sensors, there will be a large latency period in responding to these requests from the cloud. Some of the computing tasks which can

be completed by local information are performed by the edge resources. So, they can be completed immediately.

2.1.1.1 Edge computing applications

The edge computing has been used in many different applications since the advent of Internet of things. The industry 4.0 gave rise to smart factories.

2.1.1.1.1 Smart factories

The Industrial Internet of things helps in higher productivity in manufacturing industries. Real-time access and reliability are very important in these smart factories which cannot be realized by cloud computing alone leading to the incorporation of fog and edge computing.

2.1.1.1.2 Smart agriculture

The advancements in the agricultural industry led to the collection of large amounts of data. These data are not used efficiently to improve farm-based decision-making and management.
Smart machines and sensors can be incorporated into the farming process. Farming will become more data-driven and data-enabled. Smart farming is based on location, situation awareness, and real-time events. This technology helps in giving real-time assistance.

3. MATHEMATICAL MODELLING AND PYTHON PACKAGE

3.1 PELT

The PELT algorithm utilizes the minimization of costs to find the changepoints. It is a very standard approach for finding it. To find multiple change points, the PELT algorithm is first applied to the whole data set and iteratively and independently to each partition until no further change points are detected.

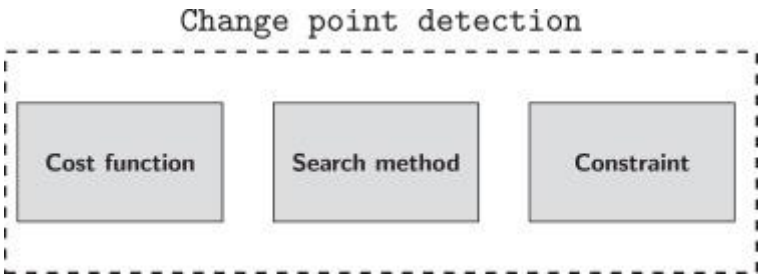

Figure 3. 1 Changepoint detection method

The key assumption of the PELT algorithm is that there is an increase in the number of changepoints linearly with the increase of the data set, that is, it assumes that throughout the data sets the change points are spread and they are not restricted to one portion of the data. Under mild conditions when the computational cost is linear in the number of data points the PELT algorithm is exact. When it is compared to approximate search methods and faster compared to other exact search methods PELT is more accurate.

3.1.1 Pelt method

One of the most common ways to detect the multiple changepoints is to minimize:

$$\sum_{i=1}^{m+1}[C(y_{\tau_{i-1}+1},\dots,y_{\tau_i})] + \beta\, f(m) \qquad \text{.....(1)}$$

- C is a cost function for a segment
- β f (m) is a penalty to guard against overfitting.

The most widely used cost function in the changepoint literature is Twice the negative log-likelihood, The other cost functions used are quadratic loss and cumulative sums based on both the segment log-likelihood and the length of the segment.

Considering the choice of penalty the frequently used is one which is linear in the number of changepoints, that is β f (m) = β m. There are Akaike Information Criterion (AIC) (β= 2p) and Schwarz Information Criterion (SIC, also known as BIC) (β = p log n) which are examples of such penalties, where p is the number of additional parameters introduced by adding a changepoint. The PELT method is designed for such linear cost functions

From the basis of the algorithm of [37], instead involves a pruning step within the dynamic program. To minimize, The search method is proposed by Jackson, et al.[37]

$$\sum_{i=1}^{m+1}[C(y_{\tau_{i-1}+1},\dots,y_{\tau_i})+\beta] \qquad \text{.....(2)}$$

- C is a cost function for the i^{th} segment
- β is a penalty to guard against overfitting.

Equation 2 is equivalent to equation 1 where f (m) = m.

By pruning, the PELT technique modifies [37] the optimal partitioning method. It combines optimal partitioning and pruning to obtain accurate and efficient computing cost in n. The best segmentation is F (n), where

$$F(n)=\min_{\tau}\{\sum_{i=1}^{m+1}[C(y_{\tau_{i-1}+1},\dots,y_{\tau_i})+\beta]\} \qquad \text{.....(3)}$$

Conditioning on the last point of change, τ_m and calculating the optimal segmentation of the data up to that changepoint gives,

$$F(n)=\min_{\tau_m}\{\min_{\tau|\tau_m}\sum_{i=1}^{m}[C(y_{\tau_{i-1}+1},\dots,y_{\tau_i})+\beta]+C(y_{\tau_m+1},\dots,y_n)\} \qquad \text{.....(4)}$$

This may also be done for the second to last, third to last, and so on changepoints. The cyclical character of this conditioning is highlighted by the fact that the inner minimisation is similar to equation 3. In reality, the inner minimisation equals F (τ_m), and so equation 3 may be rewritten as

$$F(n)=\min_{\tau_m}\{F(\tau_m)+C(y_{\tau_m+1},\dots,y_n)\} \qquad \text{.....(5)}$$

3.1.1.1 Finding the Changepoints

- **STEP 1:** Start by calculating F (1) and then repeatedly calculate F (2), …, till F(n).
- **STEP 2:** The optimal segmentation up to τ_{m+1} in every step, is stored.
- **STEP 3:** Finally, when F (n) is reached the optimal segmentation for the entire data has been identified and the number and location of changepoints have been recorded.

In every step, the minimisation over τ_m replaces all the predecessor values e.g. when calculating F(4) the minimisation covers τ_m= 0, 1, 2, 3. PELT method achieved its computational efficiency by removing candidate values of τ_m from the minimisation at each step. The essence of pruning in this context is to remove those values of τ which can never be minima from the minimisation performed at each iteration.

3.1.2 POWER OF PELT ALGORITHM

The study will find out the power of the PELT algorithm for the finite sample size for specific alternatives of one changepoint.
The problem for the changepoint hypothesis will be stated as:
H_0: No changepoint in the data.
H_1: There is a changepoint in the data.
The likelihood ratio statistic is:

$$Q_n = \max_{1<k\leq n-1}(-2\log_k) \qquad \ldots\ldots\ldots\ldots(6)$$

Where,

- $k = \frac{L(\hat{\theta}_0)}{L(\hat{\theta}_1)}$, is the ratio of the probabilities of the sample before and after the change.
- K is the changepoint. K is not fixed, and the location of K is unknown.
- Q_n is a growing function of $\max_{1<k\leq n-1} \frac{1}{k}$

H_0 is rejected if,

$$Q_n = \max_{1<k\leq n-1}(-2\log_k) > R \qquad \ldots\ldots\ldots\ldots(7)$$

Where,

N= size of the sample

R= some bound that depends on the level of significance.

Depending on the size of the test, *R* grows asymptotically as *n* for a given *x* so that,

$$Q_n = \frac{(x+f(logn))^2}{a^2(logn)} \approx 2logn \qquad \ldots\ldots\ldots\ldots(8)$$

whether there is change, then it occurs at a certain point in the data. Thus for a changepoint *k*, $1 \leq k \leq n-1$ and as n$\rightarrow \infty$, then we have that *k*, $n-k \rightarrow \infty, \frac{k}{n} \rightarrow \epsilon(0,1)$. Therefore, this test is consistent, since for a given size α the power of the test converges to 1.

The test rejects the null hypothesis if $Q_n^{0.5}$ > R, where R is the asymptotic critical value which depends on the size of α and the sample size n. For a given level α, the power of the test is the probability of accepting this alternative correctly, that is,

$$(\alpha) = \mathrm{P}(Q_n^{0.5} > R|\mathrm{H1}) \qquad \ldots\ldots\ldots\ldots(9)$$

Since the distribution of $Q_n^{0.5}$ under H_1 is not known, simulations will be used to estimate the power of the test. For a sample size n, 1000 replicates will be made and each replicate $Q_n^{0.5}$ will be estimated. The power function for a given level α will be estimated as,

$$(\alpha) = \frac{1+\#[Q_n^{0.5} > (R_n)]}{1+N} \qquad \ldots\ldots\ldots\ldots(10)$$

where $\#(Q_n^{0.5} > R_n(\alpha)$ denotes the number of times $Q_n^{0.5} > R_n(\alpha)$.

The power of the test at each changepoint position was assessed for a specific sample. The changepoint k was put at = 20, 40, 60, 80, 100, 120, 140, 160, 180 for a sample of size n=200. At each changepoint position, 1000 simulations will be run. The value of the test statistic will be determined in each of 1000 runs. u was calculated using the crucial values R1 and R2.

3.1.3 PELT ALGORITHM

```
# change point detection
model = "l1"  # "l2", "rbf"
algo = rpt.Pelt(model=model, min_size=3, jump=5).fit(mys
ignal)
my_bkps = algo.predict(pen=10)

# show results
fig, (ax,) = rpt.display(mysignal, my_bkps, my_bkps, fig
size=(10, 6))
plt.show()
```

3.2 Ruptures

Ruptures[27], a python library for offline change point detection has been used, its algorithm exact and approximate detection for various parametric and non-parametric models. Ruptures allow to analyse and segment the non-stationary data. It is the most widespread change point detection library available. It can be used along with other libraries such as pandas, matplotlib.

