International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Prediction of Personality Traits from Text using Time Efficient Preprocessing and Deep Convolution Neural Network

Gurpreet Singh Chhabra, Anurag Sharma

Abstract*: Persons express their sentiments as part of day-by-day communiqué. Personality traits reveals persons thoughts, about their feelings and persons behavior. Hence personality traits sets psychology how one person different from one another. Person thoughts expressed by what he/she write about any situation. Henceforth personality traits prediction is the vital research area. This research area belongs to NLP (Natural Language Processing), the most widely accepted of these traits are as: Openness (OPE), Conscientiousness (CON), Extraversion (EXT), Agreeableness (AGR), and Neuroticism (NEU). In this paper we have proposed to use time efficient sentence tokenization algorithm, efficient text preprocessing prominence on emoji's followed by CNN deep learning classifier , proposed prediction model uses convolution filter for feature selection, further we have compared prediction model with machine learning based prediction model. We have also compared brute force tokenization method with proposed tokenization algorithm over dataset different in size.

***Keywords* : NLP, CNN, SVM, KNN..**

I. INTRODUCTION

Written text contains huge information unlike sentiment inside text i.e. through written text we can predict opinion of person by whom it is written and one more promising area is prediction of personality from written text, text processing is the NLP (Natural Language Processing) technique.

NLP takes input as set of Unicode characters usually UTF-8, then basic processing need is convert set of characters into lexical charters (phrases, words and syntactic markers).

Before going to deep dive into text classification we need to understand machine learning or deep learning workflow fig.-1 depicts the same.

In this paper we have proposed the use of finite automata for string tokenization which is time efficient then brute force algorithm, further we will user Convolution neural network as classification techniques. In section II we have discuss some literature, III how data prepared for classifier, IV methodology, V experiential evaluation, at last we will conclude our research.

Manuscript published on 30 September 2019
* Correspondence Author
Mr. Gurpreet Singh Chhabra*, CSE, MATS University, Raipur C.G., India. Email: Gurpreet.kit@gmail.com
Dr. Anurag Sahrma, CSE, MATS University, Raipur C.G., India. Email: anusiraag@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

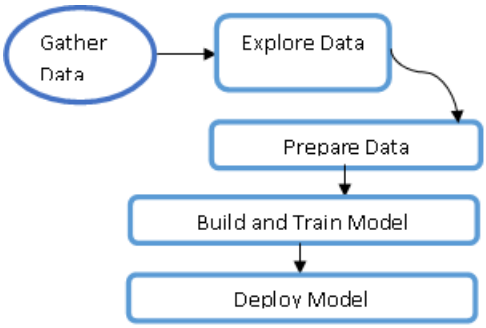


Fig.-1 Work flow for Prediction Model

II. LITERATURE SURVEY

Said A. Salloum et. al. did research over text mining, survey mainly emphasis on how information extracted from research articles[1] using data mining techniques, as we know text mining is the data mining technique. Paper concluded that for text analysis word cloud and word frequency technique used. Paper at last came into conclusion that association rule mining, clustering, word cloud and word frequency mainly used, applying association rule mining followed by word frequency provide better result for text analysis.

Xuelian Deng et. al. reviewed over different feature selection techniques for text analysis, author also reviewed that most popular text classification techniques like KNN, SVM, Naïve Bayes, Decision Tree and Neural Network [2]. Author concluded following tabular Information as below:

Table:1 Specification [2]

Text Distance Technique	Type
Euclidean distance	Distance based
Hamming distance	Distance based
Cosine similarity	Vector space model based
Dice coefficient	Vector space model based
Jaccard coefficient	Vector space model based
Pearson correlation	Correlation based
SMTP	Information theory based
IT-Sim	Information theory based

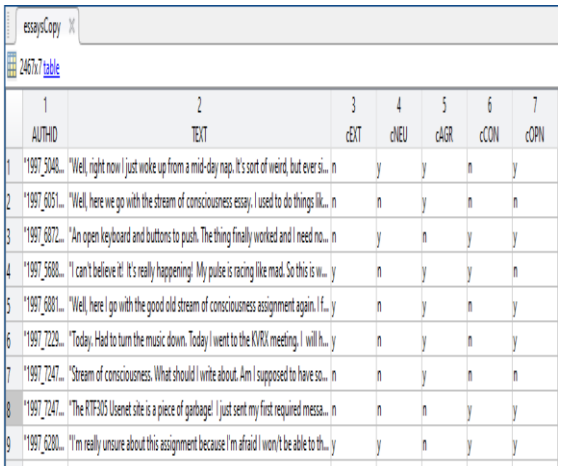
EMD-based	distribution based
K-L divergence-based	distribution based

Tomas Mikolov et. al.[3] discussed continues beg of word model (CBOW), skip gram model. Author concluded that Word vector provide state of the art performance. Word vector will over larger dimensionality dataset. Word vector became vital building block for NLP research.

