ccps using cnn

by Aditya Prakash

Submission date: 17-Sep-2020 04:02PM (UTC+0530)
Submission ID: 1389348060
File name: Control_Chart_Patterns_Recognition_Using_CNN_Final_Report.pdf (1.68M)
Word count: 5078
Character count: 26699

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 predominantly in the form of X chart, are widely used to recognize situations when manufacturing systems need control actions. The 8 fundamental control patterns are, normal (NOR), cyclic (CYC), systematic (SYS), stratification (STA), uptrend (UT), downtrend (DT), upward shift (US), and downward shift (DS).

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

2. **Systematics Pattern(SYS)**: The systematic pattern appears as a high point always follows a low point and vice-versa, thus point-to-point fluctuations can be predicted by SYS.

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

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

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

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 unnatural pattern identifies a process when it is out of control. The abnormal CCPs in the production process usually correspond to some specific causes. Hence, the recognition of abnormal patterns can help to find the problems in time and the scope of abnormal causes can be narrowed as well. Earlier in control chart applications, human experience was required to deduce whether the production process is abnormal or not and if it is found abnormal, 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 based on small probability events, which can be easily carried out. However, covering all abnormal patterns with rules is quite difficult due to the complexity of the production process. To compensate this gap, an efficient automated control chart pattern (CCP) recognition system can be implemented, which ensures consistent, unbiased interpretation of CCPs leading to lesser number of false alarms and easier execution of control charts.

The advantage neural networks provide is that the provision of explicit rules or templates is not required here. Rather, it learns to recognize patterns directly through typical example patterns during a training phase and has the ability to identify an arbitrary pattern not previously encountered. However, there is no guarantee that a neural network will identify such patterns correctly. One disadvantage with neural networks is that the information it contains is implicit and virtually inaccessible to the user.

2. Literature Review

To recognize CCPs, many methods have been applied. Some of them directly feed the raw CCPs data into the recognition model, while others first use statistical knowledge such as mean square amplitude, standard deviation, peak, average, etc. to extract features from raw data, then feed these extracted features into the model.

Pham et al. (1997) propose that different CCPs have different geometric characteristics, hence by the help of constructing features, CCPs recognition accuracy can be improved. Even so, there are lots of different features that can be constructed, making it a challenge to select the appropriate feature subset so as to reduce the computational complexity and to improve the accuracy of classification models. Gauri et al. (2007) applied classification and regression tree (CART) to select feature subset. Addeh et al. (2018) applied the association rules (AR) to select the best feature subset. But some features selected by these methods are still highly correlated, leading to redundancy. Thus, to reduce redundancy between features, dimension reduction methods such as independent component analysis (ICA) (Lu et al., 2011) and principal component analysis (PCA) (Tai-Fu et al., 2012) are used. After computing input form of the data, whether it is raw or extracted feature data, a model will be built to realize accurate recognition of different CCPs. Some classical methods such as fuzzy inference systems (FIS), support vector machines (SVM), artificial neural networks (ANN) are applied in CCPs recognition systems.

A fuzzy method for recognition of unnatural CCPs is proposed by Gulbay et al. (2007). Zaman et al. (2018) combined the fuzzy c mean (FCM) with adaptive neuro-fuzzy inference system (ANFIS) to realize the CCPs recognition and to get comparable classification accuracy. Ebrahimzadeh et al. (2011) applied the SVM method to CCPs recognition for its excellent generalization performance. Though, appropriate parameters selection for SVM is a bit difficult. Therefore, to optimize the parameters of SVM models automatically, some optimization algorithms such as genetic algorithms (GA) (Zhao et al., 2017) and particle swarm optimization (PSO) (Yongman et al., 2013) are used. In connection to it, many studies introduced ANN into the CCPs recognition. By using multilayer perceptron (MLP), Cheng et al. (1997), constructed a modular neural network, trained it applying back-propagation (BP) algorithm and realized the recognition of two kinds of CCPs. The CCPs recognition module designed by Ghomi et al. (2011) combines two types of neural network, one is multilayer perceptron whereas another one is learning vector quantization (LVQ). Spiking neural network(SNN) to the CCPs recognition is applied by Awadalla et al. (2012), which considered the continuity 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, since 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 structure (Zhang et al., 2018) is established by Deep Learning. The deep learning model is composed of simple but non-linear modules which can be transformed to a higher level representation or slightly more abstract level representation (Lecun et al., 2015) from lower level representation. Thus, it is depicted as a multi-level representation learning method. Its structure makes it automatically to extract features from raw data. There are various deep learning methods such as deep belief network (DBN), deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN) etc. and out of them CNN is one of a classical deep learning model that has been applied in many fields, such as image recognition, object detection, pose tracking etc.

From past few years, CNN is frequently being used to recognize one dimensional (1D) signals. For the real-time classification of patient-specific electrocardiogram (ECG), Kiranyaz et al. (2016) applied 1D-CNN. 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 that occur on statistical quality Control charts to detect the unnatural deviation in state 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 simulation 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 recognitions. It is the process to generate thousands of random samples following a particular distribution using original data. In this project we have used raw eye tracking 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 tracking datasets and its obtained the mean and variances and generate data of various 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	$y_t = \mu + r(t) \times \delta + a \times sin(2/T)$	$r(t) \sim N(0, 1)$
2	Systematic, SYS	$y_t = \mu + r(t) \times \delta + d \times (-1)^t$	$\delta = 1\sigma$
3	Stratification, STR	$y_t = \mu + r(t) \times \delta'$	$\delta' \in (0.2\sigma, 0.4\sigma)$
4	Upward Trend, UT	$y_t = \mu + r(t) \times \delta + t \times g$	$d \in (1\sigma, 3\sigma)$
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	$y_t = \mu + r(t) \times \delta + k \times s$ $k = 1\ if\ t \geq P, else = 0$	$T = 16$ $g \in (0.005\sigma, 0.25\sigma)$
7	Downward Shift, DS	$yt = \mu + r(t) \times \delta - k \times s$ $k = 1\ if\ t \geq P, else = 0$	$P \in (10, 20)$ $s \in (1\sigma, 3\sigma)$ $t = 1, 2, 3, ...L$

- μ and σ = mean and standard deviation estimate of the in-control production process.
- $r(t)$ represents the inevitable accidental fluctuation which is subject to gaussian distribution $N(0, 1)$.
- d = Degree of system state departure.

- a =Amplitude of cyclic pattern.
- T = Period of the cycle.
- g = Gradient of a data trend.
- P = Time point when the shift anomaly occurs.
- s = Amplitude of the shift pattern.