4. CASE STUDY

4.1 Flow chart of the study

The milling machine is monitored through the sensors and then the raw data is collected. After the raw data is collected, it is processed by a processor closer to the machine by edge computing to detect the changepoints through desired algorithms. After the changepoints are detected, the processes are segmented. Once the features are segmented, they are gathered into the cloud or anywhere preferred by the user. The programming operation is carried out in Colabatory, an IDE for Python released by Google to enable machine learning with storage on the cloud.

The below flowchart gives the overall idea of the processes to be carried.

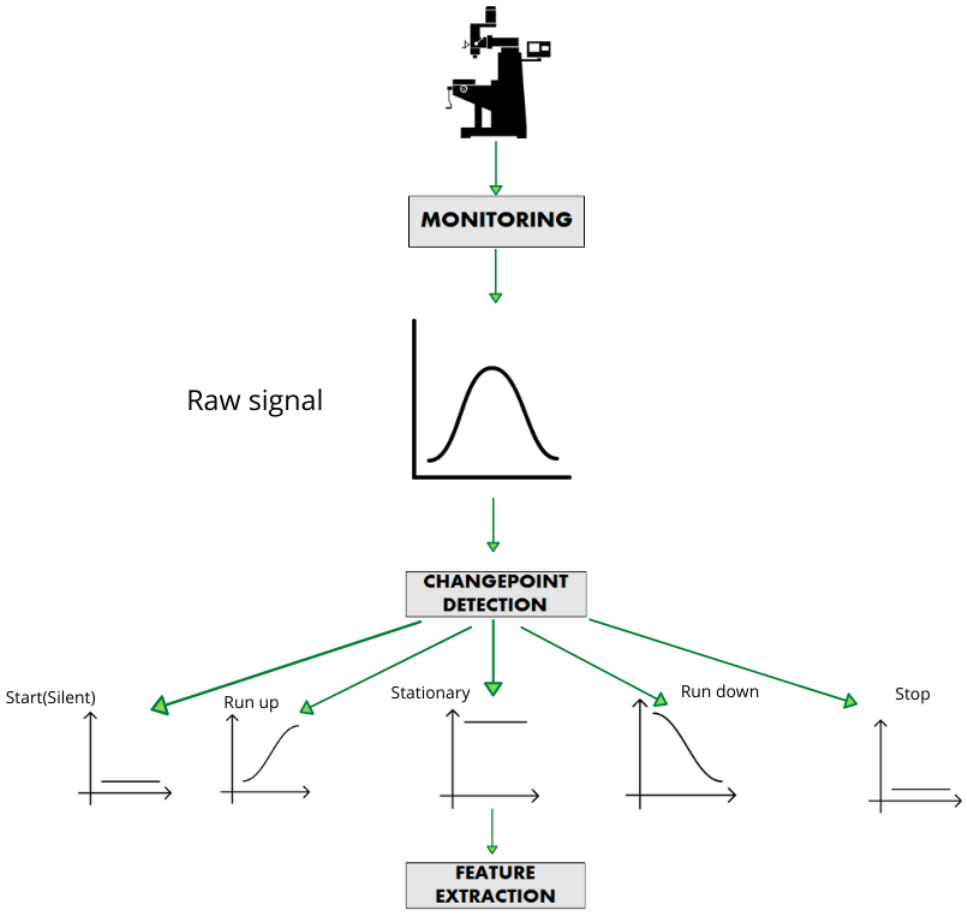

Figure 4. 1 Flowchart of the case study

4.1.1 Monitoring

The sensors installed on the welding machine measures the physical input and then convert it into an electrical signal which can be processed future. Most of the IoT sensors determine changes in electrical properties such as resistance, capacitance, and inductance. They are converted as relevant data which is a raw signal, and it is read by a microprocessor.

4.1.2 Changepoint detection

The raw data is then processed in the microprocessor using several mathematical modeling to determine the changepoint. The modeling used in the following case is PERT and Dynamic Programming. In this stage, the exact breakpoints are identified. The number of these breakpoints is either known or unknown.

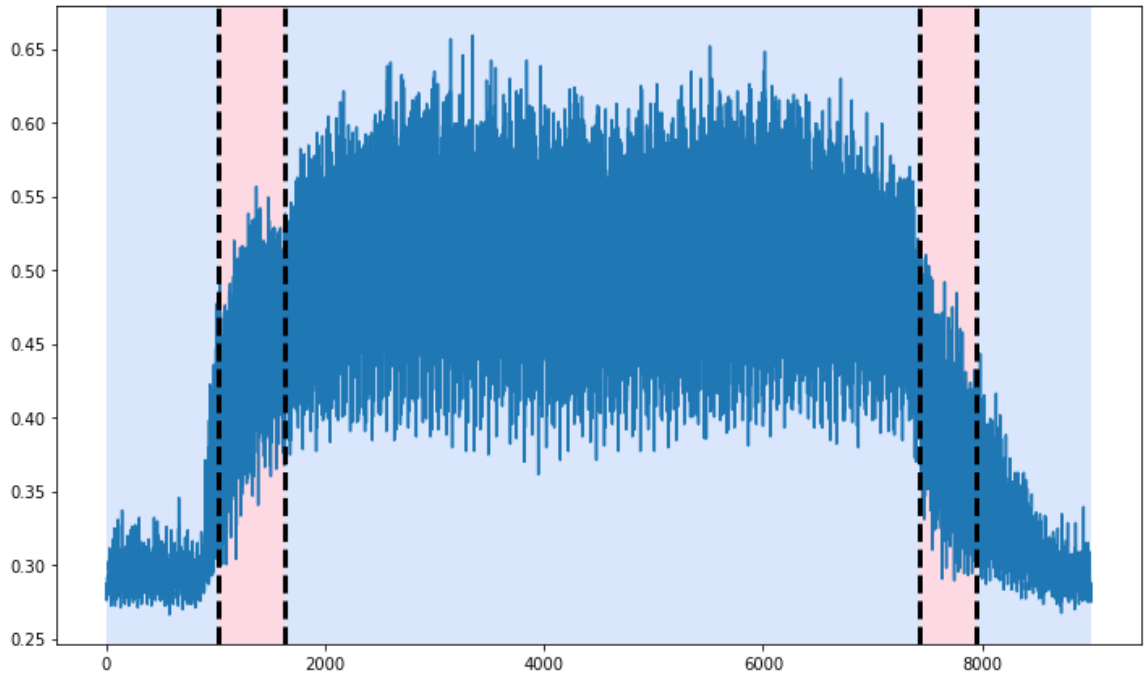

Figure 4. 2 General example of Changepoint detections

The number of breakpoints identified here is.

(0)0, (1)1020, (2)1635, (3) 7425, (4) 7945 (5) 9000.

4.1.3 Feature Extraction

When the breakpoints are known then the processes or stages can be segmented. When all the processes are identified, then the stationary process is found by the variation in the mean of the results and then it can be plotted accordingly. These results can be stored in the desired location either in the cloud or locally.

4.2 Dataset

To validate the algorithm before implementing. A dataset is used to examine and derive the results. The dataset used is from the NASA ames Prognostics Data Repository [34]. They collect the data sets from universities, agencies, or companies. These data are used for the development of the algorithm. A milling data set has been used. Experiments were conducted on a milling machine for feeds, depth of cut, and different speeds. BEST lab at UC Berkeley awarded the donated the following dataset.

The dataset contains 166 samples and each sample containing 9000 sensor measurements.

case	run	DOC	feed	material	M1	M2	M3	M4	M5
1	1	1.5	0.5	1	0.314941	0.301514	0.303955	0.300293	0.299072
1	2	1.5	0.5	1	0.301514	0.308838	0.299072	0.306396	0.319824
1	3	1.5	0.5	1	0.29541	0.296631	0.292969	0.290527	0.286865
1	4	1.5	0.5	1	0.316162	0.311279	0.302734	0.322266	0.323486
1	5	1.5	0.5	1	0.284424	0.289307	0.284424	0.288086	0.27832
1	6	1.5	0.5	1	0.307617	0.310059	0.306396	0.310059	0.301514
1	7	1.5	0.5	1	0.322266	0.299072	0.313721	0.296631	0.307617
1	8	1.5	0.5	1	0.308838	0.303955	0.306396	0.29541	0.294189
1	9	1.5	0.5	1	0.284424	0.288086	0.286865	0.284424	0.286865
1	10	1.5	0.5	1	0.279541	0.284424	0.279541	0.3125	0.291748
1	11	1.5	0.5	1	0.301514	0.303955	0.294189	0.292969	0.322266
1	12	1.5	0.5	1	0.391846	0.333252	0.358887	0.324707	0.336914
1	13	1.5	0.5	1	0.317383	0.328369	0.310059	0.317383	0.305176
1	14	1.5	0.5	1	0.351563	0.299072	0.299072	0.330811	0.302734
1	15	1.5	0.5	1	0.29541	0.292969	0.290527	0.314941	0.297852

Table 4. 1 Sample of the milling machine data

The examples of the few samples are plotted below.

