# Internship Report

*by* Aditya Prakash

**Submission date:** 18-Sep-2020 11:03PM (UTC+0530)
**Submission ID:** 1389589793
**File name:** Control_Chart_Edited.docx (2.32M)
**Word count:** 398
**Character count:** 30311

# Control Chart Patterns Recognition Using Convolutional Neural Network

Summer Internship

**Report by**

**Sumit Gaurav**

**Sarthak Taunk**

# 1. Introduction

In the world of market competition of enterprises, product quality is always seen as a crucial key factor. Significant improvements have been brought to the production quality of a number of enterprises by evolving use of Statistical process control (SPC). Including industries of machining, this remarkable boost in production quality has been introduced to chemical industries, electronic industries etc. The basic idea behind SPC is to monitor diverse stages of the production process by using mathematical statistics methods. Through the use of SPC, production anomalies, deviations can be detected on time and according to it further, necessary measures can be implemented to eliminate potential hazards. Although, recognition of unnatural patterns is a critical task in statistical process control (SPC).

Process quality control, its principal objective is to achieve and maintain an acceptable level of the desired process quality characteristic steadily and consistently. In reference to its objective, accurate monitoring and effective control over the manufacturing system is tremendously important. Manufacturing of products with the desired quality needs sincere monitoring of production processes for distinguishing any unnatural deviation in the state of the process. And for determining whether a process is running in its intended mode or in presence of unnatural patterns, a control chart is used, which is an important statistical process control tool. The patterns exhibited on the control charts can provide essential information about the process. To point out quality failures and to detect root abnormal causes in time, recognition and analyzation of CCPs is thus considered. Control charts which are chiefly in the form of X chart, are widely used to recognize circumstances when manufacturing systems need control actions. The 8 most patterns formed on control chart are, normal pattern(NOR), stratification pattern (STA), systematic pattern (SYS), cyclic pattern (CYC), upward shift pattern (US), downward shift pattern (DS), uptrend pattern (UT) and downtrend pattern(DT).

1. **Normal Pattern(NOR)**: stipulates the production process in control.

2. **Systematics Pattern(SYS)**: SYS predicts points to point fluctuations. Its pattern emerges as a high point always following a low point and likewise.

3.**Stratification Pattern(STR)**: This shows that the data is more intensive and variance of data becomes nugatory.

4. **Cyclic pattern**: Appearance of peaks and troughs can be found in cyclic pattern, periodically.

5. **Trend pattern**: In trend patterns, a continuous rise (i.e. upward trend) or fall (i.e. downward trend) is shown by data.

6. **Shift pattern**: Unlike trend patterns, in shift patterns data results in sudden rise (i.e. upward shift) or sudden fall (i.e. downward shift) in the mean of data.

These patterns are broadly classified as natural/normal and unnatural/abnormal. A process under control is indicated by a natural pattern while in contrast to it an un natural pattern indicates out of control process. Conventionally in the production pr ocess, abnormal CCPs correspond to some abnormal causes. Hence, the recogni tion of abnormal patterns is helpful to identify the problems timely and the extent of abnormal causes can be narrowed as well.

Earlier in control chart applications, individuals' experience was indispensable to d educe whether the production process is abnormal or not and if it is found abnorm al, then to find the corresponding cause. By the evolution of industrial automation, the role of manual observation is partially replaced by the rule-
based discriminant system, where the discriminant rules of control charts are base d on minor probabilistic events, which can be easily carried out. Anyway, covering all abnormal patterns with rules is quite difficult due to intricacy in the production pr ocess. To compensate for this, a coherent automated pattern (CCP) recognition s ystem can be implemented, which ensures consistent, neutral interpretation of CC Ps resulting in a marginal number of false alarms and easy execution of control ch arts.

The advantage neural networks provide is provision of blunt rules or templates is n ot required here. Somewhat, it learns to recognize patterns straightly through typic al example patterns during the training phase and has the potential to recognize a n inconsistent pattern not previously encountered.

# 2. Literature Review

There are two major methods used to recognize control chart patterns:(I) directly f eed the raw CCPs data into the model and (II) uses the statistical properties of dat a like standard deviation, mean, distribution etc to extract the feature from data an d fed into the recognition model.

Pham et al. (1997) propose the approach that uses control chart patterns for featur e extraction instead of its numerical data and statistical properties for recognizing i ts patterns. It has two major steps: (I) feature extraction from CCP and (II) pattern r ecognition. This paper also uses techniques like heuristics and deep neural netwo rks. Gauri et al. (2007)proposed the application of CART to select the subset of fea tures. Addeh et al. (2018) used a method for pattern recognition which uses optimi zed RBFNN. This method is divided into four parts: feature extraction, feature sele ction, classification, and learning algorithm. After testing on 1600 datasets having 200 datasets for each pattern this method gave very good accuracy.

A fuzzy method for recognition of unnatural CCPs is proposed by Gulbay et al. (2 007). Zaman et al. (2018) used an efficient hybrid recognition method for CCP. Thi s method is divided into two parts: (I) feature selection and extraction part and (II) r ecognizer part. In the first part, a representation of each pattern using statistical fe atures is proposed and in the second part ANFIS along with FCM is proposed. Ebr ahimzadeh et al. (2011) proposed the SVM method due to its generalization perfor mance for recognition of CCP. Though, appropriate parameters selection for SVM is a bit difficult. Therefore, to optimize the parameters of SVM models automaticall y. Cheng et al. (1997) described two methods for pattern recognition: (I) multilayer perceptron trained by back propagation and (II) modular neural network and its per formance is evaluated by monte carlo simulation in which modular neural network showed better accuracy than back propagation. (Zhao et al., 2017) used improved supervised locally linear embedding and SVM for recognition in which it extract 1 2 dimensional statistical features and shape feature of control chart where as (Gho mi et al., 2017) used ANN to identify unnatural patterns formed on shewhart's cont rol chart to identify the out of control process. Spiking neural network(SNN) to the CCPs recognition is applied by Awadalla et al. (2012), which considered the contin uity of control chart data over time. Existing researches say that feature extraction based CCPs recognition method usually has better performance. But to select the feature subset which is best, feature screening methods are essentially needed, si nce the construction of features depends on human experience.

Deep learning is known for its outstanding performance and has been extensively studied consequently. An effective mapping from inputs to outputs by a network str ucture (Zhang et al., 2018) is established by Deep Learning. The deep learning mo del is simple but non-
linear modules which transform lower level representation into higher level repres entation and also extract features directly from raw data. There are various deep n eural networks such as artificial neural networks, convolutional neural networks, r ecurrent neural networks which have shown great significance in computer vision. Kiranyaz et al. (2016) implemented one dimensional CNN for ECG classification w

hich achieved a very good accuracy on NIT-
BIH arrhythmia data due to its speed and computational efficiency.. To analyze the chemo-metric data based on 1D-
CNN, a new method is proposed by Malek et al. (2017). The literature here, shows that through CNN, features from raw data can be extracted and it will benefit in the processing of complex classification.

# 3. Objective

The objective of our project is to recognize the Unnatural Control Chart Patterns th at occur on statistical quality Control charts to detect the unnatural deviation in stat e of process as well as to identify quality failure and root abnormal cause in time.

# 4. Methodology

We have divided the methodology in two parts i.e; data preparation using data sim ulation and Feature extraction using Deep neural networks.

## 1) Data Preparation using Monte carlo simulation

**Data simulation** is one of the widely used techniques for control chart pattern rec ognitions. It is the process to generate thousands of random samples following a p articular distribution using original data. In this project we have used raw eye tracki ng datasets and obtained its distribution, mean and variances to generate data of various patterns.