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

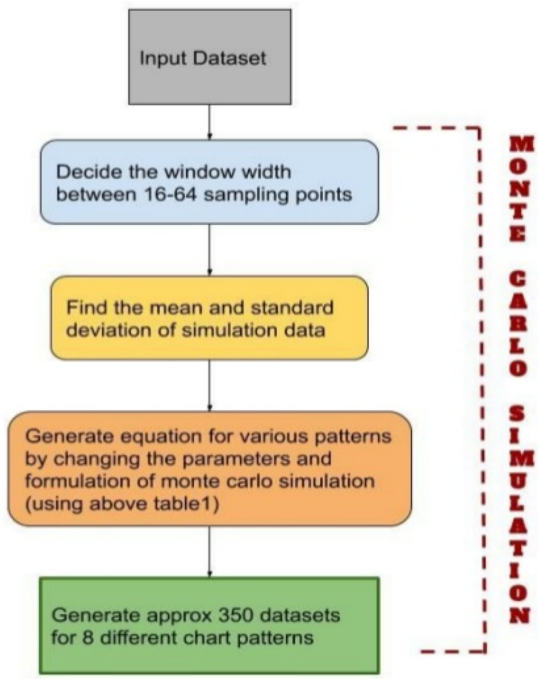


Figure1: Flowchart for Monte carlo simulation

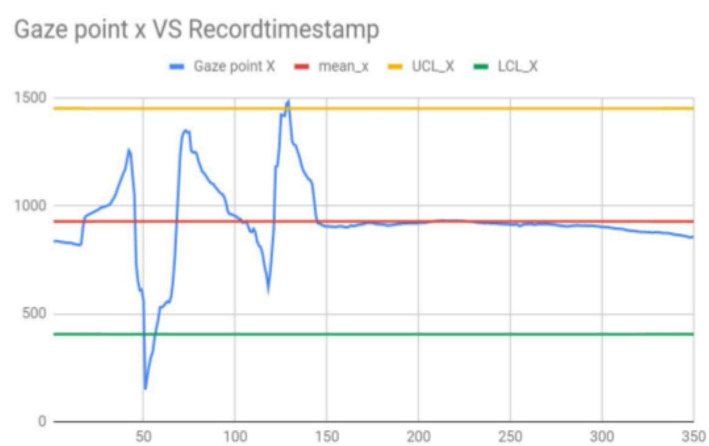
1. For the given sample of eye tracking gaze point datasets shown in the table2, we have done Data Simulation using monte carlo simulation on approx 10,944 raw eye gaze points data and prepared 8 different control chart pattern datasets with window length of 32 sampling points i.e; we make 342 datasets from 10944(342*32) data 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 sampling points.

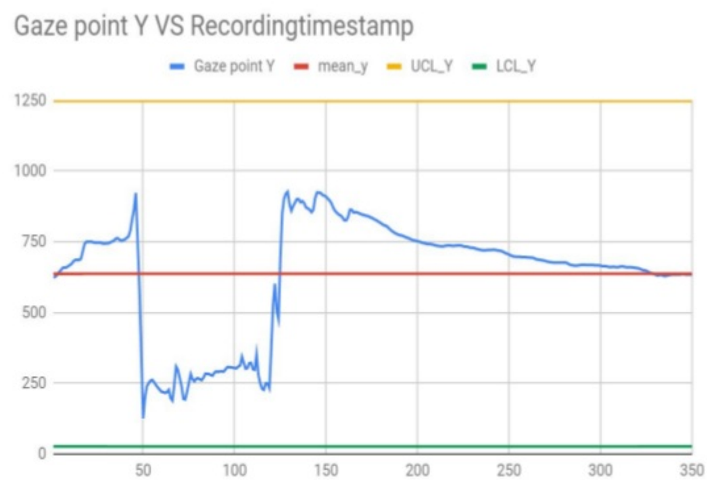
3. Similarly we have done for next dataset having 32 sample length, and like that from 10944 (342*32) eye tracking data we have prepared 342 pieces for each 8 control chart patterns having sample length of 32.
4. We have written code in python using pandas and numpy for monte carlo simulation.

Visualization of raw eye tracking datasets

I. Gaze point X VS Recording timestamp

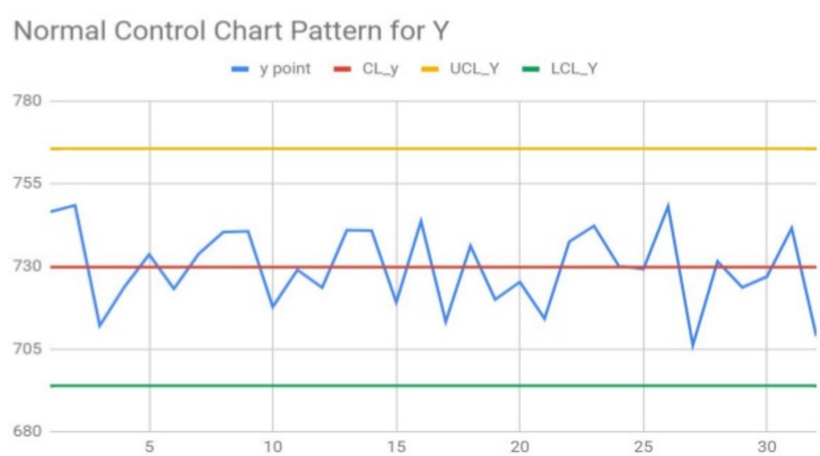


II. Gaze point Y VS Recording timestamp

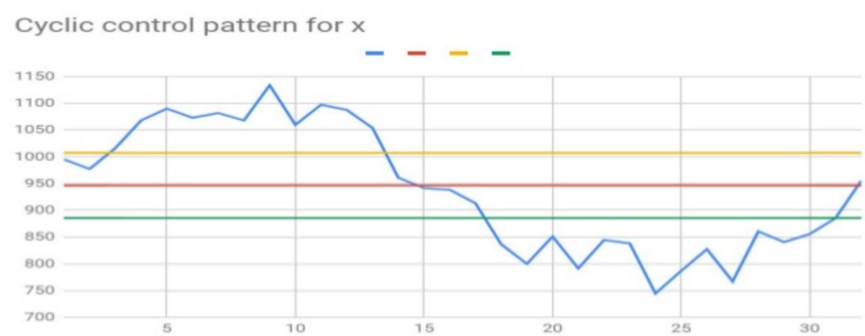


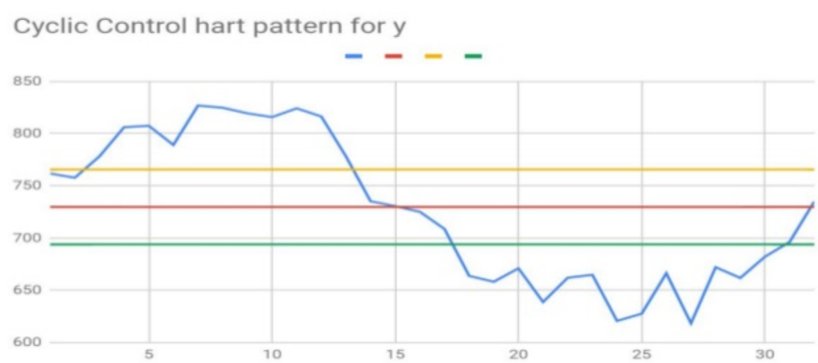
Visualization of Generated data of control chart patterns