Row 16

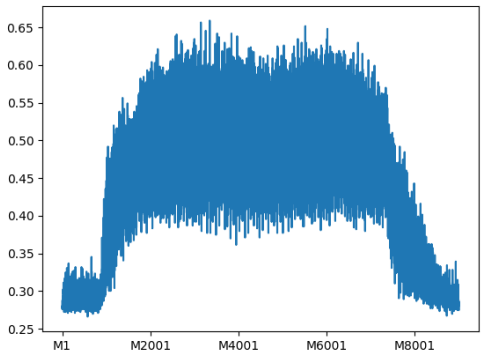

Figure 4. 3 Raw input of row16

Row 17

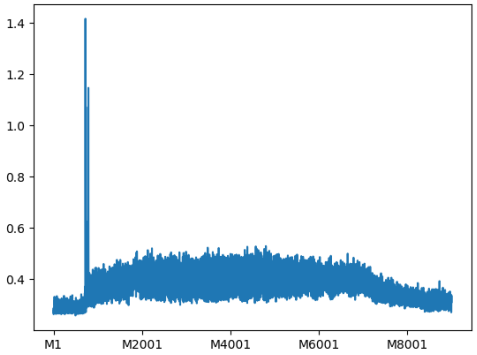

Figure 4. 4 Raw input of row17

Row 35

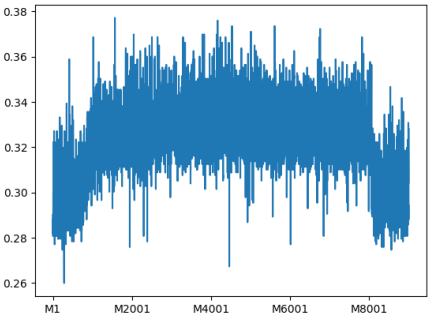

Figure 4. 5 Raw input of row 35

Row 108

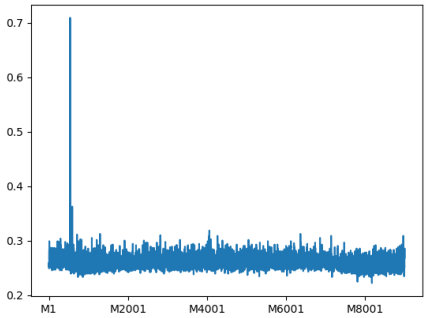

Figure 4. 6 Raw input of row108

Row 18

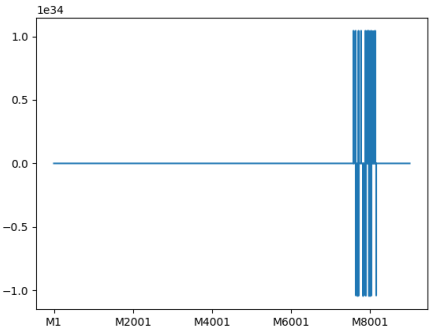

Figure 4. 7 Raw input of row 18

Row 25

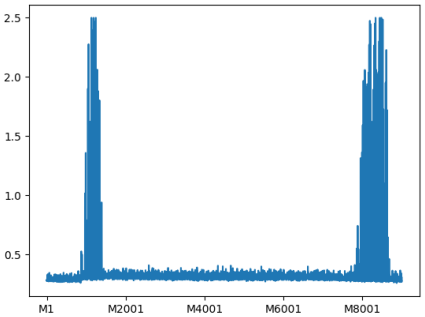

Figure 4. 8 Raw input of row 25

4.3 Methods for detecting Changepoint

The changepoints can be determined by several statistical models, in this experiment the methods considered are as follows.

4.3.1 PELT

When the number of changepoints is multiple and unknown the pelt method is used to determine the changepoints. The details and functions of PELT are already explained in the 2 chapters. A random set of rows has been considered for this experiment. Row 5 has been considered.

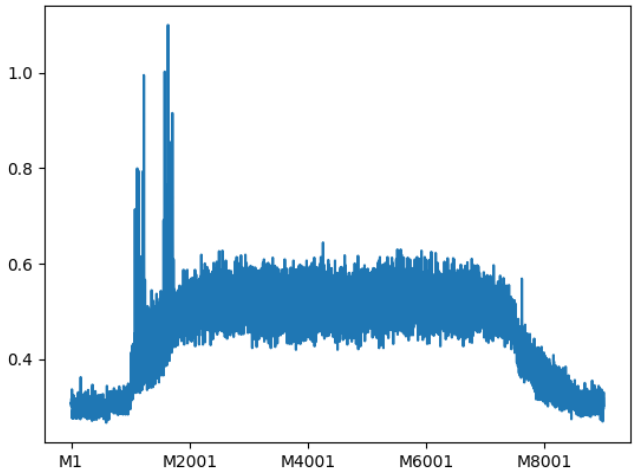

Figure 4. 9 Raw input of row 5

4.3.1.1 Algorithm for changepoint detection

```
# change point detection
model = "l1"  # "l2", "rbf"
algo = rpt.Pelt(model=model, min_size=3, jump=5).fit(
mysignal)
my_bkps = algo.predict(pen=10)
# show results
fig, (ax,) = rpt.display(mysignal, my_bkps, my_bkps,
figsize=(10, 6))
plt.show()
```

Once the algorithm is successfully carried out the changepoints are successfully identified and the graph shows their positions for visual understanding

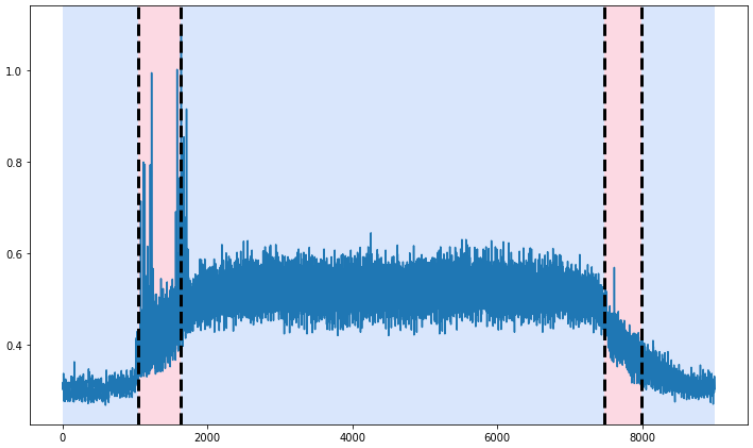

Figure 4. 10 Changepoint detections of row 5 by PELT

The breakpoints are also identified which are as follows 1045, 1625, 7475, 7995, 9000.
Then the processes are identified as (0,1045), (1045, 1625), (1625, 7475), (7475, 7995), (7995,9000).

4.3.2 Dynamic programming

When the number of the breakpoints is known they can be used in this model. Where the number of breakpoints can be entered, and the algorithm manages to detect them. The details about dynamic programming are already explained above.

4.3.2.1 Algorithm

```
# changepoint detection
model = "l1"  # "l2", "rbf"
algo = rpt.Dynp(model=model, min_size=3, jump=5).fit(
mysignal)
my_bkps = algo.predict(n_bkps=4)
```

```
# show results
rpt.show.display(mysignal, my_bkps, my_bkps, figsize=
(10, 6))
plt.show()
```

In this line of the program, the number of breakpoints can be given. Here it is given as "4".

```
my_bkps = algo.predict(n_bkps=4)
```

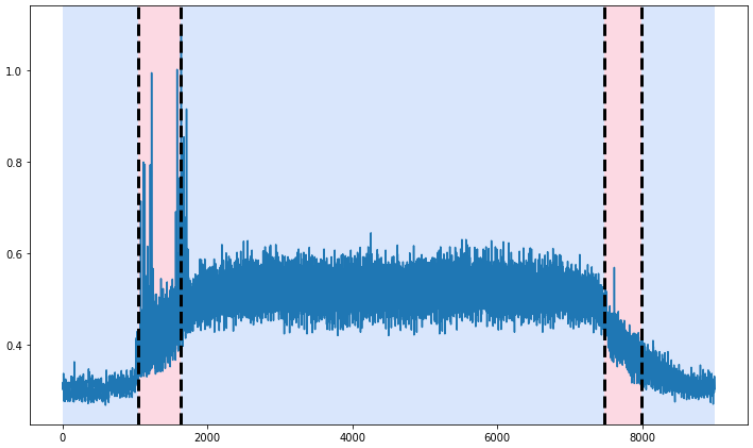

Figure 4. 11 Changepoint detections of row 5 by DyP

The breakpoints are also identified which are given as 1045, 1625, 7475, 7995, 9000.

Then the processes are identified as (0, 1045), (1045, 1625), (1625, 7475), (7475, 7995), (7995,9000).

4.4 Segmentation of the processes

After identifying the breakpoints in either of the above-mentioned methods, it is essential to segment the processes. Using the data frame on the python these processes can be made as a list. Now, these data must be examined. The most intuitive way is to understand the mean of all the processes. The stationary process can be identified as the highest mean of all processes. Since the machine will be performing its maximum during this stationary process. When it is identified, the processes on the left side can be said start-up or warm-up processes, and then the processes on the right to the stationary process can be identified as warm down or the ending phase.

In python, to find mean the preinstalled operation called "mean" can be used directly.