### ➢ Monte Carlo Simulation

In this project We have used Monte Carlo simulation of Data simulation on eye trac king datasets and its obtained the mean and variances and generate data of vario us patterns by changing various parameters as shown in the Table1 :

Table1: Parameters and formulas of Data simulation

| class | Description | equations | Remarks |
|---|---|---|---|
| 0 | Normal, NOR | $y_t=\mu + r(t) \times \delta$ | $\mu = 0,\ \sigma = 1$ |
| 1 | Cyclic, CYC | $yt = \mu + r(t) \times \delta + a \times sin(2\pi/T)$ | $r(t) \sim N(0,1)$ |
| 2 | Systematic, SYS | $yt = \mu + r(t) \times \delta + d \times (-1)t$ | $\delta = 1\sigma$ |

| | | | |
|---|---|---|---|
| 3 | Stratification, STR | $yt = \mu + r(t) \times \delta'$ | $\delta' \in (\mathbf{0.2\sigma}, \mathbf{0.4\sigma})$ |
| 4 | Upward Trend, UT | $yt = \mu + r(t) \times \delta + t \times g$ | $d \in (1\sigma, 3\sigma)$ |
| 5 | Downward Trend, DT | $y_t = \mu + r(t) \times \delta - t \times g$ | $a \in (1.5\sigma, 2.5\sigma)$ |
| 6 | Upward shift, US | $yt = \mu + r(t) \times \delta + k \times s$<br>$k = 1\ if\ t \geq P, else = 0$ | $T = 16$<br>$g \in (0.005\sigma, 0.25\sigma)$ |
| 7 | Downward Shift, DS | $yt = \mu + r(t) \times \delta - k \times s$<br>$k = 1\ if\ t \geq P, else = 0$ | $P \in (10,20)$<br>$s \in (1\sigma, 3\sigma)$<br>$t = 1,2,3,\ldots.L$ |

- $\mu$ and $\sigma$ = mean and standard deviation estimate of the in-control production process.
- $r(t)$ represents the inevitable accidental fluctuation which is subject to gaussian di stribution $N(0,1)$.
- $d$ = Degree of system state departure.
- $a$ =Amplitude of cyclic pattern.
- $T$ = Period of the cycle.
- $g$ = Gradient of a data trend.
- $P$ = Time when the shift anomaly occurs.
- $s$ = Amplitude of the shift pattern.

The sequence (window width) length of data simulation 'L' should be small, because long er width of window causes larger lag of anomaly detection. In General length of the windo w is set to "16-
64" sampling points. The flowchart monte carlo simulation is shown in figure1 below:

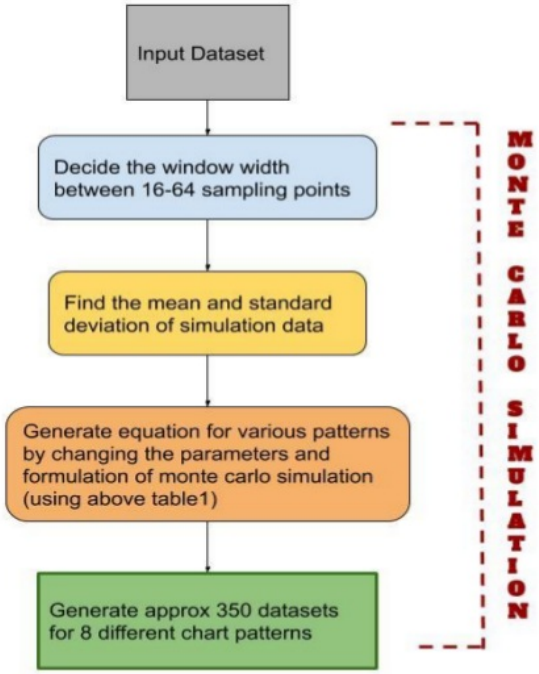
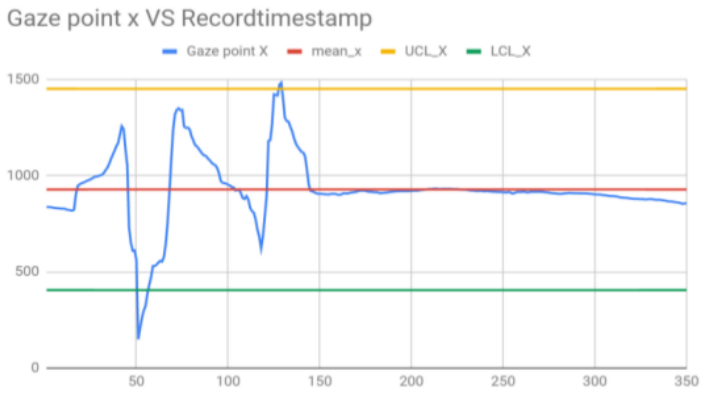



**Figure1: Flowchart for Monte carlo simulation**

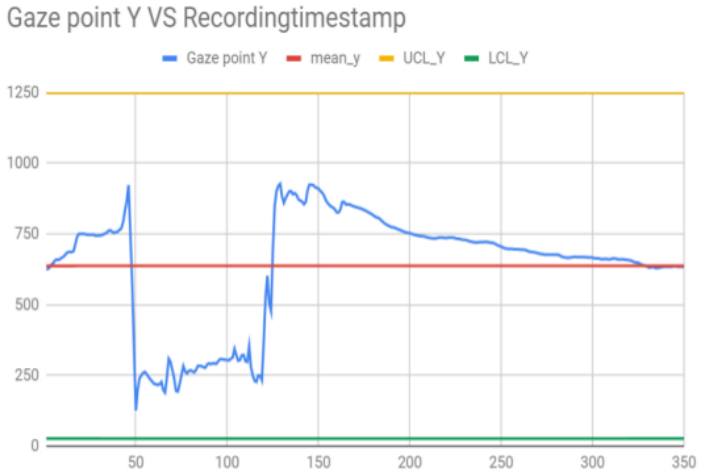
1. For the given sample of eye tracking gaze point datasets shown in the table2, we have done Data Simulation using monte carlo simulation on approx 10,944 raw ey e gaze points data and prepared 8 different control chart pattern datasets with win dow length of 32 sampling points i.e; we make 342 datasets from 10944(342*32) d ata having 32 sampling points each.
2. For that first dataset we took the first 32 sampling points and generated 8 different patterns datasets using monte carlo simulation of the same window length of 32 sa mpling points.
3. Similarly we have done for next dataset having 32 sample length, and like that fro m 10944 (342*32) eye tracking data we have prepared 342 pieces for each 8 cont rol chart patterns having sample length of 32.
4. We have written code in python using pandas and numpy for monte carlo simulatio n.