I. Normal Patterns

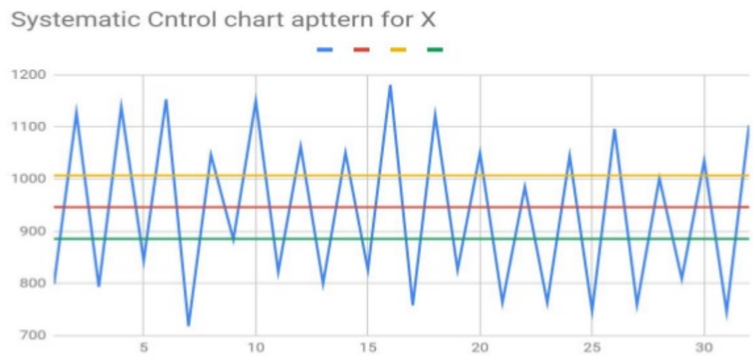


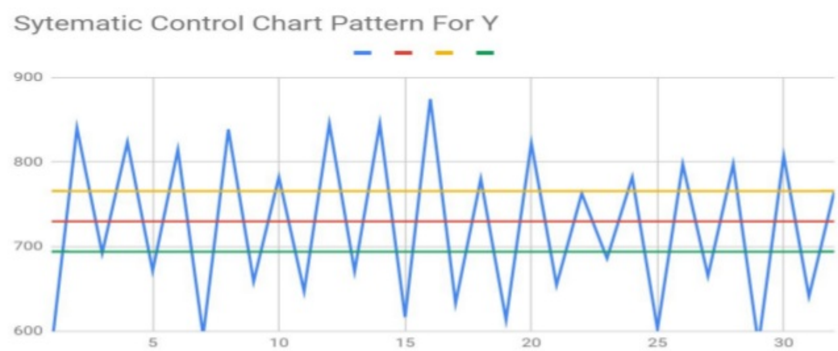
II. Cyclic patterns



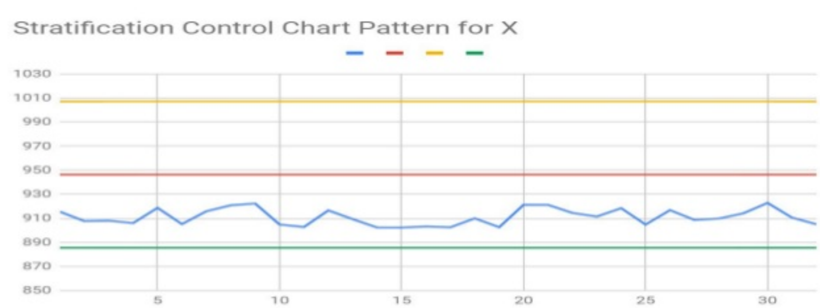


III. Systematic patterns

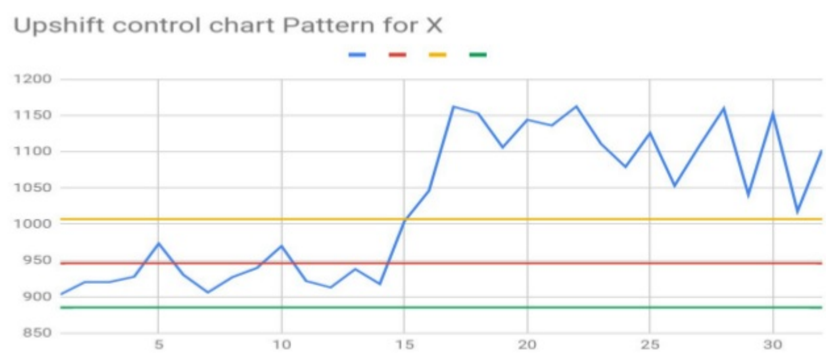


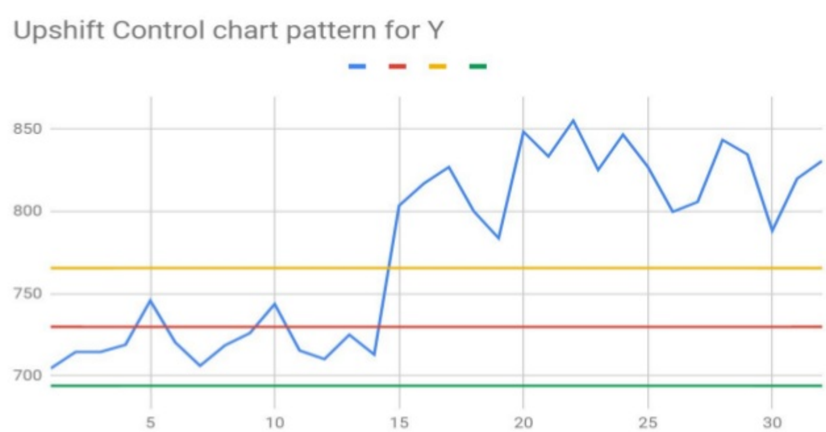


IV. Stratification patterns

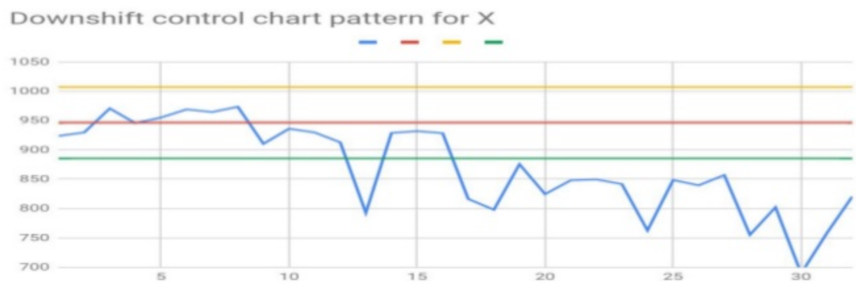


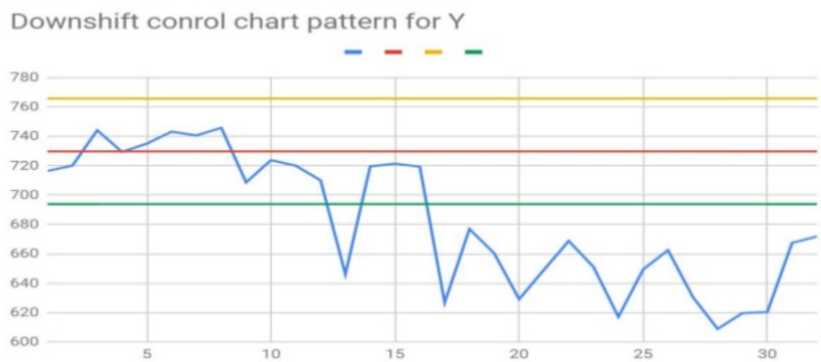
V. Upshift patterns



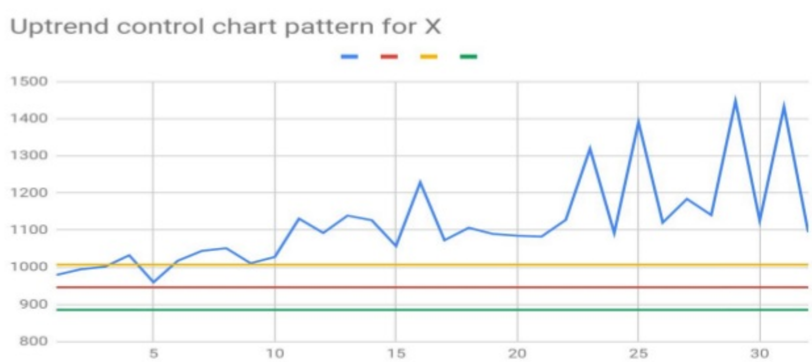


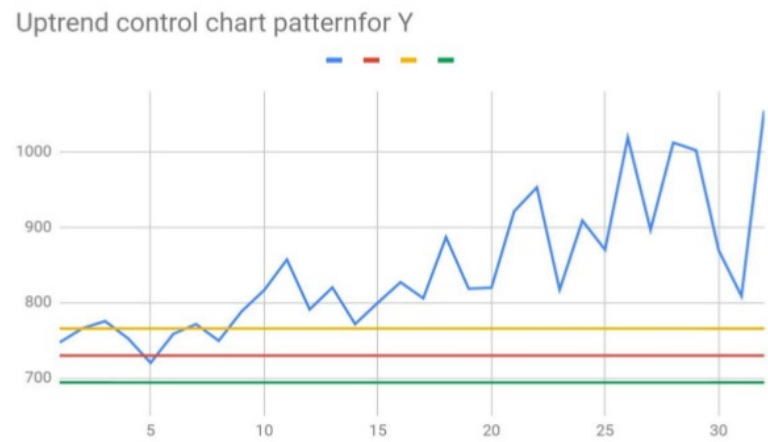
VI. Downshift patterns



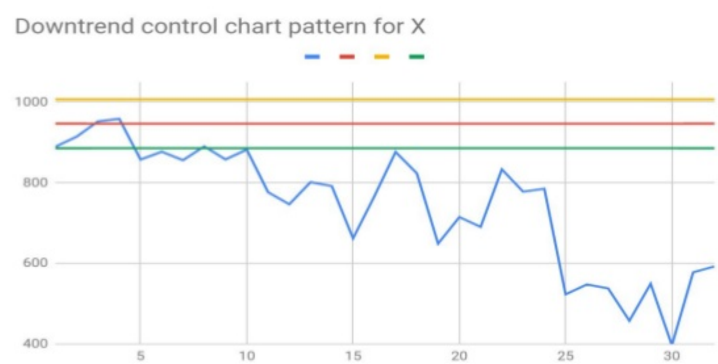


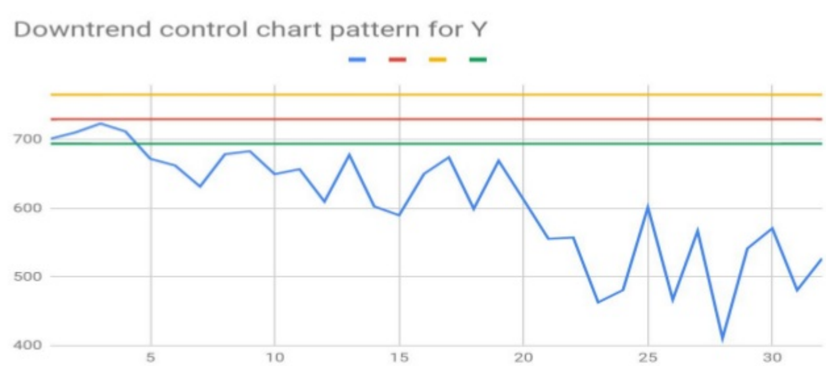
VII. Uptrend patterns





VIII. Downtrend patterns





2) Features extraction using Deep neural Networks

Feature extraction is the second step after data generation for different patterns using monte carlo simulation. We have used various neural networks and 1 dimensional convolutional neural networks to extract the feature from the generated data for different patterns.

How Convolutional neural networks work ?

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

- The convolutional layers: Extract features from the input data and generate feature 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 parameters of the neural network.

Forward Propagation

A. Forward Propagation: Convolutional layer

Every convolutional layer contains several filters to extract local features from the generated feature maps of the previous layer which is followed by an **activation function** 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.

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

To calculate each unit generated feature map the weight of filters used are the same, which is called weight sharing. The depth of feature maps is determined by the number of filters. The performance of CNN is affected by the size and number of filters per 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**-** Max pooling is used to reduce the image size by mapping all the elements of a given window into a single value by picking the maximum value from the elements in the window.

Average Pooling**-** It's the same as max-pooling except that it averages the value of all elements in the window instead of picking the maximum value.

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

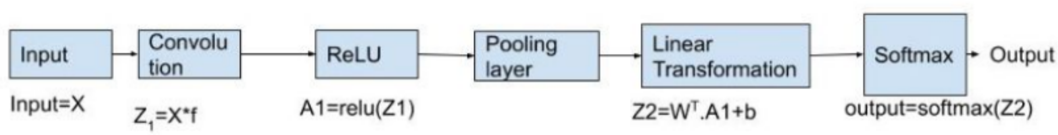


Figure2: Computation Graph for Forward Propagation

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

$$Z1 = X * f$$

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

$$A1 = ReLU(Z_1)$$

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.