4.4.1.1 Algorithm

```
list1=[]
for i in range(len(my_bkps)-1):
globals()['data' + str(i)]=mydata.iloc[row][my_bkps[i]+5
:my_bkps[i+1]]
globals()['mean' + str(i)]=(globals()['data' + str(i)].
mean())
list1.append(globals()['mean' + str(i)])
```

Then the maximum of the average has been identified and the location of the breakpoints is found. Using the two breakpoints the stationary process can be determined. The stationary process got the most relevant data for the operation, so it is fetched with high importance.

4.4.1.2 Algorithm

```
max(list1)
maxpos=list1.index(max(list1))

sp1=my_bkps[maxpos]
sp2=my_bkps[maxpos+1]
print("The stationary phase is from",sp1,"to",sp2)
```

Output

```
The stationary phase is from 1625 to 7475.
```

4.5 Fetching the process

Once the segmentation of the processes is done. Then the processes can be plotted as graphs for visual monitoring, or the data can be used for further investigation. The data can be transferred to the cloud using Amazon web service or Microsoft Azure based on the infrastructure available for the architecture. It can also be held locally depending on the need of the user.

4.5.1.1 Algorithm

```
#Fetching the process
stationaryprocess.plot()
images_dir = '/gdrive/MyDrive/thesis_data/SP'
plt.savefig(f"{images_dir}/"+str(i)+".png")
plt.close()
```

In the above scenario, the data has been stored in google drive for easier access. Likewise, the data can be stored offline too. The same pattern can be used for sending the data to the cloud location for further handling of the data.

4.6 Implementation

The structure of the algorithm can be customized in different ways to accommodate the demand of the user. Here it has been used in two different ways; they are interpreting the data by a single row. The user can choose the desired row he wants to examine in particular. The second option compiles the whole data and runs the algorithm row by row for all the rows automatically. They both have their functions.

4.6.1 Interpreting each row

The Interpretation is carried out individually for each row. The user must enter the row or the position of the data that must be examined. Then the algorithm fetches only the data of that particular row and carries the test for it. In the algorithm given below the number 5 has been entered, so the algorithm identifies the row number 5(which is row number 6 in excel, since the indexing in python starts from 0) and then carries out the remaining operations.

4.6.1.1 Advantages

- It is used to study the particular set of data that the user needs.
- It is fast compared to the other method since only one row is processed at a time.
- It can be used to detect anomalies once the data has arrived.

4.6.1.2 Algorithm

```
#Reading the data
mydata = pd.read_csv("Merged1.csv")

#Entering the row to be interpreted
row=5
mysignal = mydata.iloc[row][5:]
#mysignal
mysignal = mysignal.to_numpy()

#change point detection
model = "l1"  # "l2", "rbf"
algo = rpt.Pelt(model=model, min_size=3, jump=5).fit
(mysignal)
my_bkps = algo.predict(pen=10)

# show results
fig, (ax,) = rpt.display(mysignal, my_bkps,
my_bkps, figsize=(10, 6))
plt.show()
df=pd.DataFrame(my_bkps)
my_bkps.insert(0,0)
list1=[]
for i in range(len(my_bkps)-1):
globals()['data' + str(i)]=
mydata.iloc[row][my_bkps[i]+5:my_bkps[i+1]]
globals()['mean' + str(i)]=(globals()
['data' + str(i)].mean())
list1.append(globals()['mean' + str(i)])

#finding the maximum average
max(list1)
maxpos=list1.index(max(list1))
#Stationary Phase
sp1=my_bkps[maxpos]
sp2=my_bkps[maxpos+1]
print("The stationary phase is from",sp1,"to",sp2)

#Fetching the stationary Phase
stationaryprocess=mydata.iloc[row][sp1:sp2]
stationaryprocess.plot()
images_dir = '/gdrive/MyDrive/thesis_data/SP'
plt.savefig(f"{images_dir}/"+str(i)+".png")
plt.close()
```

4.6.2 Compiling whole data

This allows the user to compile all the data when they are collected overall. It is used to study the performance after the halt of the machine. All the data are processed together, and it is a time-consuming process when the amount of data is high. In this case, data with 167 rows and 9000 columns are validated.

4.6.2.1 Advantage

- There is no manual intervention needed to enter the number of rows.
- All the data can be processed in a single execution.

4.6.2.2 Algorithm

```
for i in range(0,166):
   mysignal = mydata.iloc[i][5:]
   mysignal = mysignal.to_numpy()
   model = "l1"  # "l2", "rbf"
   algo = rpt.Pelt(model=model, min_size=3, jump=5)
   .fit(mysignal)
   my_bkps = algo.predict(pen=10)
   print(my_bkps)
   my_bkps.insert(0,0)
   list1=[]
   for j in range(len(my_bkps)-1):
     globals()['data' + str(j)]=mydata.iloc[i]
     [my_bkps[j]+5:my_bkps[j+1]]
     globals()['mean' + str(j)]=(globals()
     ['data' + str(j)].mean())
     list1.append(globals()['mean' + str(j)])
   max(list1)
   maxpos=list1.index(max(list1))
   sp1=my_bkps[maxpos]
   sp2=my_bkps[maxpos+1]
   stationaryprocess=mydata.iloc[i][sp1:sp2]
   print(sp1,sp2)
   stationaryprocess.plot()
   images_dir = '/gdrive/MyDrive/thesis_data/SP'
   plt.savefig(f"{images_dir}/"+str(i)+".png")
   plt.close()
```

5. RESULTS AND CONCLUSION

5.1 Discussion from the Literature

In this section, the findings from the case study are discussed, and also the findings from the literature review and the previous research.

The literature suggests that Changepoint detection has been a topic of discussion since the 1950s in the statistical and mathematical world. Later due to the advent of technology there is more amount data are gathered and the methods to study them were always rising. Changepoint detection plays a key and vital role in understanding data when there are different sets of data involved. Also, changepoint detection can be used to identify the fault in the machine when it is deviating the nominal value. This can be very useful in preventive maintenance in improving efficiency and also to reduce the cost of the breakdown and lowering the breakdown time.
The practical examples of changepoint detections are given below [15]

- Anomaly detection: By using the relevant data the abnormal circumstances/activities can be examined.
- Fault analysis: A supervisor can monitor and decide on the fault analysis performed using the detectable variables.
- Safety protection: The machine can be stopped to protect the operator while performing a dangerous process due to deviation found in monitoring.

Changepoint detection has been widely used in different kinds of fields because of its modern application. In Climatology the changepoint detection is used to monitor the data of the environment and satellite images to predict the weather. It is also used to determine the trends and seasonal changes. And, to monitor the vegetation and forests [8], [9].

Changepoint is also widely used in finance to understand different trends in the market. It can be used to monitor financial assets and their performances. It is also used in the field of medicine to understand genomes and other studies.

The literature suggests the parameters necessary to be monitored in the welding process.

- Current (A) [4],
- Voltage (V) [5,6],
- Speed of torch (in/min or m/min) [7],
- gas flow rate (L/min) [8],
- wire feed rate (in/min or m/min) [9,10]

5.2 Result from the case study

After investigating all the data of the dataset from the sensor of the vibration of the spindle, the changepoints were detected and some empirical results were found. All the data were different and varying so the study result can be considered to be valid for most of the operations of the machine. Here the results are shown for the milling machine and similarly, this can be used for the welding machine with certain adjustments according to the dataset. It is also very essential to study the historical performance of the machine to understand the anomalies.

5.2.1 Comparisons between PELT and Dynamic processing

The exact data are compared in both the methods and the results of some key data are subjected here for discussion.

5.2.1.1 Sample 1: ROW 2

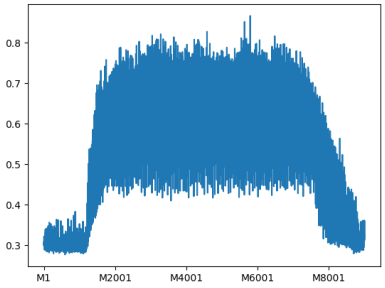

Figure 5. 1 Raw Input of row 2

The raw input of row 2 is a very standard dataset where the changepoints are widely spread across the dataset. It incorporates all the processes of the milling machine Start, run-up, stationary, run-down, stop.

5.2.1.1.1 PELT

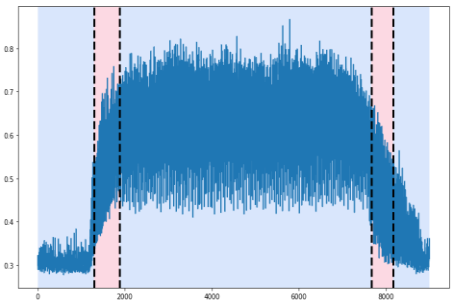

Figure 5. 2 Changepoint detections on row 2 by PELT

The raw input of row 2 seems to follow a normal process involving five stages. Since the breakpoints are widely spread throughout the dataset like stated in the assumption of the PELT modelling, it would be ideal for the algorithm to determine the changepoints and locate them.