## Visualization of raw eye tracking datasets

### I. Gaze point X VS Recording timestamp

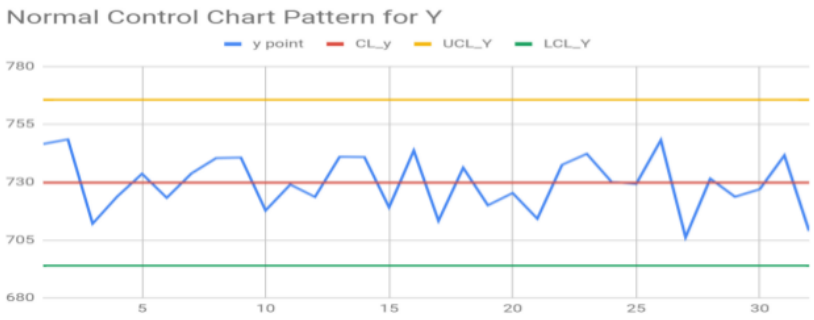


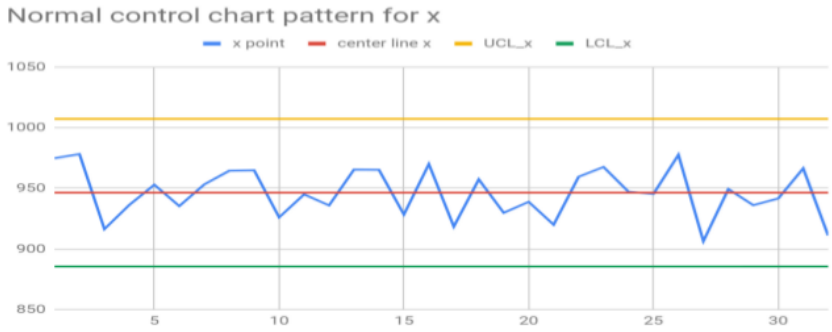

### II. Gaze point Y VS Recording timestamp



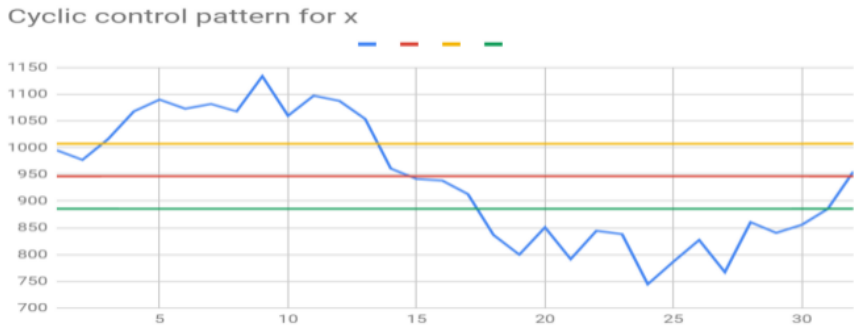

## Visualization of Generated data of control chart patterns

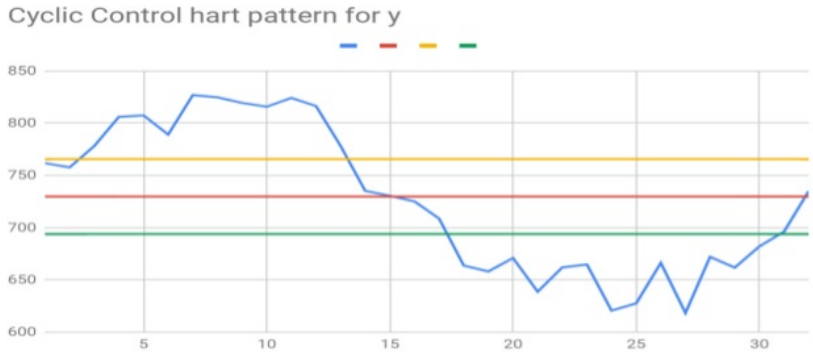
### I. Normal Patterns



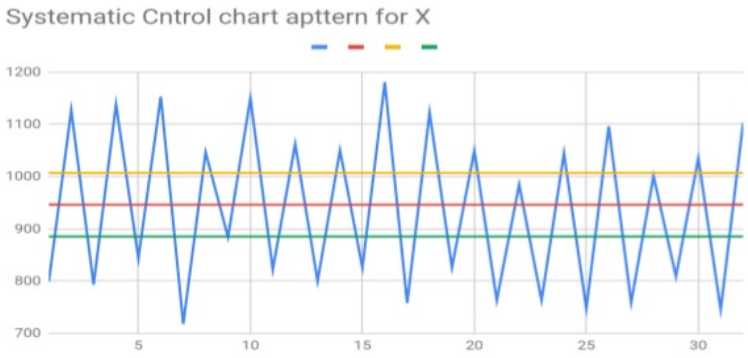



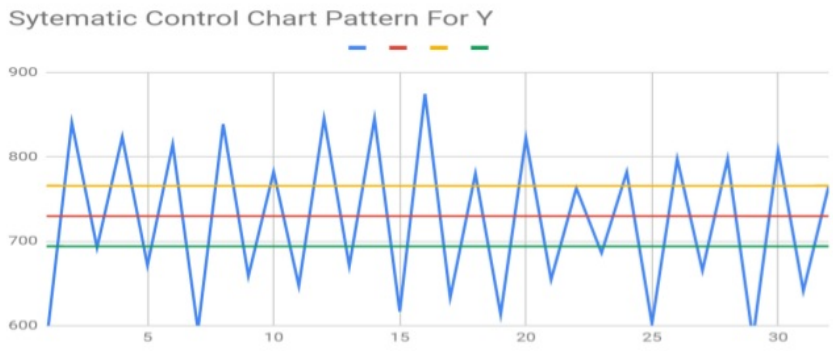

## II. Cyclic patterns



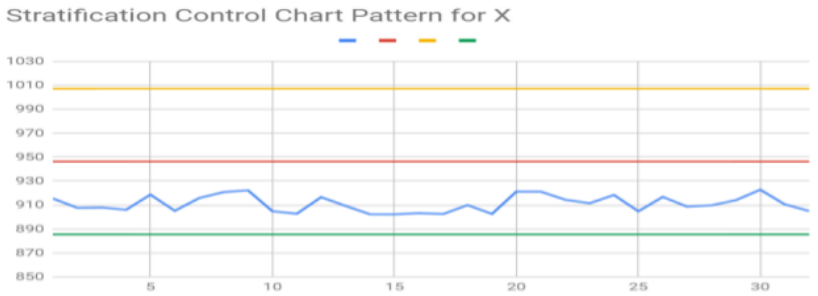


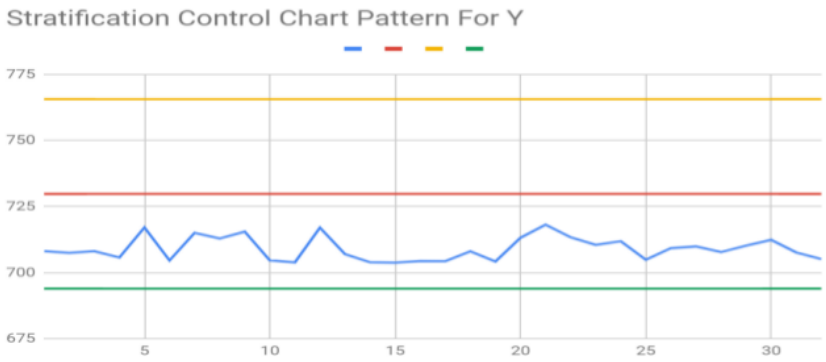

## III. Systematic patterns



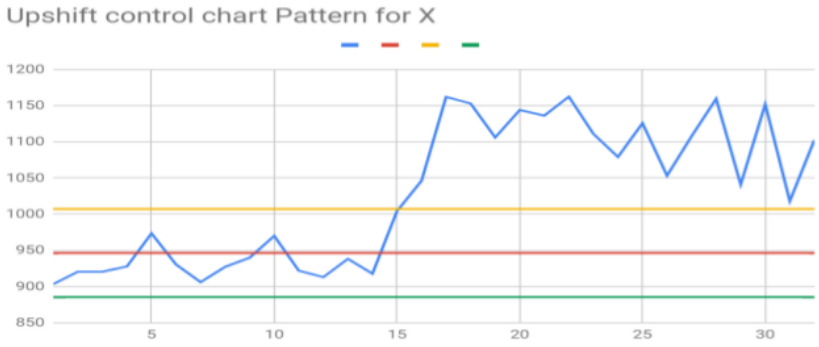



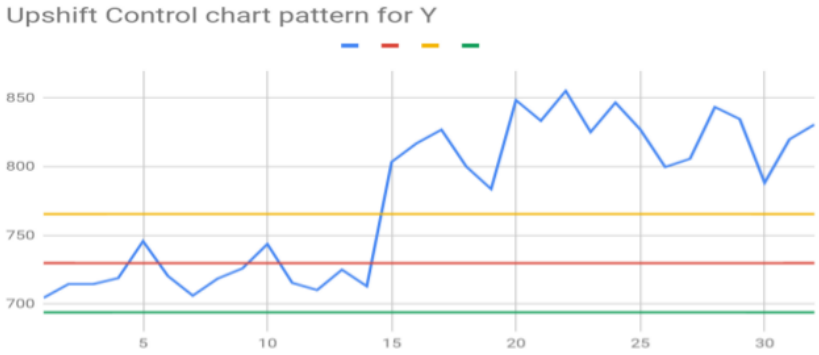

## IV. Stratification patterns



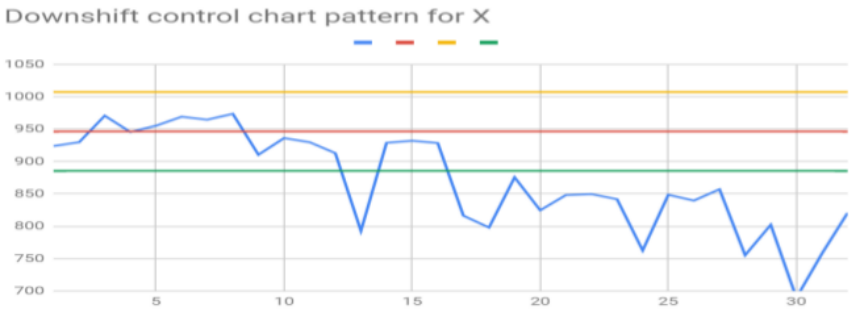



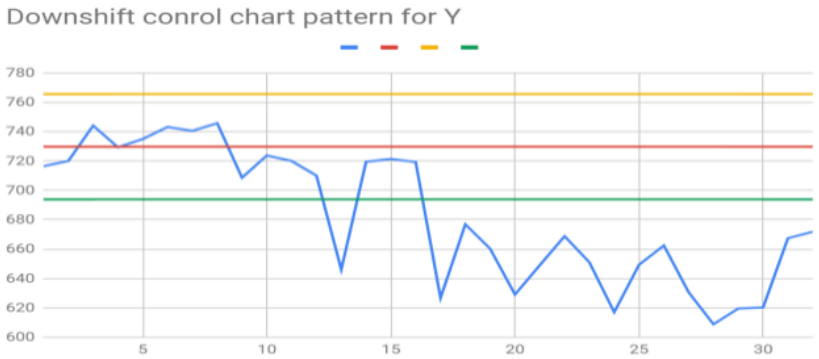

## V. Upshift patterns



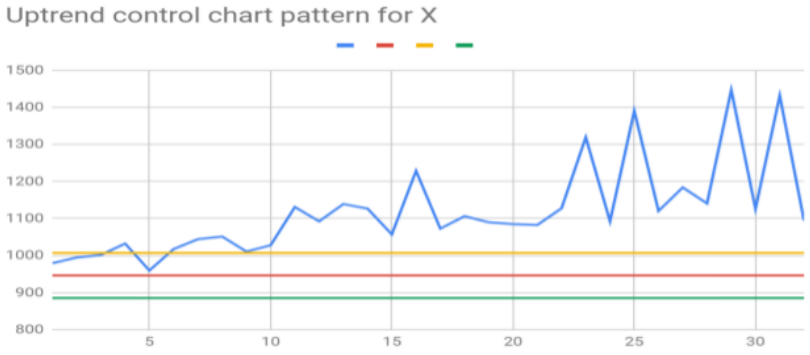


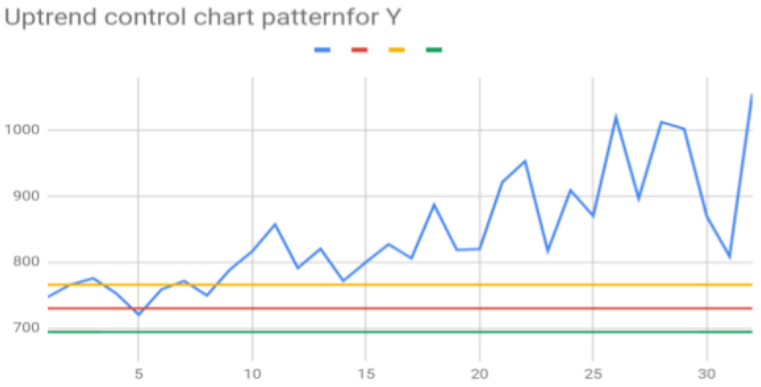

## VI. Downshift patterns



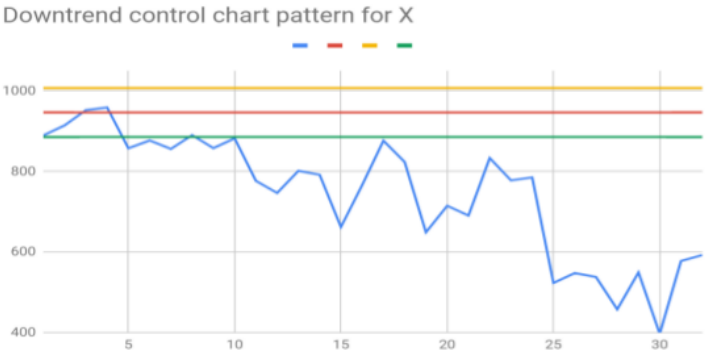



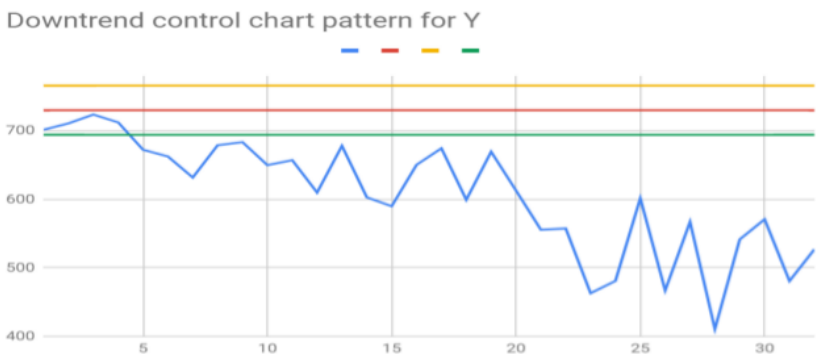

## VII. Uptrend patterns




## VIII. Downtrend patterns






## 2) Features extraction using Deep neural Networks

Feature extraction is the second step after data generation for different patterns us ing monte carlo simulation. We have used various neural networks and 1 dimensio nal convolutional neural networks to extract the feature from the generated data f or different patterns.

### How Convolutional neural networks work ?

The structure of convolutional neural networks is divided into two parts, The first p art is convolutional layers and pooling layers to extract features and generate feat ure maps, and the second part is fully connected(dense) layers for final output.

- The convolutional layers: Extract features from the input data and generate featur e maps.
- The fully connected(dense) layers: Uses feature maps from convolutional layer to generate output

There are two important processes involved during the training of deep neural network:

I. **Forward propagation**: Receive input data, process the information, and generate output
II. **Backward propagation**: Calculate error(cost function) and update the parameter s of the neural network.

## Forward Propagation

**A. Forward Propagation: Convolutional layer**

Each Convolutional neural network has filters to extract the local features from input data, these lo

cal extracted features are then fed into activation units to generate the output feature maps . There are various types of activation functions, few of them are sigmoid function, Tanh function, ReLU function, Leaky ReLU, Softmax.