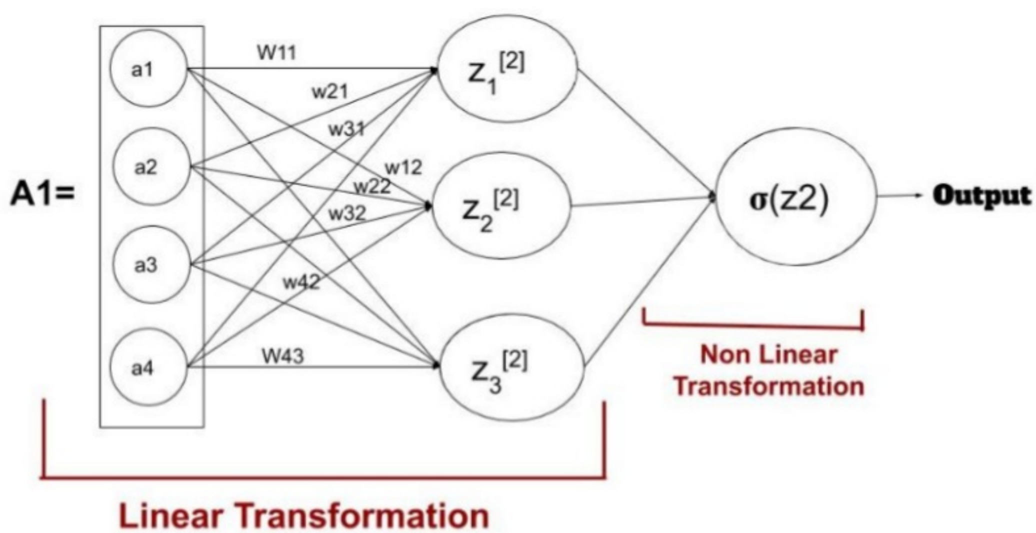


Figure3: Linear transformation and nonlinear transformation in fully connected

layer

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

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

The equation for linear transformation is:

$$Z2 = W^T . A1 + b$$

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 thus to capture those relationships we introduced activation function in the network which adds non-linearity to the data. We have used the Softmax activation function in the output layer.

Softmax function is the combination of multiple sigmoids. Sigmoid function returns value between 0 and 1, which are probabilities of a data point belonging to a particular class. Thus sigmoid is widely used binary classification problems. A softmax function is used for multiclass classification problems which returns the probability for a data point belonging to each individual class.

$$softmax = \sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1.....,K$$

Applying softmax activation function, This will be our final output

$$O = softmax(z2)$$

Backward Propagation

During the forward propagation process, we randomly initialized the weights, biases and filters. These values are treated as parameters from convolutional neural network algorithms.In the backward propagation process, the model tries to update these parameters such that the overall loss and recognition accuracy is more accurate.

For updating these parameters, we have used gradient descent techniques which find the value of parameters at which loss is minimum. The generic equation for updating the parameter values:

New parameter=old parameter-(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:

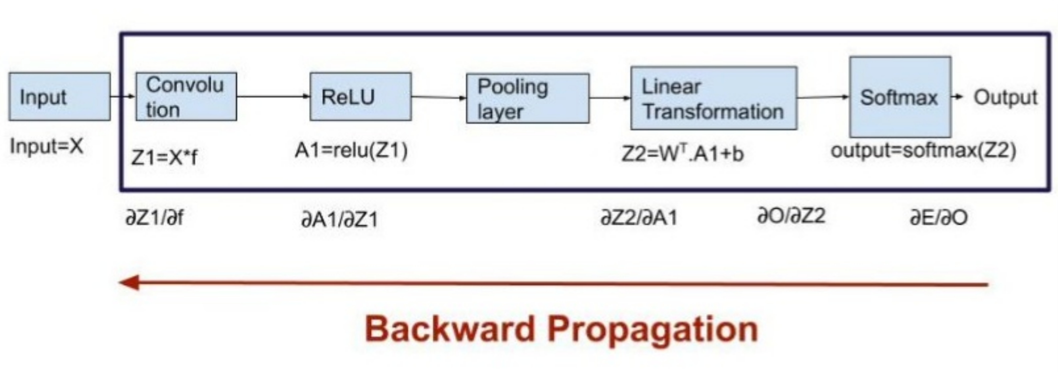


Figure4: Computation graph for Backward Propagation

Backward propagation: Fully Connected Layer

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

$$E/W = E/O \ . \ O/Z2 \ . \ Z2/W$$

Where

E/O = change in error with respect to output

$O/Z2$ = change in output with respect to Z2

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

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

$$W_{new} = W_{old} - lr * E/W$$

Similarly we will update bias using following equation:

$$b_{new} = b_{old} - lr * E/b$$

Backward Propagation: Convolution layer

The parameters for the convolution layer is a filter matrix which we had randomly initialized during the forward propagation process. Now we are going to update these values using 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.

$$E/f = E/O.O/Z2.Z2/A1\ .A1/Z1.Z1/f$$

Where

$Z_2/A1$ = change in Z_2 with respect to A1

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

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

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

$$f_{new} = f_{old} - lr * (E/f)$$

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.

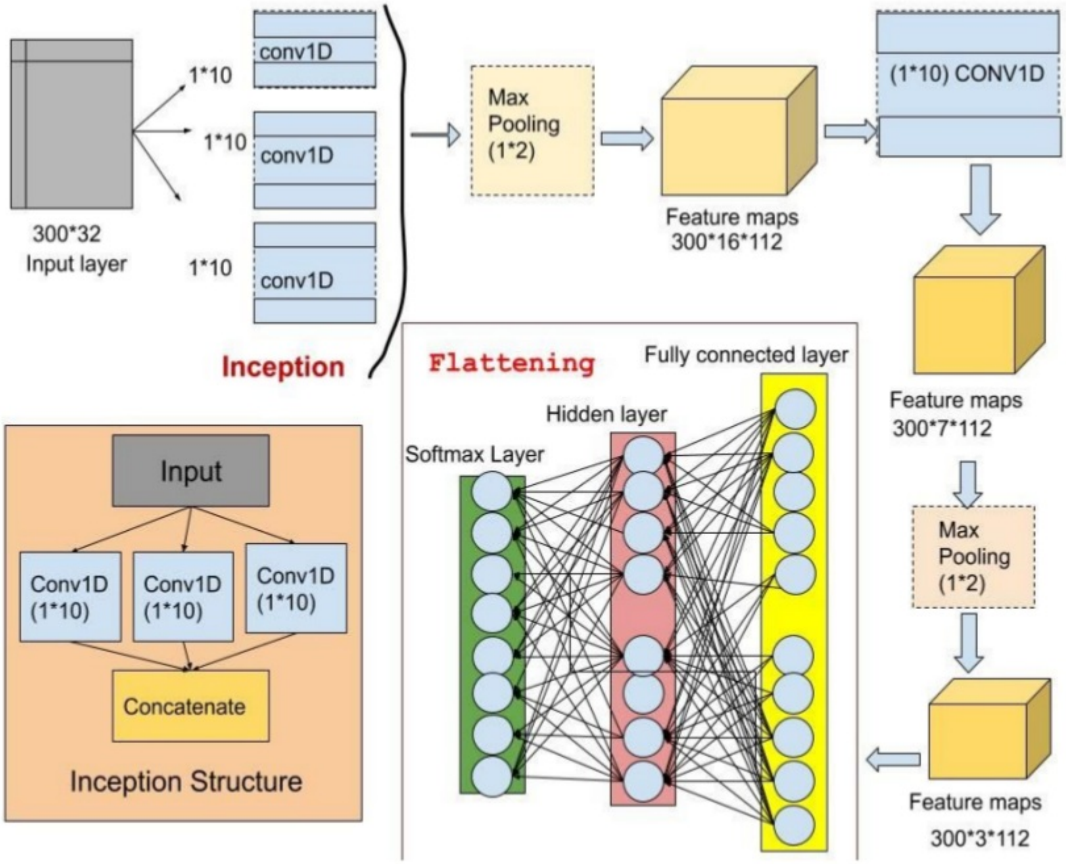
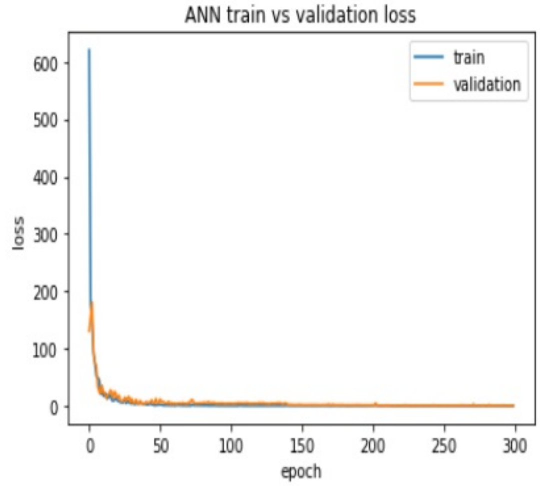