Navonil Majumder et. al. said that personality is a combination of an individual's behavior, emotion, motivation, and thought pattern characteristics. Our identity has an awesome effect on our lives; it influences our life decisions, prosperity, wellbeing, and various different inclinations. Automatic recognition of a person's personality traits has numerous essential practical applications. With regards to estimation examination, for instance, the items and administrations prescribed to a man ought to be those that have been decidedly assessed by different clients with a comparative identity compose. Author given future direction of research as incorporate more features and preprocessing [IEEE 2017].

III. DATASET USED AND PREPRATION

In this research we have used "essays" dataset, dataset downloaded from the following URL. Url:-https://github.com/SenticNet/personality-detection, below shows the snippet on the dataset used, here dataset contains seven features as:



essaysC.opy

2467x7 table

	1 AUTHID	2 TEXT	3 cEXT	4 cNEU	5 cAGR	6 cCON	7 cOPN
1	"1997_5048...	"Well, right now I just woke up from a mid-day nap. It's sort of weird, but ever si...	n	y	y	n	y
2	"1997_6051...	"Well, here we go with the stream of consciousness essay. I used to do things lik...	n	n	y	n	n
3	"1997_6872...	"An open keyboard and buttons to push. The thing finally worked and I need no...	n	y	n	y	y
4	"1997_5688...	"I can't believe it! It's really happening! My pulse is racing like mad. So this is w...	y	n	y	y	n
5	"1997_6881...	"Well, here I go with the good old stream of consciousness assignment again. I f...	y	n	y	n	y
6	"1997_7228...	"Today. Had to turn the music down. Today I went to the KVRX meeting. I will h...	y	n	y	n	y
7	"1997_7247...	"Stream of consciousness. What should I write about. Am I supposed to have so...	n	n	y	n	n
8	"1997_7247...	"The RTF305 Usenet site is a piece of garbage! I just sent my first required messa...	n	n	n	y	y
9	"1997_6280...	"I'm really unsure about this assignment because I'm afraid I won't be able to th...	y	y	n	y	y

Fig.-2 Snippet of essays dataset

- Document ID
- Written Text
- cEXT: Extraversion Personality trait
- cNEU: Neuroticism Personality trait
- cAGR: Agreeableness Personality trait
- cCON: Conscientiousness Personality trait
- cOPN: Openness Personality trait

Further we have to prepare dataset for deep learning classfier we have converted data to difeern class as, if any document having cEXT=y then we have to put "1", algorithm for the same as follows:

Algorithm: Prepare Dataset

```
1. Input dataset.
2. Specify range and delimiter.
3. col=3 : 7
4. row=1 : n // n is depend on End of file
5. class=''
6. while row=1 to n
   if col(3)='y'
      class=concatenate(class,1)
         if col(3)='n'
           class=concatenate(class,0)
         if col(4)='y'
      class=concatenate(class,1)
         if col(4)='n'
           class=concatenate(class,0)
         if col(5)='y'
      class=concatenate(class,1)
         if col(5)='n'
           class=concatenate(class,0)
         if col(6)='y'
      class=concatenate(class,1)
         if col(6)='n'
           class=concatenate(class,0)
         if col(7)='y'
      class=concatenate(class,1)
         if col(7)='n'
           class=concatenate(class,0)
// Missing Class row removed
if class=''
   remove row from dataset
```

Illustration (example)

#AUTHID	TEXT	cEXT	cNEU	cAGR	cCON	cOPN
1997_504851.txt	text1	n	y	Y	n	y
		0	**1**	**1**	**0**	**1**
1997_605191.txt	text2	n	n	y	n	n
1997_687252.txt	text3	n	y	n	y	y
		0	**1**	**0**	**1**	**1**
1997_568848.txt						
	Missing Class					

2467x3 table

	1 AUTHID	2 TEXT	3 class
1	"1997_5048...	"Well, right now I just woke up from a mid-day nap. It's sort of weird, b...	1101
2	"1997_6051...	"Well, here we go with the stream of consciousness essay. I used to do ...	100
3	"1997_6872...	"An open keyboard and buttons to push. The thing finally worked and ...	1011
4	"1997_5688...	"I can't believe it! It's really happening! My pulse is racing like mad. So...	10110
5	"1997_6881...	"Well, here I go with the good old stream of consciousness assignment...	10101
6	"1997_7229...	"Today. Had to turn the music down. Today I went to the KVRX meetin...	10101

Fig.-3 Prepared Classful dataset

IV. RESEARCH GAPE AND METHDOLOGY

Research Gap

- Tokenization of sentence takes too much time i.e. time complexity is high.
- Much existing algorithm does not much emphasis on emoji's.
- Existing machine learning classifiers having lesser accuracy.
- Existing deep learning classifier having lesser accuracy due to less efficient preprocessing over input dataset.
- Missing class may affect the prediction accuracy.