I have used mainly ReLU activation functions on hidden layers and Softmax activation function of output layer. The advantage of using ReLU activation function over others is that it does not activate all the neurons at the same time. After applying ReLU functions the neurons will be deactivated if the output value of the linear transformation is less than zero.

$$\square(\square) = \square\square\square(\square, \square)$$

To find each unit generated feature map the weight of filters used are the same known as weight sharing. And the number of features used in the network determines the depth of the feature maps and thus the number of features plays an important role in the performance of any CNN. After convolution, it is then connected to the pooling layer.

The pooling layer is used to reduce the size of extracted feature maps and usually set after the convolutional layer. Max pooling and average pooling are two most widely used pooling methods. Max pooling is used to reduce the size of data by picking the maximum value from the elements in the window whereas average pooling takes the average value of all the elements in the window.

The computation graph of forward propagation is shown the figure below:

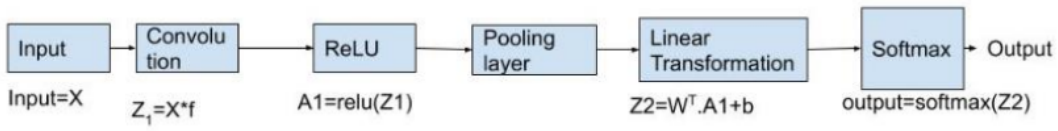



**Figure2: Computation Graph for Forward Propagation**

**Step-**
**1 If □ is our generated input data and □ is the filter then our generated data is convolved with the filter and expression would be:**

$$\square\square = \square * \square$$

**Step-2 Applying ReLU activation function on extracted local feature maps matrix ($Z_1$).**

$$\square\square = \square\square\square\square(\square_{\square})$$

**Step-3 Set Pooling Layer (we have used Max pooling layer) after convolutional layer.**

**B. Forward Propagation: Fully connected layer**

Convolutional layer has extracted some valuable features from the input data. After step-3, Now these extracted features are sent to the fully connected layer that generates the final output. The output from the convolutional layer is a 2D matrix then the generated feature maps from the convolutional layer are first converted into a 1 dimensional array, once the generated data is converted into 1D array, it is sent to the fully connected layer. All of these individual values are treated as separated features.

Fully connected layer performs two operations on the incoming data-
first is linear transformation and second is non-linear transformation.