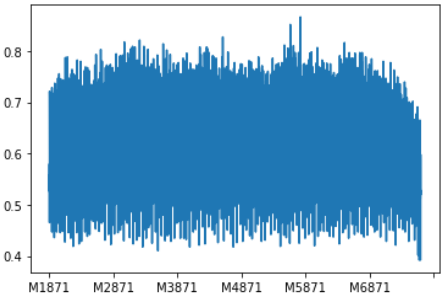

Figure 5. 3 Stationary process on row 2 by PELT

```
The breakpoints are [0, 1295, 1875, 7675, 8165, 9000]
The stationary phase is from 1875 to 7675.
Program Executed in 94.62296830000014.
```

Like predicted before the algorithm worked very well and detected the changepoints and identified their position. It can be understood that the algorithm is fast and effective when the data of the processes were following a perfect structure and sequence.

5.2.1.1.2 Dynamic Processing

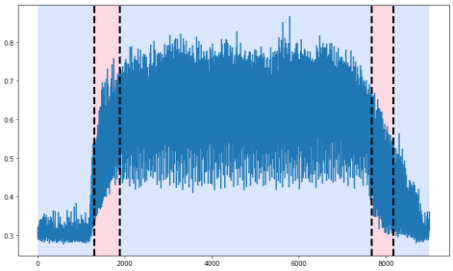

Figure 5. 4 Changepoint detections on row 2 by DyP

Again, the same raw input of row 2 is taken and the algorithm using dynamic programming is used to detect the changepoint. It is done to compare both the results achieved.

```
The breakpoints are [1295, 1875, 7675, 8165, 9000]
```

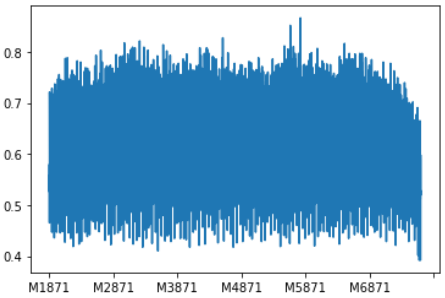

Figure 5. 5 Stationary process on row 2 by DyP

```
The stationary phase is from 1875 to 7675.
Program Executed in 244.7910427679999
```

The number of breakpoints is given as 4 for the dynamic programming and the results of the breakpoint are the same in both PELT and Dynamic programming. It has to be understood that this raw input follows the traditional set of processes such as Start, run-up, stationary, run-down, and stop. So, it is possible to achieve the same results in both cases.

5.2.1.2 Sample 2: Row 9

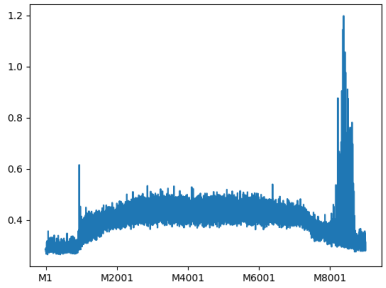

Figure 5. 6 Raw input of row 9

The raw input of row 9 is quite different from the normal observations. The vibration of the spindle seems to be higher while performing some operation and it can be visually observed here. Also, there can be some noise in the sensor data because of other external factors. Apart from the spike at the end of the process, the row 9 data seem to be quite similar to the normal processes shape.

5.2.1.2.1 PELT

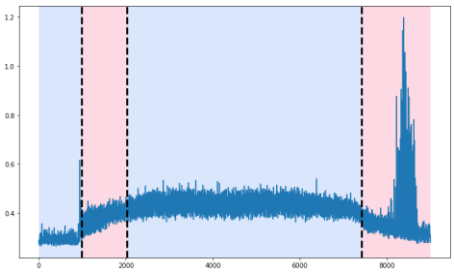

Figure 5. 7 Changepoint detections on row 9 by PELT

The hike in the data has not been particularly considered as a changepoint since it is very sharp and rapid. It can be understood that it is due to some discrepancy. But the algorithm identifies the changepoint quite before the spike regions since the values are gradually lowering which identifies that there is a process change.

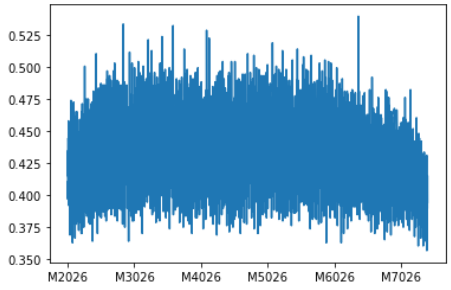

Figure 5. 8 Stationary process on row 9 by PELT

```
The breakpoints are [990, 2030, 7425, 9000].
The stationary phase is from 2030 to 7425.
Program Executed in 02.58818904500004
```

5.2.1.2.2 Dynamic Processing

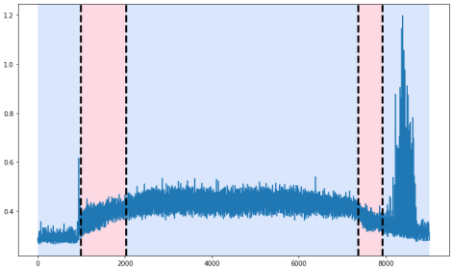

Figure 5. 9 Changepoint detections on row 9 by DyP

```
The breakpoints are [990, 2030, 7355, 7915,9000].
```

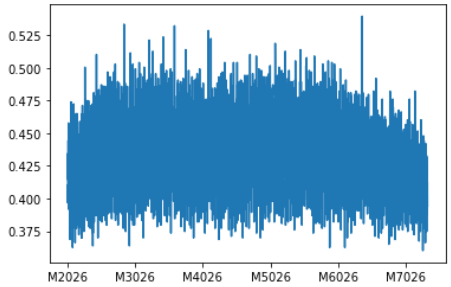

Figure 5. 10 Stationary process on row 9 by DyP

```
The stationary phase is from 2030 to 7355. Program Executed in
238.0945
```

In this set of data, there can be some differences noticed between the two results. There are some differences between breakpoints and stationary phase in both the results. The dynamic programming still follows four breakpoints, so it is obliged to find four breakpoints irrespective of the data.

5.2.1.3 Sample 3: Row 33

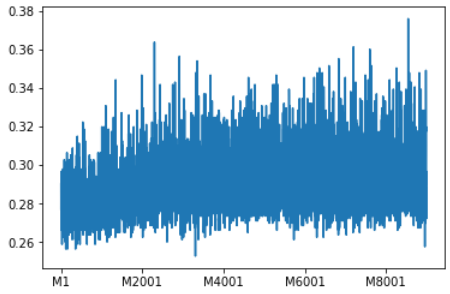

Figure 5. 11 Raw input of row 33

The raw input of row 33 is quite different since it doesn't show any signs of gradual increase indicating different processes involved instead it seems to be very normal from start to end.

5.2.1.3.1 PELT

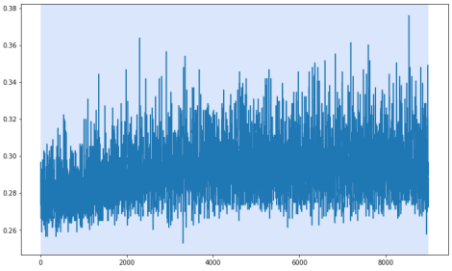

Figure 5. 12 Changepoint detections on row 33 by PELT

```
The breakpoints are [9000].
```

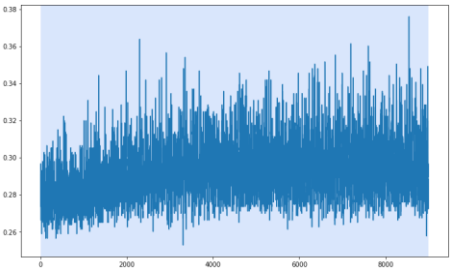

Figure 5. 13 Stationary process on row 33 by PELT

```
The stationary phase is from 0 to 9000. Program Executed in
190.8593248369998
```

It can be noted that there is not any changepoint detected. Since the data itself is stationary statistically.

5.2.1.3.2 Dynamic Processing

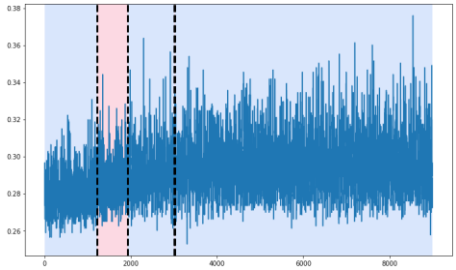

Figure 5. 14 Changepoint detections on row 33 by DyP

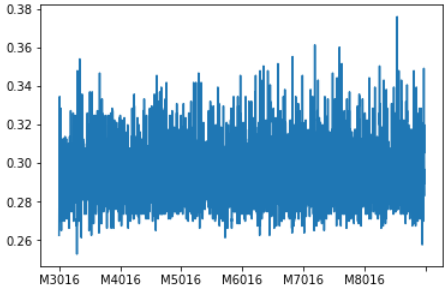

Figure 5. 15 Stationary process on row 33 by DyP

```
The breakpoints are [1210, 1930, 3010, 3020, 9000]
The stationary phase is from 3020 to 9000.
Program Executed in 251.55086548500003
```

In this method, there are 4 breakpoints even if the process itself looks stationary. The algorithm is determined to find the breakpoints obligatory. There is a huge difference between the two methods. It must be noted this difference can cause a huge discrepancy in the monitoring of the processes because of the wrong signals which can raise a false alarm.