Methodology

Prediction of personality traits is from written text is nothing but text classification, primary step for any text classification require sentence tokenization. Most of earlier algorithm uses brute force method for string tokenization having time complexity O(nm) where n is length of string and m is length of pattern (Word Separator), here we have proposed to use finite automata string searching algorithm which has O(n) as worst case time complexity which is lesser then brute force tokenization.

Preprocessing

FA String Matcher

1. n ← length [T]
2. q ← 0
3. for i ← 1 to n
4. do q ← δ (q, T[i])
5. If q =m
6. then s←i-m
7. print "Pattern occurs with shift s" s

Compute Transition Function

1. m ← length [P]
2. for q ← 0 to m
3. do for each character a ∈ ∑*
4. do k ← min (m+1, q+2)
5. repeat k←k-1
6. Until
7. δ(q,a)←k
8. Return δ

Fig. 4 shows the finite automata which will search space [\s] in input sentence

Q = {q1, q2, q3}

∑= {a-z, A-Z, 0-9, \s}

F= {q3}

Different person's uses emoji's during conversation or whenever they want to express something henceforth emoji's also plays important role to predict the personality of person. As per the data of *Swiftkey* emoji's report person uses different emoji's as their mood. Fig.-5 shows the report.

Further we will tokenize emoji's in input document sentence and assign weight as per following algorithm.

Algorithm

1. Get all the updated Emoticons.
2. Assign weight to each emoticon as per likelihood of emoticons.
3. Create feature vector or weight matrix of emotions.
4. Apply thresholding to feature vector.
5. Take threshold weight and convert weight matrix to smile positive as 1 and smile negative as 0.
6. Update feature

Features: For features of sentence we are using neural network way, as we know beg of words representation uses sparse which is sum of sparse one hot encoded vectors corresponding to each particular word, in neural network usually use dense representation in which each word replaced by dense vector which is much shorter, it can have ~300 values. Example of this vector is word2vector embedding. Further we have to use convolution filter in which we need to know about 1-D convolution called n-gram vector.

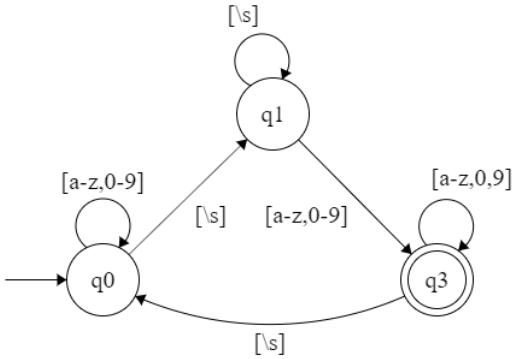


Fig.-4 Finite Automata for [\s] searching

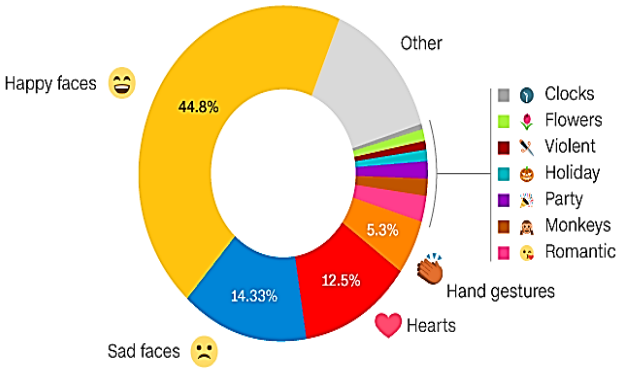


Fig.-5 Swiftkey emoji's Report [CNN 2015]

1-Gram (Example)

Word		Word Embedding		
Cat	→	0.7	0.4	0.5
Sitting		0.2	-0.1	0.1
There		-0.5	0.4	0.1

2- Gram (Example)

Word		Word Embedding		
Cat	→	0.7	0.4	0.5
Sitting		0.2	-0.1	0.1
There		-0.5	0.4	0.1

Convolution filter				
0.6	0.4	0.5	→	0.9
0.2	-0.1	0.2		

The convolution provide high activation for 2-gram. In our proposed prediction model we have taken n-gram length as [2,3,4,5]. Further for each of the n-gram lengths, we have created a block of convolution, batch normalization, dropout, and max pooling and ReLU layers and connecting every block to input layer. Proposed prediction model can be depicted as fig.-5, involved validation of prediction model followed by 8 steps.