Figure5: Architecture of Improved one dimensional convolutional neural network

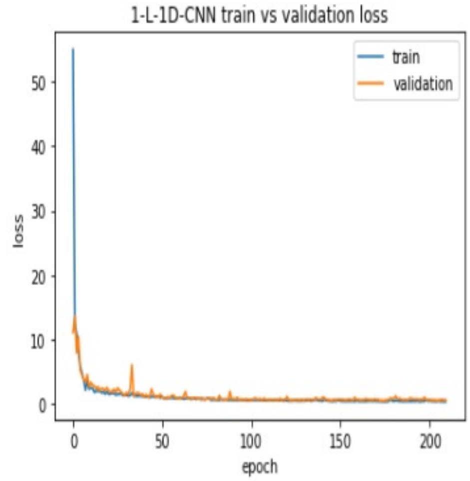
5. Results

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

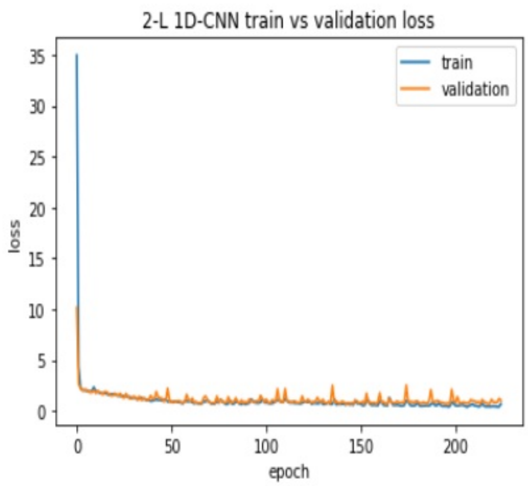
1) **Graph between training and Validation loss of models per epoch**
   The following graph shows how the loss is decreasing with the increase in the number of epochs.