5.2.1.4 Sample 4: Row 117

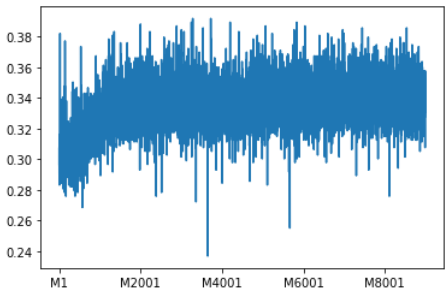

Figure 5. 16 Raw input of row 117

5.2.1.4.1 PELT

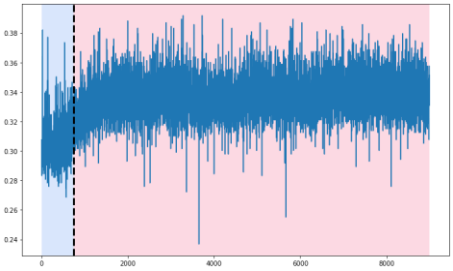

Figure 5. 17 Changepoint detections on row 117 by PELT

```
The breakpoints are [740, 9000]
```

In this dataset, there is no presence of standard processes instead it is almost during the run-up and extended up to the stationary process.

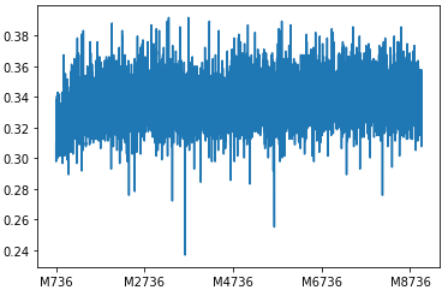

Figure 5. 18 Stationary process on row 117 by PELT

```
The stationary phase is from 740 to 9000.
Program Executed in 191.50136976500107
```

5.2.1.4.2 Dynamic Processing

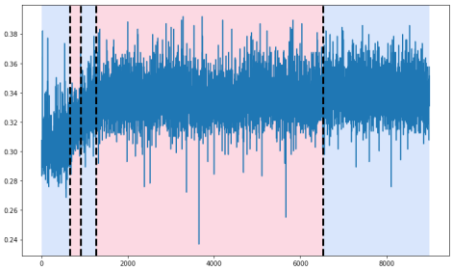

Figure 5. 19 Changepoint detections on row 117 by DyP

```
The breakpoints are [660, 900, 1265, 6525, 9000]
```

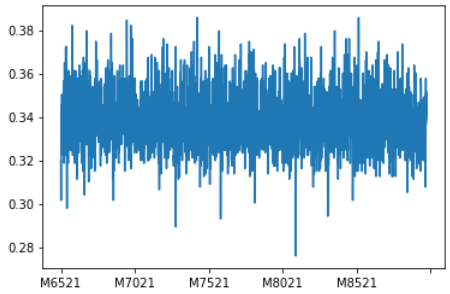

Figure 5. 20 Stationary process on row 117 by DyP

```
The stationary phase is from 6525 to 9000.
Program Executed in 256.0959811570001
```

Also, in this set of data, there can be some differences noticed between the two results. There are some differences between breakpoints and stationary phase in both the results. The dynamic programming still follows four breakpoints.

5.2.1.5 Sample 5: Row 118

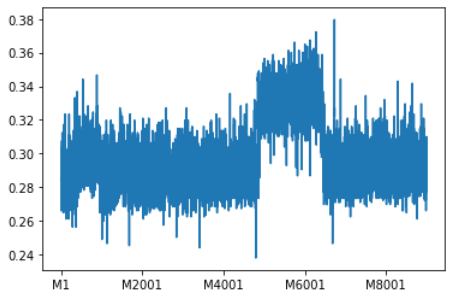

Figure 5. 21 Raw input of row 118

5.2.1.5.1 PELT

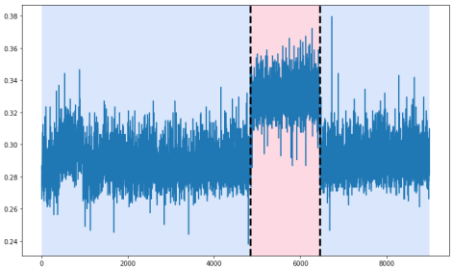

Figure 5. 22 Changepoint detections on row 118 by PELT

The stationary process is identified as the process with the highest mean value of all the processes. It is tricky here because it can't be concluded as a stationary process. Other processes are also stationary processes.

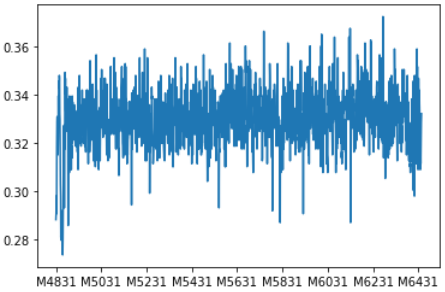

Figure 5. 23 Stationary process of row 118 by PELT

```
The breakpoints are [4835, 6450, 9000]
The stationary phase is from 4835 to 6450.
Program Executed in 76.20090490500115
```

5.2.1.5.2 Dynamic Processing

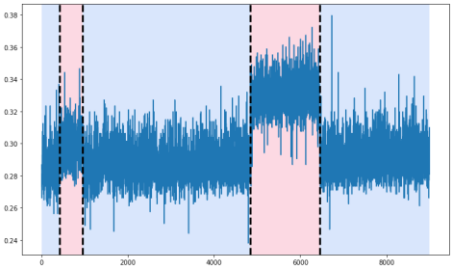

Figure 5. 24 Changepoint detections of row 118 by DyP

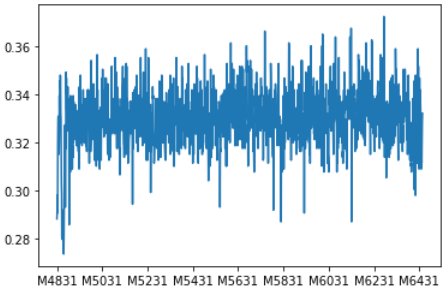

Figure 5. 25 Stationary process of row 118 by DyP

```
The breakpoints are [420, 955, 4835, 6450, 9000]
The stationary phase is from 4835 to 6450.
Program Executed in 234.42001677499866
```

5.3 The result from the other sensors data

There other different kinds from the milling machine like mentioned before. The algorithm has been tested randomly on other sensor data. The results were quite interesting to support the robustness of the algorithm such that it can be used with different kinds of dataset.

The datasets verified on the samples of

- Direct Current,
- Alternate Current,
- Vibrations of the table.

It must be noted it is not an in-depth analysis of the dataset like the dataset of vibration of the spindle.

5.3.1 Direct Current

The Dataset of direct current is very gradual compared to the vibration of the spindle. The dataset follows the processes very clearly. Below given are some examples of the dataset.

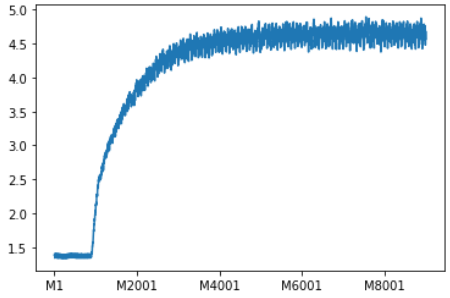

Figure 5. 26 Raw input of row 36 of DC sensor data

5.3.1.1 Changepoint detection by PELT

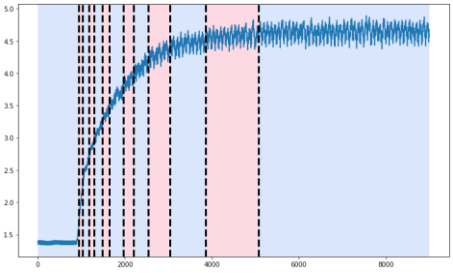

Figure 5. 27 Changepoint detections of row 36 by PELT

```
The breakpoints are [945, 1025, 1170, 1295, 1480, 1650, 1960,
2205, 2535, 3030, 3850, 5080, 9000]. The stationary phase is
from 5080 to 9000.
```

The changepoints identified in this process are more in numbers using the PELT method. It can be a time-consuming process if this is the state. It will be interesting to see the results of the same data using the dynamic programming method.