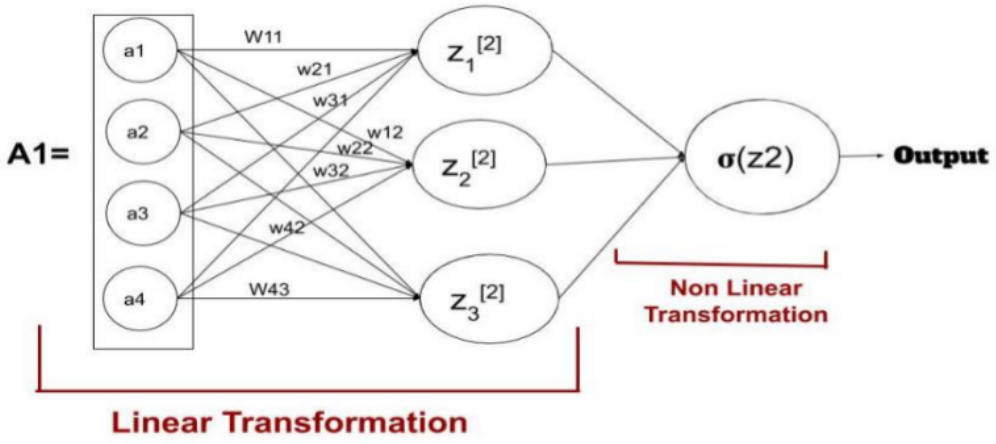



**Figure3: Linear and nonlinear transformation in fully connected layer**

**Step-4 defines (randomly initialize) weight and bias matrix and applies linear transformation on the values.**

$$A1 = \begin{bmatrix} a1 \\ a2 \\ a3 \\ . \\ . \\ an \end{bmatrix} \quad W = \begin{bmatrix} W11 & W12 & . & . & . & W1m \\ W21 & W22 & . & . & . & W2m \\ W31 & W32 & . & . & . & W3m \\ W41 & W42 & . & . & . & W4m \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ Wn1 & Wn2 & . & . & . & Wnm \end{bmatrix} \quad b = \begin{bmatrix} b1 \\ b2 \\ b3 \\ . \\ . \\ bm \end{bmatrix}$$

The equation for linear transformation is:

$$\square\square = \square^{\square}.\ \square\square + \square$$

Here, A1 is the extracted local feature map obtained from step-3, W is a weight matrix, and b is a bias matrix which is constant.

**Step5-Apply softmax activation function on Z2**

Now the final step in the forward propagation - the non linear transformation.

The linear transformation individually cannot capture all the complex relationships and th us to capture those relationships we introduced activation function in the network which a dds non-
linearity to the data**.** We have used the **Softmax activation** function in the output layer u sed for multiclass classification problems which return probability of a dataset belonging t o each class.

$$\square\square\square\square\square\square\square = \square(\square)_{\square} =$$

$$\frac{\square^{\square_{\square}}}{\sum_{\square=1}^{\square} \blacksquare\square^{\square_{\square}}} \text{ for } j = 1\ldots\ldots, K$$

Applying softmax activation function, This will be our final output

$$\square = \square\square\square\square\square\square\square(\square 2)$$

## Backward Propagation

During the forward propagation process, the parameters of convolutional neural networks are randomly initialized weight, bias and filters. In the backward propagation process, the model tries to update all these parameters to decrease the overall loss and make the model more accurate. We have used the concept of gradient descent techniques for updating the parameters which find the value of parameters at which loss is minimum. The general equation for updating the parameters is:

**NewParameter = OldParameter-(learning_rate*gradient_of_parameter)**

The learning rate is a constant value which determines the amount of change needed to the old value of parameter and slope or the gradient to determine whether the values should increase or decrease. In order to update the old value of parameters we need to find the gradient of parameters that is change in error with respect to parameters.The computational graph of backward propagation is shown in the figure below:

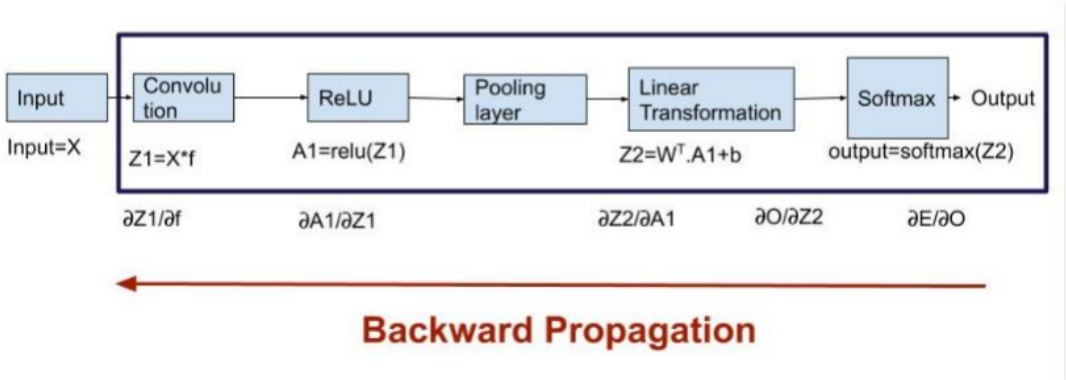



**Figure4: Computation graph for Backward Propagation**

**Backward propagation: Fully Connected Layer**

There are two parameters in a fully connected layer - weight matrix and bias matrix.

$$\frac{\partial E}{\partial W} = \frac{\partial E}{\partial O} \cdot \frac{\partial O}{\partial Z2} \cdot \frac{\partial Z2}{\partial W}$$

Where

□□/□□= change in error with respect to output

□□/□□2= change in output with respect to Z2

□□2/□□= change in Z2 with respect to W(weights)

The shape of $\partial E/\partial W$ and the weight matrix W will be the same. Now we update the old weight matrix $W_{old}$ using equation given below:

$$□_{□□□} = □_{□□□} − □□ ∗ □□/□□$$

Similarly we will update bias using following equation:

$$□_{□□□} = □_{□□□} − □□ ∗ □□/□□$$

**Backward Propagation: Convolution layer**

The parameters for the convolution layer is a filter matrix which we had randomly initialize d during the forward propagation process. Now we are going to update these values usin g the following equation.

**New parameter = Old parameter-(learning rate*gradient of parameter)**

To update the filter matrix, we need to find the gradient of the parameter dE/df.

$$□□/□□ = □□/□□.□□/□□.□□□/□□.□□□/□□□.□□□/□□$$

Where

□□$_2$/□□$_1$= change in $Z_2$ with respect to A1

□□$_1$/□□$_1$= change in $A_1$ with respect to Z1

$\partial Z1/\partial f$ = change in Z1 with respect to f

Now after finding the value of $\partial E/\partial f$, we are going to use this new value to update the original(older) filter value:

$$□_{□□□} = □_{□□□} - □□ * (□□/□□)$$

In this project we have generated 342 pieces for each pattern using Data Simulation techniques, 300 of which were used for training the 5 different neural network models and 42 for testing the models.

- Training Data- (300*32*2), label-(nor,cyc,sys,str,us,ds,ut,dt)
- Test Data- (42*32*2), label-(nor,cyc,sys,str,us,ds,ut,dt)
- label-(nor,cyc,sys,str,us,ds,ut,dt)-encode-(0,1,2,3,4,5,6,7)

We have used Five different neural networks for recognition of our control chart patterns and applied a keras tuner to find the optimized number of units and filter size and trained each model for around 250 epochs.

1. Artificial neural network
2. 1 layer 1-D CNN
3. 2 layer 1-D CNN
4. 3 layer 1-D CNN
5. Improved 1-D CNN (having inception layer)

The architecture and structure of each neural network used for control chart patterns recognition is shown in the table2 below :

| Layer | ANN | 1L 1-D CNN | 2L 1-D CNN | 3L 1-D CNN | Improved 1-D CNN |
|---|---|---|---|---|---|

| **Input Layer** | 32*2 | 32*2 | 32*2 | 32*2 | 32*2 |
|---|---|---|---|---|---|
| **layer1** | Flatten | Conv1D(1*3,112) | Conv1D((1*10,128) | Conv1D((1*3,80) | conv1D (1*10,16),(1*10,32), (1*10,64) |
| **layer2** | Dense(448) | Max Pooling (pool_size=2) | Max Pooling (pool_size=2 | Max Pooling pool_size=2 | Concatenate |
| **layer3** | Dense(448) | flatten | (1*10,128) | (1*3,112) | (1*10,32) |
| **layer4** | ---------- | Dense(80) | Max Pooling (pool_size=2 | Max Pooling (pool_size=2) | Max Pooling pool_size=2 |
| **layer5** | ---------- | Dense(8) | flatten | (1*3,64) | (1*10,112) |
| **layer5** | ---------- | ---------- | Dense(80) | Max Pooling (pool_size=2 | Max Pooling pool_size=2 |
| **layer6** | ---------- | ---------- | Dense(8) | flatten | flatten |
| **layer7** | ---------- | ---------- | ---------- | Dense(48) | Dense(80) |
| **layer8** | ---------- | ---------- | ---------- | Dense(8) | Dense(8) |

In this project along with ANN and single or multi layer 1 dimensional ANN, we have also used a special type of CNN model called **Improved 1 dimensional CNN** which has a special layer called Inception layer as a layer1 which is a parallel combination of three layers of filter size (1*10) and 16, 32, 62 filters. The main advantages of having an inception layer is It allows the internal layers to pick and choose which filter size will be relevant to learn the required information. The architecture of Improved one dimensional CNN is shown in the figure5 below.