V. EVALUATION

For evaluation of proposed prediction model we have used Matlab 2019a tool, dataset used in our research is "essays", as deatiled in secion 3. In this section we have depicted each step output mentioned in fig.-5.

Table-3 Comparison of FA & Brute force Tokenization

S. No.	Dataset Size	Brute Force Execution Time in ms	FA Execution Time in ms
1	8461 KB	7.0	1.67
2	7003 KB	5.7	1.34
3	3122 KB	4.12	0.65

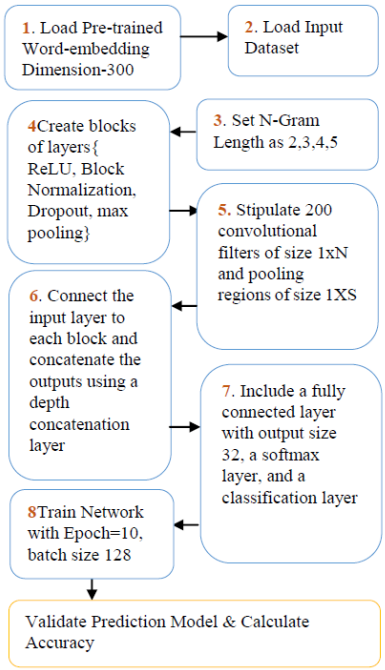


Fig.-5 Proposed Flow

Fig. 6 Comparison based on table-3 data

As we saw from fig.-6 proposed finite automata based string tokenization works time efficient over brute force string pattern matching based tokenization.

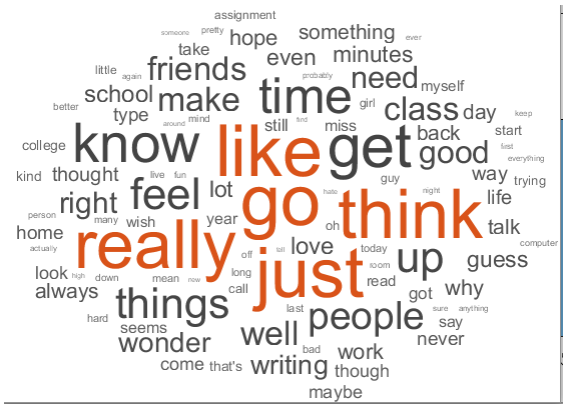


Fig.-7 Input data Visuvalization(Step-1)

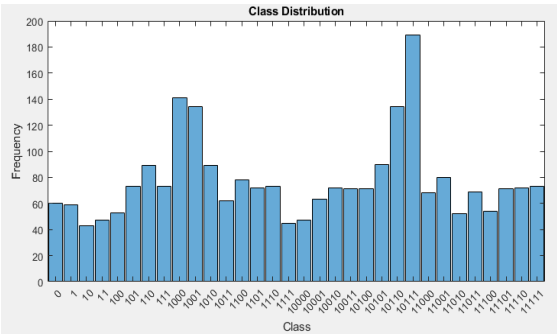


Fig.-8 32 Classes of Response Variable with their Frequency

Emojis

Fig.-9 Emoji's Seperation from document

For every n-gram lengths, created block of convolution, batch normalization, ReLU, dropout, and max pooling layers shown in fig.-10.

Adding the depth concatenation layer, the fully connected layer, the softmax layer, and the classification layer shown in fig.-11.

Connect the max pooling layers to the depth concatenation layer and the final network architecture, depicted in figure 12.