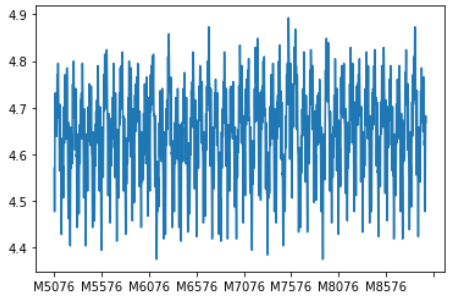

Figure 5. 28 Stationary processes of row 36 by PELT

5.3.1.2 Changepoint detection by DyP

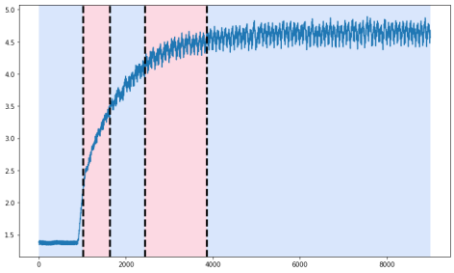

Figure 5. 29 Changepoint detections of row 36 by DyP

```
The breakpoints are [1020, 1625, 2435, 3850, 9000]. The
stationary phase is from 4835 to 6450. Program Executed in
234.42001677499866
```

Since the number of breakpoints is a prerequisite the number 4 is given and then the above results have been obtained. It can be noted that dynamic programming can be applied in certain situations. Anyway, this needs further investigation and a detailed historic analysis of the sensor's data.

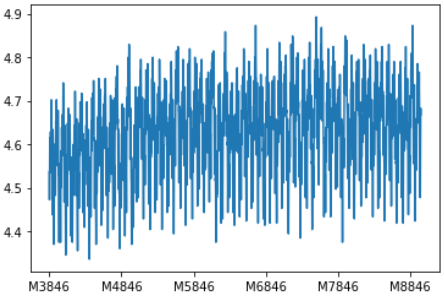

Figure 5. 30 Stationary processes of row 36 by DyP

It can be noted that there is some difference between the stationary process in both cases. But the difference is not really high in general situations, but it can be relevant in certain critical situations.

5.3.2 Alternate current

Alternate current is a key parameter to be monitored in most of the machining operations, but it is to be noted that the current normally stays constant in the circuit. It can be adjusted through the varying resistance in the case of fluctuating voltage.
The given example is the sensor of the Alternate current in the milling machine.

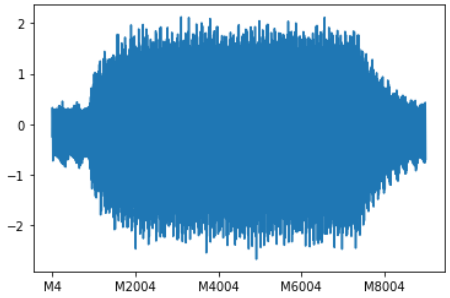

Figure 5. 31 Raw input of row 18 of DC sensor data

The raw input of the alternate current is constant like expected.

5.3.2.1 Changepoint detection by PELT

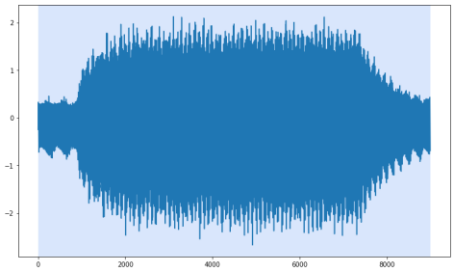

Figure 5. 32 Changepoint detections of row 18 by PELT

There is no changepoint detected using the PELT. Since the data is widely constant, there are no breakpoints in this model.

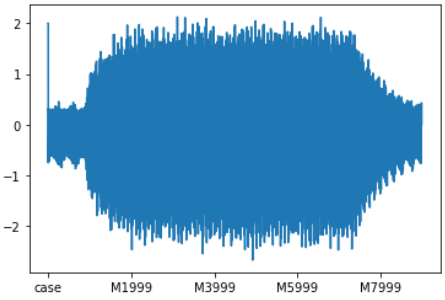

Figure 5. 33 Stationary process of row 18 by PELT

5.3.3 Vibration of table

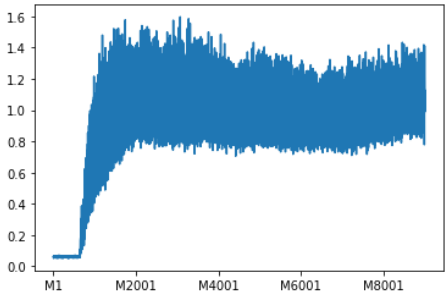

Figure 5. 34 Raw input of row 158 of sensor data of Vibration of table

In the sensor data of vibration of the table. It can be noted that there is a significant change in the data because of the milling process which leads to vibration of the table. So, it can be possible for the segmentation of the processes and then extract the necessary features.

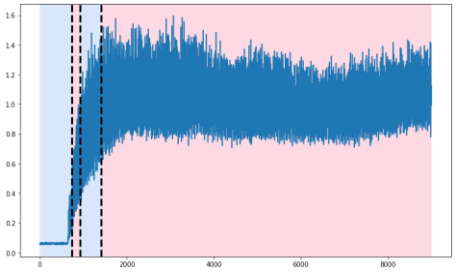

Figure 5. 35 Changepoint detections of row 158 by PELT

```
The breakpoints are [730, 920, 1405, 9000]
```

There are 4 stages in this dataset according to the changepoint detection. The stationary process can be identified as the process which has the highest mean. The same method used in the case study can be used here.

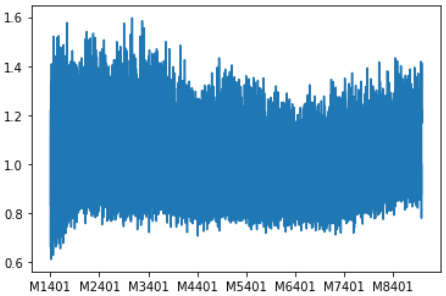

Figure 5. 36 Stationary process of row 158 by PELT

```
The stationary phase is from 1405 to 9000.
Program Executed in 171.38123119800002
```

The stationary process has been identified as from 1405 to 9000. This denotes that there are no run-down or stop processes in this data set. The processes involved in this are a start, run-up, and stationary process.

5.3.4 Inference

The Changepoint detection is applied to the different kinds of sensor data from the milling machine using the sensor data. It can be noted that the algorithm is functional for all other sensor data. The mathematical modelling can be determined based on the raw input of the sensor data and the requirement of the user. There can be some difference to be noted in Alternate current sensor data because the AC sensor data seems to be very stable and there is not much deviation for finding the changepoint detection.

5.4 Implications and recommendations

5.4.1 Verdict between the PELT and Dynamic processing

Based on the results from the case study in detecting the changepoints the PELT is more relevant to the real case scenario. There are few reasons which support this finding.

- There is no need to predefine the number of changepoints to be found in this data to be tested.
- The speed of computing can be noticed to faster in PELT compared to Dynamic processing. This will help to reduce the latency period of the processing of data.
- There is no obligatory changepoint detection in the PELT method, whereas Dynamic processing is obliged to find some breakpoints because of the requirement of finding 4 breakpoints in the case shown above even if the data is stationary itself.

Dynamic programming can also be used in certain cases, but it can be decided after the most intensive study of the data. It is also dependent on the latent time constraint of the user.

- Dynamic programming can be relevant when there are no significant findings by PELT like in Alternate current data. By understanding the historical findings, the number of breakpoints can be predetermined and entered.

5.4.2 Verdict between the Interpretation and compiling

The above set of has been executed by both Interpretation and compiling, and there are some implications found and are as follows,

- The compiling of all the data once they are collected is carried out without any manual interventions. This can be highly helpful in the automated setup where the results can be studied at the end of the day after the shift time is over.
- The time taken for executing the program using the data of 167 rows and 9000 columns took more than 4 hrs. and 30 minutes. So, it is a time-consuming process because of the heavy load of data presented for the execution.
- It must be noted that this compiling approach is an offline changepoint detection method, so it cannot be used for finding the real-time anomalies of the machine performance or to stop the machine from operating when a fault is detected.
- The interpretation of each line of data is quite faster compared to compiling the whole data method but still, the processing of data in both cases is to be enhanced.
- If interpretation can be further enhanced with respect to the speed of the program execution. It can be implemented in online changepoint detection for anomaly detections to ensure the safety of both the operator and machine.

5.5 Limitations and Future research

From the analysis, there are a certain set of data which are having some irrelevant data or invalid data. This affects the change point detection.

5.5.1 Cleaning the data

In row 18 as shown below, the data can be seen as very different from the other samples of the same sensor data.

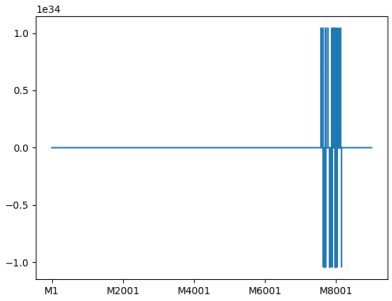

Figure 5. 37 Raw input of row 18

It can be noticed that there are certain hikes in the data and the rest is a straight line. But if it is closely examined the straight line is data of different processes and it is relevant to be studied. The hikes are due to anomalies, and these affect the change point detection of the data. In order to overcome this factor, it is essential to clean the raw input and analyze only the relevant data to avoid time wastage and efficient detections of the changepoints.

5.5.2 Using the historical values

It can be understood that the stationary process is identified by the highest mean of the processes but in sample 5. The stationary processes are just a hike from the nominal working range. It cannot be considered a stationary point. To find the right one. It is essential to study the historical performance of the machine and then set the limits for the mean of the stationary point. If the processes lie within the limit, it can be considered a stationary process irrespective of the hike found in the data. This can reduce the anomalies in the analysis.