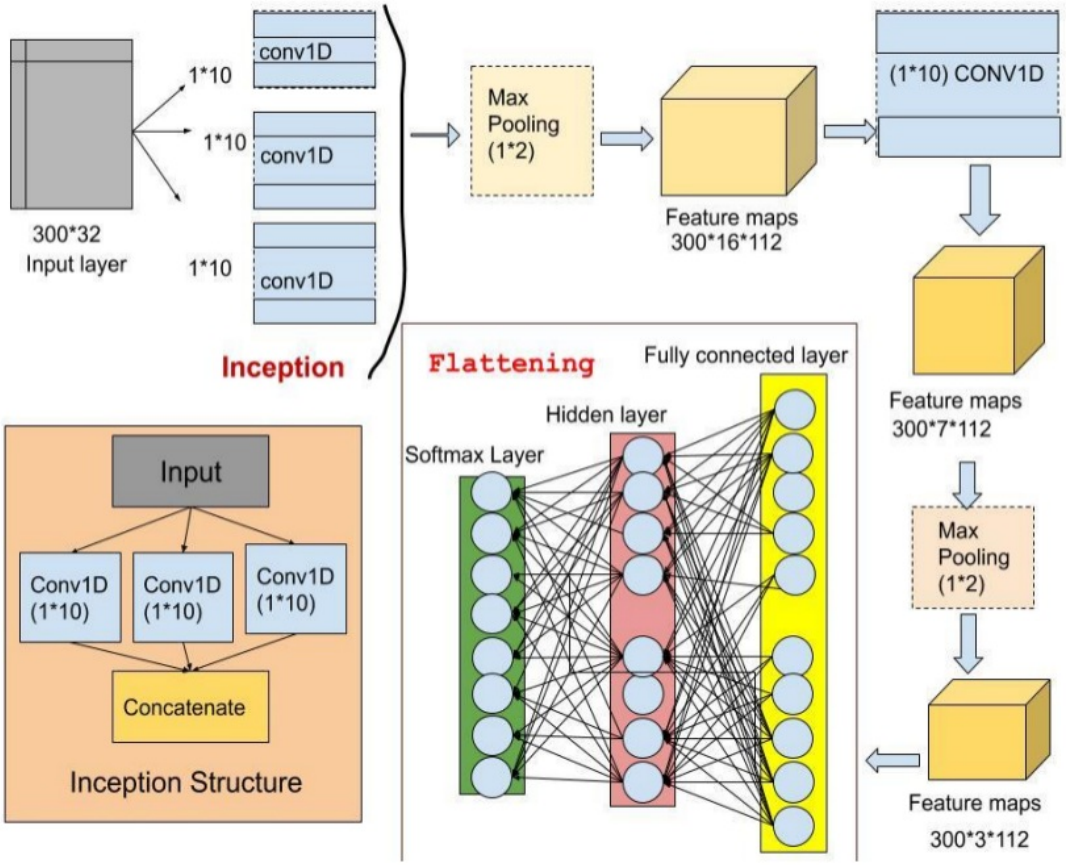
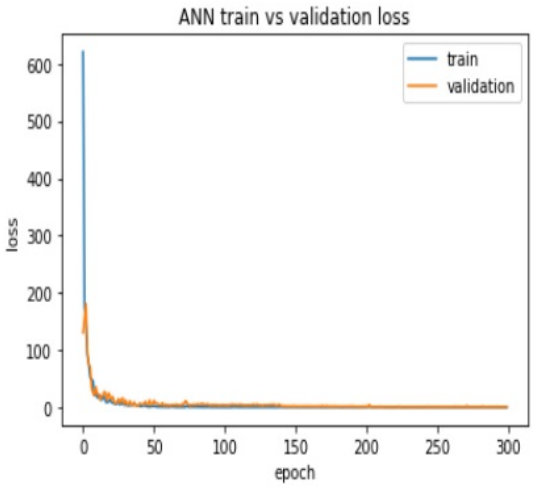



Figure5: Architecture of Improved one dimensional convolutional neural network

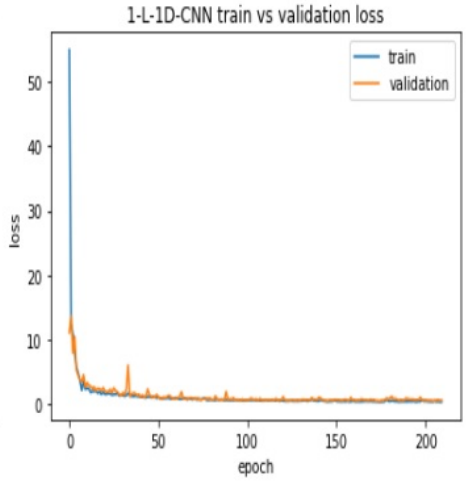
# 5. Results

After training the model for around 250 epochs, we have tested the test data on th e trained model and compare the recognition accuracy, loss, plots and other factor s.

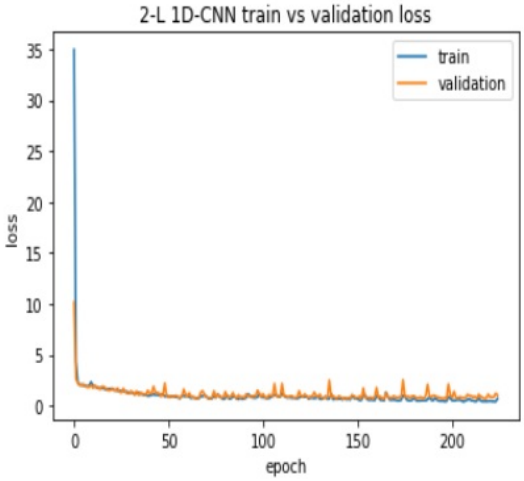
1) **Graph between training and Validation loss of models per epoch**
The following graph shows how the loss is decreasing with the increase in the num ber of epochs.