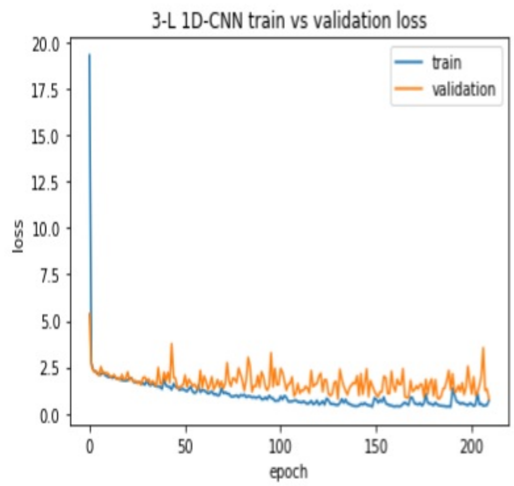
(I) ANN



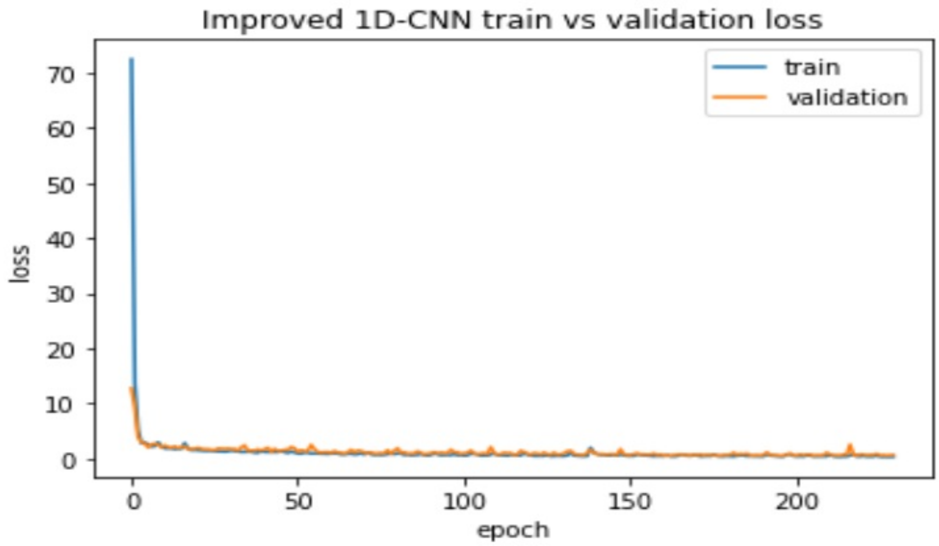
(II) 1layer 1D



(III) 2 layer 1D CNN

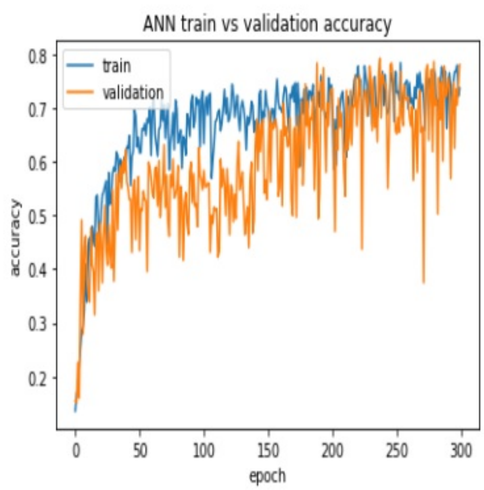


(IV) 3 layer 1D CNN

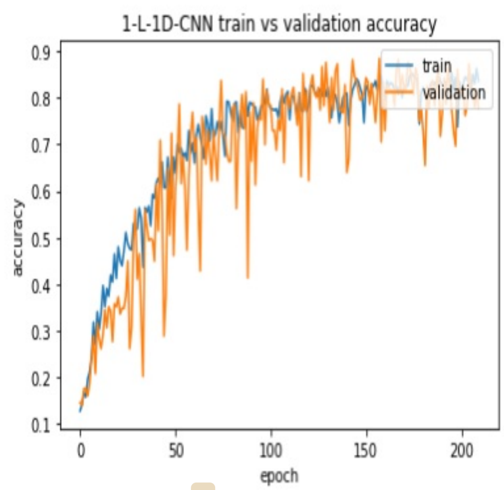


(V) Improved 1D CNN

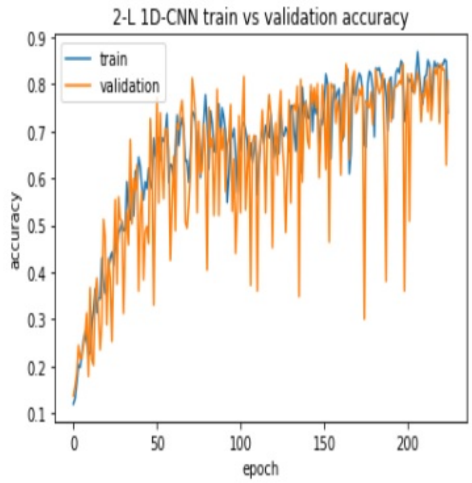
2) **Graph between training and validation accuracy of models per epoch**
The following graphs show how the accuracy of the model increases per epoch during the training process.



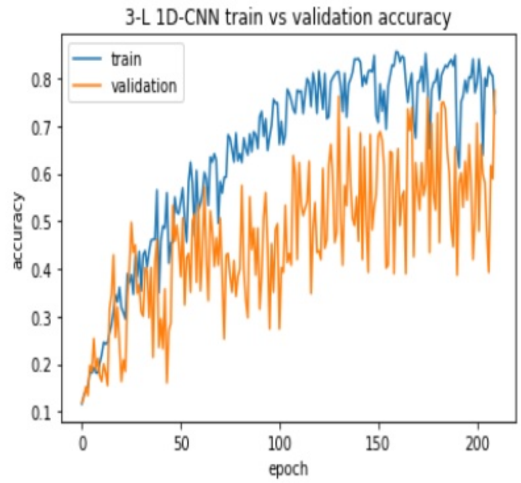
(I) ANN



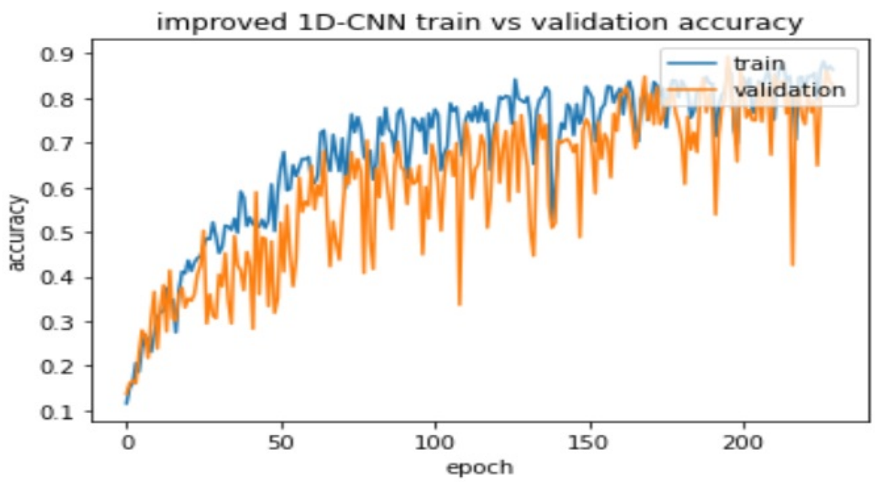
(II)1 layer 1D CNN



(III) 2 layer 1D CNN



(IV) 3 layer 1D CNN

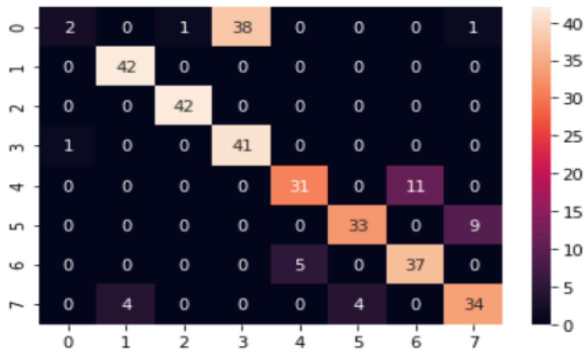


(V) Improved 1D CNN

3) Confusion matrix

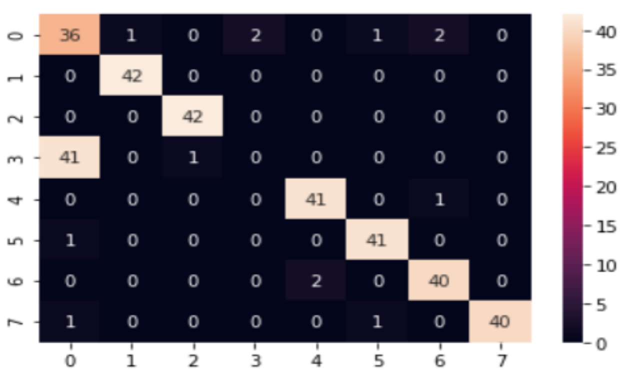
A confusion matrix is used to check the performance of a classification model on a set of test data.Calculating a confusion matrix can give you an idea of where the model is right and what types of errors it is making.

1) ANN



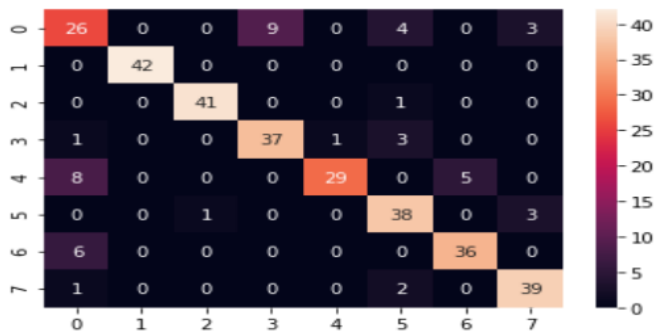
The confusion matrix heatmap for ANN shown in the above figure gives the recognition accuracy of 77.38% which is lowest among all models and misclassification of 22.62% which is highest among all models.