5.5.3 Using sampling

The latent time and execution time are quite high in some cases, it should be noted that very less latent time is required for online changepoint detection. So, it should be considered that there must be some research done to reduce. There is a suggestion that sampling techniques can be used to make the data less and effective for the analysis. The small data with the same features of the dataset will help the user to get close results like in the original dataset.

5.6 Conclusion

The purpose of this work is to understand the Changepoint detections, different methods involved in the detection, and to use them in appropriate needs. The study has been carried from the theoretical point of view from the understanding the literature which introduced the history and different applications of Changepoint detection. It has helped to understand the statistical models involved in Changepoint detection.

The case study has been conducted on 167 rows and the results imply that the PELT method is more appropriate in detecting Changepoints when the breakpoints are unknown. Again, the experiment on Interpretation of each line of data and compiling the whole data suggests that it can be used based on the necessity of the user. Interpretation can be used when a sampling quality check is been carried out. It can be used for online changepoint detection when it can receive the data from the machine in real-time. Compiling the whole data can be used to study the in-depth analysis of the performance of the machine for the day or during the period. Even though it is time-consuming it can be used in relevant operations when there is enough time for analysis.

To conclude, Firstly the thesis understood the existing solution available in the literature and used it in the technology available in the research world. However, the majority of the studies available were on the theoretical aspects, this thesis focuses on the practical application of the topic by algorithm and case study. The algorithm has been tested on a real-life scenario using the dataset from NASA. As to my knowledge, this algorithm and technique can be used for welding machines and other computer numeric machines too with further tweaks based on the historical study of the machines.

BIBLIOGRAPHY

[1]. Singh, R., Applied welding engineering: processes, codes, and standards. 2015: Butterworth-Heinemann.

[2]. Mishra, R.S. and Z. Ma, Friction stir welding and processing. Materials science and engineering: R:reports, 2005. 50(1-2): p. 1-78

[3]. Hamzeh, R. and X. Xu, Technology selection methods and applications in manufacturing: A review from 1990 to 2017. Computers & Industrial Engineering, 2019. 138: p. 106123.

[4]. Sumesh, A., et al., Decision tree based weld defect classification using current and voltage signatures in GMAW process. Materials Today: Proceedings, 2018. 5(2): p. 8354-8363

[5]. Reyes-Valdés, F., et al., Correlation between GMAW process and weld quality parameters: A neural network approach applied in the automotive industry.FABTECH and AWS 2006, 2006.

[6]. WaelKhalifa, O.A. and I.E. ElamirGadelmawla, Classification of Welding Defects Using Gray Level Histogram Techniques via Neural Network

[7]. Chan, B., J. Pacey, and M. Bibby, Modelling gas metal arc weld geometry using artificial neural network technology. Canadian Metallurgical Quarterly, 1999. 38(1): p. 43-51.

[8]. Tani, G., et al., The influence of shielding gas in hybrid LASER–MIG welding. Applied Surface Science, 2007. 253(19): p. 8050-8053.

[9]. DuPont, J. and A. Marder, Thermal efficiency of arc welding processes. Welding Journal-Including Welding Research Supplement, 1995. 74(12): p. 406s.

[10]. Arya, H. and R. Saxena, Effect of cooling rate on microstructure of SAW welded mild steel plate (gradeC 25 as per IS 1570). Int. J. Mod. Eng. Res., 2014: p.222.

[11]. Fraden, J., Handbook of modern sensors. Vol. 3.2010: Springer.

[12]. Kah, P., et al., Robotic arc welding sensors and programming in industrial applications. International Journal of Mechanical and Materials Engineering, 2015. 10(1): p. 13

[13]. Madigan, R., Arc sensing for defects in constantvoltage gas metal arc welding. Welding Journal,1999. 78: p. 322S-328S.

[14]. Koseeyaporn, P., G.E. Cook, and A.M. Strauss, Adaptive voltage control in fusion arc welding. IEEE Transactions on Industry Applications, 2000. 36(5): p.1300-1307.

[15]. Honarbakhsh-Raouf, A. and H. Ghazvinloo, Influence of Wire Feeding Speed, Welding Speed and Preheating Temperature on Hardness and Microstructure of Weld in RQT 701-british Steel Produced by FCAW. Indian Journal of Science and Technology, 2010. 3(5): p. 588-591

[16].Kah, P., et al., Robotic arc welding sensors and programming in industrial applications. International Journal of Mechanical and Materials Engineering,2015. 10(1): p. 13.

[17] Page, E. S. "Continuous Inspection Schemes." Biometrika, vol. 41, no. 1/2, 1954, pp. 100–115. JSTOR, www.jstor.org/stable/2333009. pp. 100-105

[18] M. Lavielle, G. Teyssière
Adaptive detection of multiple change-points in asset price volatility
Long-Memory in Economics, Springer Verlag, Berlin, Germany (2007), pp. 129-156

[19] K. Frick, A. Munk, H. Sieling Multiscale change point inference
J. R. Stat. Soc. Ser. B, 76 (3) (2014), pp. 495-580

[20] Bai, J., & Perron, P. (1998). Estimating and Testing Linear Models with Multiple Structural Changes. Econometrica, 66(1), 47-78. doi:10.2307/2998540

[21] Picard, F., Robin, S., Lavielle, M. et al. A statistical approach for array CGH data analysis. BMC Bioinformatics 6, 27 (2005). https://doi.org/10.1186/1471-2105-6-27

[22] Guédon, Y. Exploring the latent segmentation space for the assessment of multiple change-point models. Comput Stat 28, 2641–2678 (2013). https://doi.org/10.1007/s00180-013-0422-9

[23] S. Liu, A. Wright, M. Hauskrecht
Change-point detection method for clinical decision support system rule monitoring
Artif. Intell. Med., 91 (2018), pp. 49-56

[24] R. Maidstone
Efficient analysis of complex changepoint problems, Lancaster University (2016)

[25] J. Verbesselt, R. Hyndman, G. Newnham, D. Culvenor
Detecting trend and seasonal changes in satellite images time series
Remote Sens. Environ. (114) (2010), pp. 106-115

[26] Reeves, J., Chen, J., Wang, X.L., Lund, R., Lu, Q.Q.
A review and comparison of changepoint detection techniques for climate data
(2007) Journal of Applied Meteorology and Climatology, 46 (6), pp. 900-915.

[27] Charles Truong, Laurent Oudre, Nicolas Vayatis,(2020) Selective review of offline change point detection methods, Signal Processing,Volume 167,
(https://doi.org/10.1016/j.sigpro.2019.107299)

[28] P. Varshney and Y. Simmhan, "Demystifying Fog Computing: Characterizing Architectures, Applications and Abstractions," 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, 2017, pp.115-124.

[29] L. F. Bittencourt, M. M. Lopes, I. Petri and O. F. Rana, "Towards Virtual Machine Migration in Fog Computing,"2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, 2015, pp. 1-8

[30] M. S. D. Brito, S. Hoque, R. Steinke and A. Willner, "Towards Programmable Fog Nodes in Smart Factories,"2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W), Augsburg,2016, pp. 236-241

[31] S. Liu, J. A. Guzzo, L. Zhang, D. W. Smith, J. Lazos and M. Grossner, "Plug-and-play sensor platform for legacy industrial machine monitoring," 2016 International Symposium on Flexible Automation (ISFA), Cleveland, OH, 2016, pp. 432-435

[32] Jackson Brad, Sargle Jeffrey D, Barnes David, Arabhi Sundararajan, Alt Alina, Gioumousis Peter, Gwin Elyus, Sangtrakulcharoen Paungkaew, Tan Linda, Tsai Tun Tao(2005). An algorithm for optimal partitioning of data on an interval. *IEEE, Signal Processing Letters, 12*(2), 105-108.

[33] Gombay Edit, & Horvath Lajos (1996). On the Rate of Approximation for Maximum Likelihood tests in Change-point Models. *Journal of Multivariate Analysis, 56*, 120-152.

[34] A. Agogino and K. Goebel (2007). BEST lab, UC Berkeley. "Milling Data Set ", NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA

[35] Guoliang Lu, Yiqi Zhou, Changhou Lu, Xueyong Li,
A novel framework of change-point detection for machine monitoring, Mechanical Systems and Signal Processing,Volume 83,2017,Pages 533-548,

[36] Gachomo Dorcas Wambui, Gichuhi Anthony Waititu, Anthony Wanjoya. The Power of the Pruned Exact Linear Time(PELT) Test in Multiple Changepoint Detection.American Journal of Theoretical and Applied Statistics.Vol.4, No. 6, 2015, pp. 581-586. doi: 10.11648/j.ajtas.20150406.30

[37] Jackson Brad, Sargle Jeffrey D, Barnes David, Arabhi Sundararajan, Alt Alina, Gioumousis Peter, Gwin Elyus, Sangtrakulcharoen Paungkaew, Tan Linda, Tsai Tun Tao(2005). An algorithm for optimal partitioning of data on an interval. IEEE, Signal Processing Letters, 12(2), 105-108.