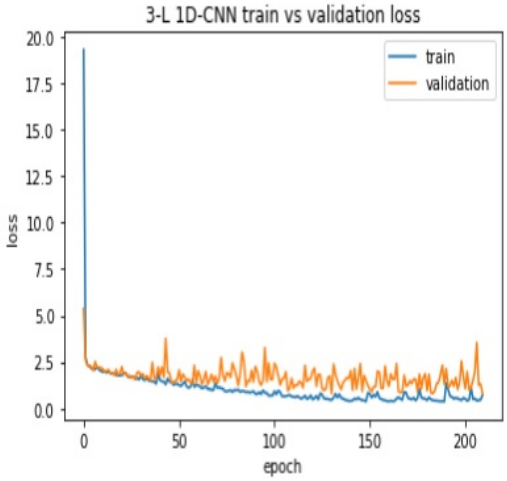

**(I) ANN**



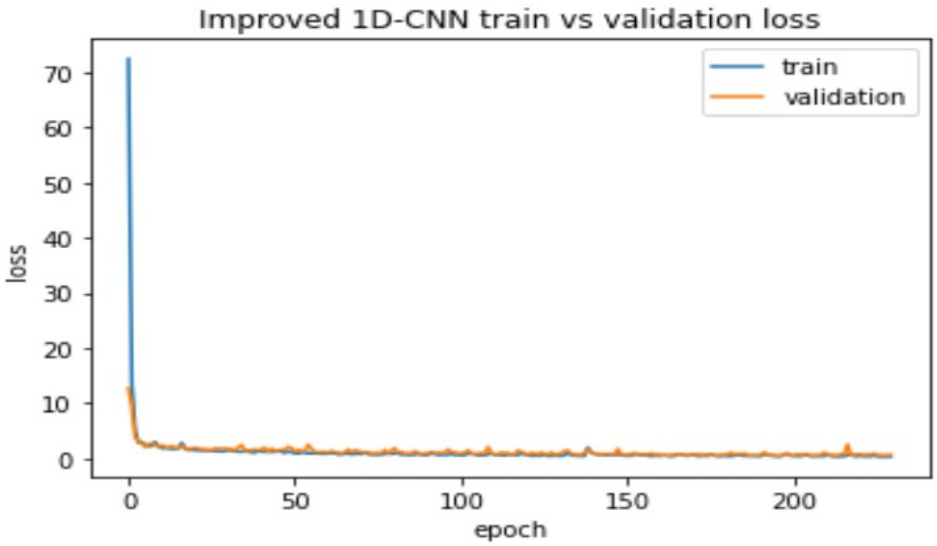

**(II) 1layer 1D**




(III) 2 layer 1D CNN



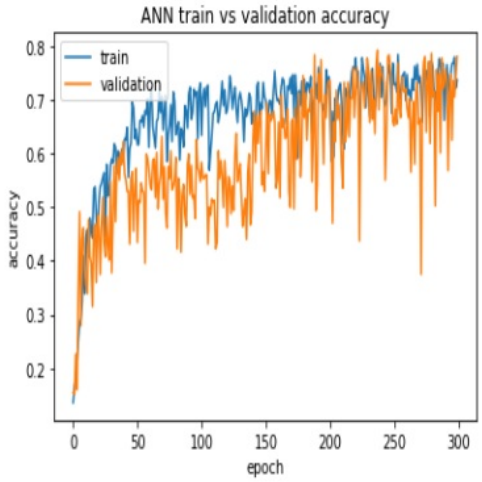

(IV) 3 layer 1D CNN



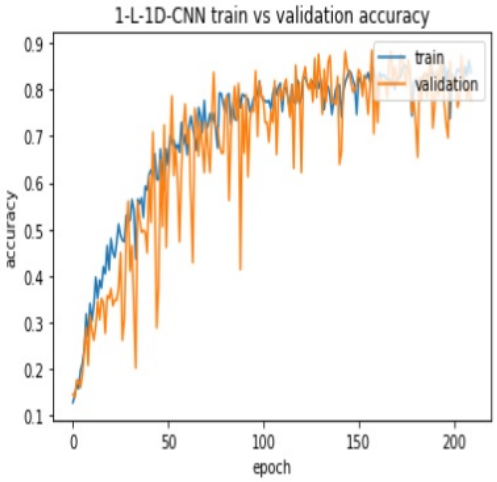

(V) Improved 1D CNN

**2) Graph between training and validation accuracy of models per epoch**

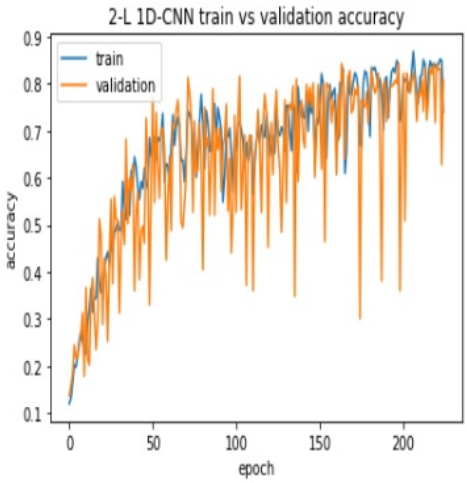
The following graphs show how the accuracy of the model increases per epoch du ring the training process.



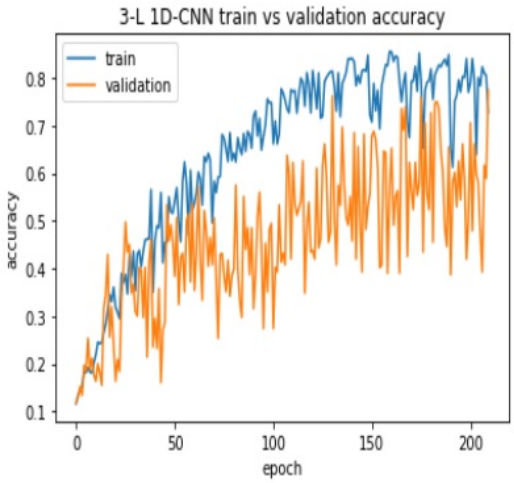

(I) ANN



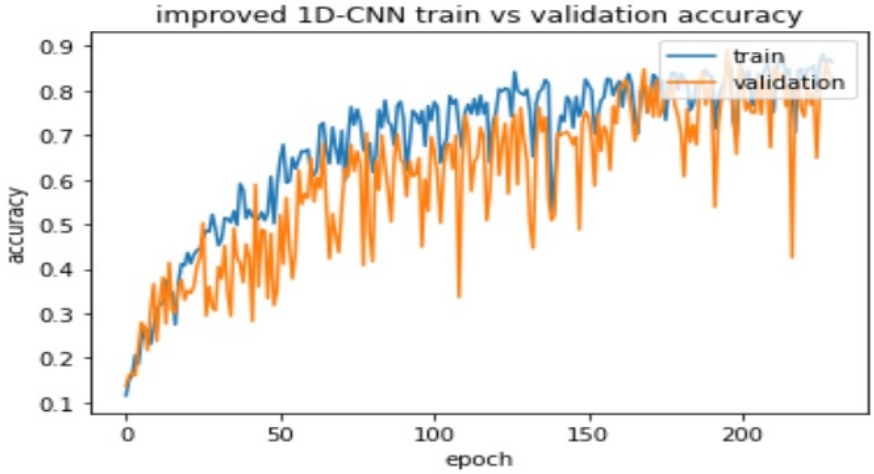

(II)1 layer 1D CNN




(III) 2 layer 1D CNN




(IV) 3 layer 1D CNN

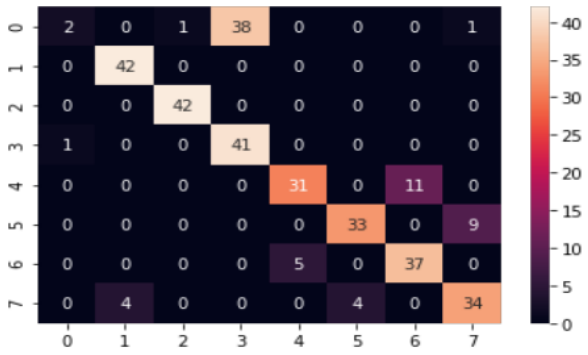



(V) Improved 1D CNN

### 3) Confusion matrix

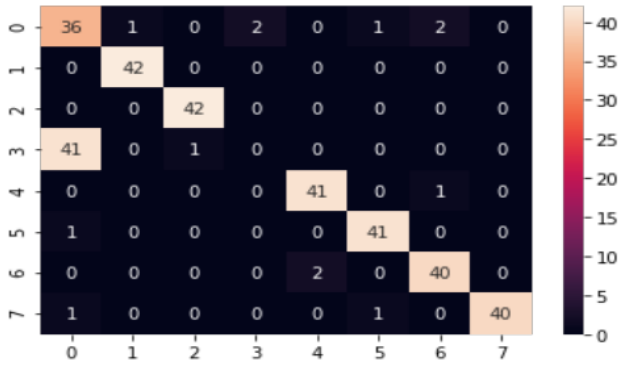
A confusion matrix is used to check the performance of a classification model on a set of tes t data.Calculating a confusion matrix can give you an idea of where the model is right and w hat types of errors it is making.

1) ANN



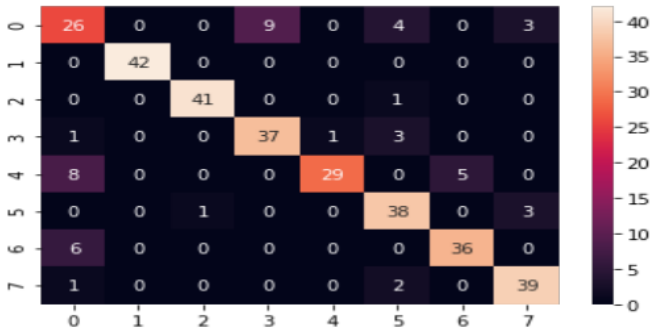

The confusion matrix heatmap for ANN shown in the above figure gives the recognition ac curacy of 77.38% which is lowest among all models and misclassification of 22.62% which is highest among all models.