2) 1 layer 1D CNN



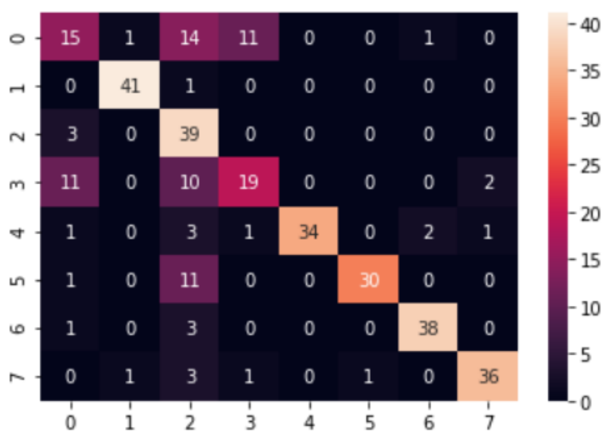
The confusion matrix heatmap for 1 layer 1D CNN shown in the above figure gives the accuracy of 83.9% and misclassification of 16.9%.

3) 2 layer 1D CNN



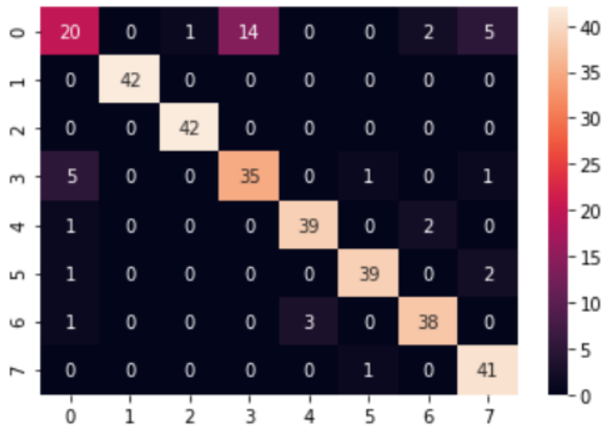
The confusion matrix heatmap shown in the above figure gives the recognition accuracy of 85.7% and has misclassification o0f 14.3%.

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 heatmap shown in the above figure gives the recognition accuracy of 88.09% which is highest 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 data	0.8062	0.84	0.889	0.835	0.90625
Accuracy on test data	0.786	0.839	0.857	0.7738	0.881

5) **Loss**

Model	ANN	1 layer 1D CNN	2 layer 1D CNN	3 layer 1D CNN	Improved 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 are 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 which allows the internal layer to pick and choose which filter size will be relevant to learn the required information. All these feature-based extraction methods help to develop automated recognition systems for control chart patterns recognition and thus when we fed the datasets, the model helps to identify or detect abnormal patterns, deviations and as well as abnormal abruption in the production process. Abnormal patterns exhibited in control charts are associated with various assignable causes adversely affecting the process stability 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 gives more consistent recognition performance and dominant over layer by layer neural network as well as some classical methods like fuzzy inference systems as well as 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 pattern and stratification 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 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 reduce the complexity of the production process and help 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

ccps using cnn

ORIGINALITY REPORT

38%	%	%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Jie Xu, Huichun Lv, Zilong Zhuang, Zhiyao Lu, Dewei Zou, Wei Qin. "Control Chart Pattern Recognition Method Based on Improved One-dimensional Convolutional Neural Network", IFAC-PapersOnLine, 2019 Publication	18%
2	www.analyticsvidhya.com Internet Source	9%
3	Gauri, S.K.. "Feature-based recognition of control chart patterns", Computers & Industrial Engineering, 200612 Publication	3%
4	Submitted to University of Lancaster Student Paper	1%
5	Susanta Kumar Gauri, Shankar Chakraborty. "Improved recognition of control chart patterns using artificial neural networks", The International Journal of Advanced Manufacturing Technology, 2007 Publication	1%

6	Meliboev Azizjon, Alikhanov Jumabek, Wooseong Kim. "1D CNN based network intrusion detection with normalization on imbalanced data", 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2020 Publication	1%
7	Roberta Avanzato, Francesco Beritelli. "Automatic ECG Diagnosis Using Convolutional Neural Network", Electronics, 2020 Publication	1%
8	Ramazan Ünlü. "chapter 13 An Assessment of Imbalanced Control Chart Pattern Recognition by Artificial Neural Networks", IGI Global, 2020 Publication	<1%
9	A. Ebrahimzadeh. "Recognition of control chart patterns using genetic algorithm and support vector machine", 2009 First International Conference on Networked Digital Technologies, 07/2009 Publication	<1%
10	pdfs.semanticscholar.org Internet Source	<1%
11	Tao Zan, Zhihao Liu, Hui Wang, Min Wang, Xiangsheng Gao. "Control chart pattern recognition using the convolutional neural network", Journal of Intelligent Manufacturing,	<1%

2019
Publication

12	Submitted to Universiti Kebangsaan Malaysia Student Paper	<1%
13	Monark Bag, Susanta Kumar Gauri, Shankar Chakraborty. "An expert system for control chart pattern recognition", The International Journal of Advanced Manufacturing Technology, 2011 Publication	<1%
14	R.-S. GUH, J. D. T. TANNOCK. "Recognition of control chart concurrent patterns using a neural network approach", International Journal of Production Research, 5/20/1999 Publication	<1%
15	"Availability and Maintainability in Engineering Design", Handbook of Reliability Availability Maintainability and Safety in Engineering Design, 2009 Publication	<1%
16	www.jm-warmte.nl Internet Source	<1%
17	Chih-Hsuan Wang. "A hybrid approach for identification of concurrent control chart patterns", Journal of Intelligent Manufacturing, 05/29/2008 Publication	<1%

18	M.A. Wani, S. Rashid. "Parallel Algorithm for Control Chart Pattern Recognition", Fourth International Conference on Machine Learning and Applications (ICMLA'05), 2005 Publication		<1%
19	"Proceedings of the Third International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’18)", Springer Science and Business Media LLC, 2019 Publication		<1%
20	Ataollah Ebrahimzadeh. "Recognition of control chart patterns using swarm intelligence and neural networks based on the statistical and shape features", IEEJ Transactions on Electrical and Electronic Engineering, 01/2011 Publication		<1%
21	www.iieom.org Internet Source		<1%
22	D. T. PHAM and M. A. WANI. "Feature-based control chart pattern recognition", International Journal of Production Research, 7/1/1997 Publication		<1%
23	Zhou, Xueliang, Pingyu Jiang, and Xianxiang Wang. "Recognition of control chart patterns using fuzzy SVM with a hybrid kernel function", Journal of Intelligent Manufacturing, 2015. Publication		<1%

24	open.library.ubc.ca Internet Source	<1 %
25	Sepp Hochreiter, Jürgen Schmidhuber. "Long Short-Term Memory", Neural Computation, 1997 Publication	<1 %
26	SusantaKumar Gauri, Shankar Chakraborty. "Feature-Based Recognition of Control Chart Patterns: A Generalized Approach", Quality Technology & Quantitative Management, 2016 Publication	<1 %
27	"Artificial Intelligence Applications and Innovations", Springer Science and Business Media LLC, 2020 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		