2) 1 layer 1D CNN




The confusion matrix heatmap for 1 layer 1D CNN shown in the above figure gives the acc uracy of 83.9% and misclassification of 16.9%.

3) 2 layer 1D CNN

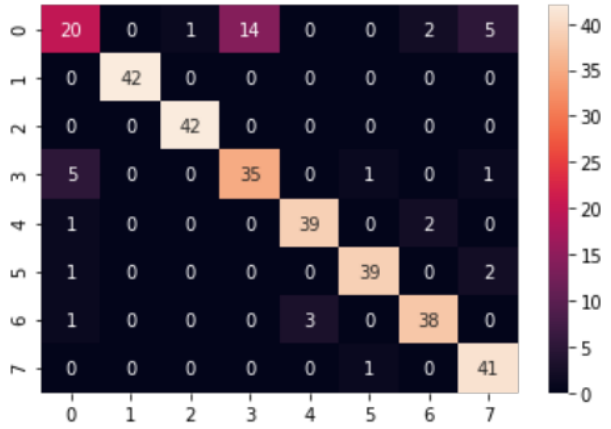



The confusion matrix heatmap shown in the above figure gives the recognition accuracy of 85.7% and has misclassification o0f 14.3%.

4) 3 layer 1D CNN


The confusion matrix heatmap shown in the above figure gives the recognition accuracy of 78.6% which is lowest after artificial neural networks and also has misclassification of 21.4 % which is highest after ANN .

5) Improved 1D CNN




Among all CCPs models, Improved 1D CNN has the highest accuracy. Confusion matrix h eatmap shown in the above figure gives the recognition accuracy of 88.09% which is highes t and has lowest misclassification value of 11.91%.

### 4) Accuracy

| Model | ANN | 1 layer 1D CNN | 2 layer 1D CNN | 3 layer 1D CNN | Improved 1-D CNN |
|---|---|---|---|---|---|
| Accuracy on train da ta | 0.8062 | 0.84 | 0.889 | 0.835 | 0.90625 |
| Accuracy on test dat a | 0.786 | 0.839 | 0.857 | 0.7738 | 0.881 |

### 5) Loss

| Model | ANN | 1 layer 1D | 2 layer 1D | 3 layer 1D | Improved |
|---|---|---|---|---|---|

| | | CNN | CNN | CNN | 1-D CNN |
|---|---|---|---|---|---|
| **Loss on train data** | 0.388 | 0.30 | 0.2955 | 0.33 | 0.2466 |
| **Loss on test data** | 0.6682 | 0.336 | 0.3122 | 0.69 | 0.2761 |

# 6. Discussion

We have compared the accuracy and loss of five different models for recognition of control charts patterns to detect or identify the unnatural patterns and deviations occurring on control charts during the production process. These five different types of model are Artificial neural network, 1 layer 1D convolutional neural network, 2 layer 1D convolutional neural network, 3 layer 1D convolutional neural network and Improved 1D convolutional neural network. The model which has only one convolutional layer is 1 layer convolutional neural network and if the model has 2 layer of convolutional layer is 2 layer convolutional layer similarly for 3 layer convolutional layer. Improved 1D convolutional neural network has a special type of layer called Inceptional layer. We have trained these models on 300 training datasets for each pattern that is normal pattern, cyclic pattern, systematic pattern, stratification pattern, upshift pattern, downshift pattern, uptrend pattern, downtrend pattern and test the model on 42 test datasets for each pattern. These datasets are generated by applying monte carlo simulation on raw eye tracking datasets. While training the model, the model undergoes forward and backward propagation to give the final output. After training each model for around 250 epochs, we fed the validation data and got accuracy and loss for each model. After analyzing the results like accuracy , loss confusion matrix and all the plots of loss and accuracy for training and validation datasets, Improved 1D-
CNN has highest recognition accuracy of 90% on train data and 88% on test data as well as lowest loss of 0.2466 on train datasets and 0.27 on test datasets where as Artificial neural network has minimum recognition accuracy of 80.62% on train data and 77.38% on test data as well as maximum loss 0.388 on train data and 0.69 on test datasets. This is due to the presence of the inception layer in Improved 1-
D CNN allows the inner layer to pick the optimized filter size to learn the patterns in the data.
All these feature-
based extraction methods help to develop automated recognition systems for cont

rol chart patterns recognition and thus when we feed the datasets, the model helps to identify or detect abnormal patterns, deviations and abnormal abruptions as well, in the production process. Abnormal patterns formed in control charts are related with various assignable causes which greatly affect the stability of the production process and thus the recognition of these patterns can help us to find those causes and eliminate the potential hazards caused by these factors to make our production process smooth.

# 7. Conclusion

Feature-
based extraction methods like convolutional neural networks are very powerful techniques for the recognition of control chart patterns.The results indicate that feature-
based Feature extraction methods like 1D convolutional neural network having inception layer give more consistent recognition performance and dominance over layer by layer neural network and some classical methods like fuzzy inference systems, support vector machines. After analyzing the confusion matrix heatmap for all proposed neural network models in our thesis, there is a tendency for stratification patterns to be mostly confused with normal patterns and similarly shift patterns with trend patterns. But out of all five models used in this thesis Improved 1D convolutional neural network gives highest accuracy and reduces the misclassification between stratification-normal patterns and trend-
shift patterns. This indicates that the performance of our recognition model can be improved further by identification of new features that will be helpful in discriminating normal pattern with stratification pattern as well as shift pattern with trend pattern. And thus these efficient automated CCP recognition systems can help to identify eight most common control chart patterns that are normal pattern, stratification pattern, cyclic pattern, systematic pattern, upward shift pattern, downward shift pattern, Utrend pattern and downtrend pattern. After the recognition of pattern , it informs the users about various root assignable causes associated with pattern along with the necessary preemptive actions also reduces the complexity of the production process and helps in judging whether the process is normal or abnormal and the recognition of unnatural patterns in control charts provides clue to reveal the potential quality problem in manufacturing process.

# 8. References


1) Pham, D. T., and Wani, M. A., (1997). Feature-based control chart pattern recognition. International Journal of Production Research, 35(7), 1875-1890
2) Gauri, S. K., and Chakraborty, S. (2007). A study on the various features for effective control chart pattern recognition. The International Journal of Advanced Manufacturing Technology, 34(3-4), 385-398.
3) Addeh, A., Khormali, A., and Gorillaz, N. A. (2018). Control chart pattern recognition using RBF neural networks with new training algorithms and practical features. ISA Transactions, 79, 202-216
4) Gulbay, M., and Kahraman, C. (2007). Development of fuzzy process control charts and fuzzy unnatural pattern analyses. Comp
5) Zaman, M., and Hassan, A., (2018). Improved statistical features-based control chart patterns recognition using anfis with fuzzy clustering. Neural Computing and Applications, (4), 1-15.
6) Ebrahimzadeh, A., and Ranaee, V. (2011). High efficient method for control chart patterns recognition. Acta technica ČSAV, 56(1), 89-101.
7) Zhao, C., Wang, C., Hua, L., Liu, X., Zhang, Y., and Hu, H. (2017). Recognition of control chart pattern using improved supervised locally linear embedding and support vector machine. Procedia Engineering, 174, 281-288.
8) Cheng, and C.-S. (1997). A neural network approach for the analysis of control chart patterns. International Journal of Production Research, 35(3), 667-697
9) Ghomi, S. M. T. F., Lesany, S. A., and Koochakzadeh, A. (2011). Recognition of unnatural patterns in process control charts through combining two types of neural networks. Applied Soft Computing, 11(8), 5444-5456
10) Awadalla, M. H., and Sadek, M. A. (2012). Spiking neural network-based control chart pattern recognition. Alexandria Engineering Journal, 51(1), 27-35
11) Kiranyaz, S., Ince, T., and Gabbouj, M. (2016). Real-time patient-specific ECG classification by 1-d convolutional neural networks. IEEE Transactions on Biomedical Engineering, 63(3), 664-675
12) Malek, S., Melgani, F., and Bazi, Y. (2017). One -dimensional convolutional neural networks for spectroscopic signal regression. Journal of Chemometrics, 32(5), e2977

# Internship Report

ORIGINALITY REPORT

| 1% | % | % | % |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Bingsheng He, Qiong Luo. "Cache-oblivious nested-loop joins", Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06, 2006<br>Publication | 1% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | Off |
| Exclude bibliography | On | | |