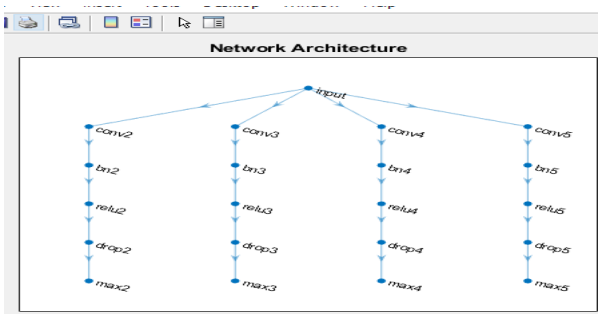


Fig.-10 After Step-4 Network Architecture

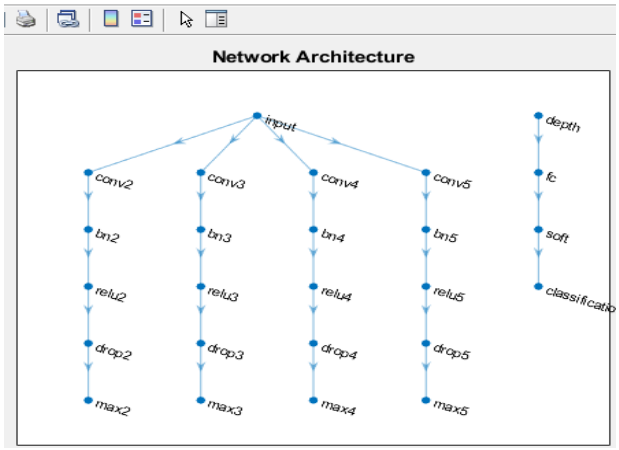


Fig.-11 After Step-5 Network Architecture

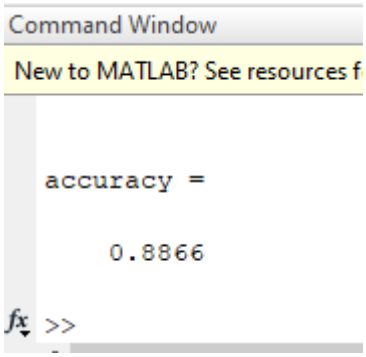


Fig. 14 Accuracy Achieved

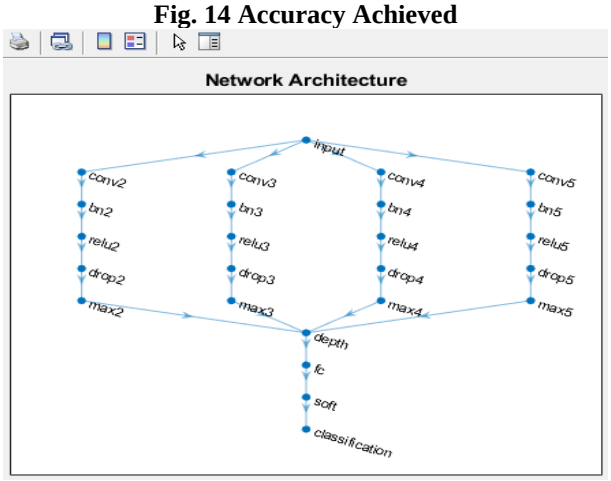


Fig.-12 After Step-6 Network Architecture

Finally proposed prediction model training process as number of epochs=10 and achieved 88.38% accuracy as highlighted in fig.13.

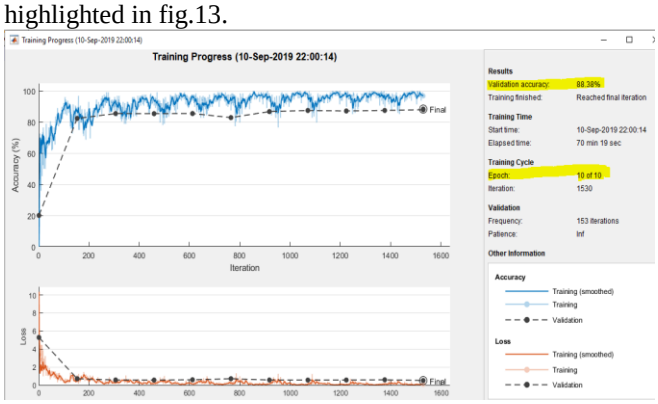


Fig.-13 after Step-7, 8 Training Process

VI. CONCLUSION

In this paper we have proposed the use of CNN based deep learning model for personality traits prediction, importantly in this paper we have compared brute force and Finite Automata tokenization, observed that (Table-3) FA based tokenization performs well, further concentrated on emoji's as earlier prediction does not make musk emphasis on emoji's, at last we have trained over CNN network with 10 epochs with batch size 128 and achieved 88.6 accuracy

REFERENCES

1. Said A. Salloum, Mostafa Al-Emran, Azza Abdel Monem and Khaled Shaalan , Using Text Mining Techniques for Extracting Information from Research Articles, Springer International Publishing AG 2018 K. Shaalan et al. (eds.), Intelligent Natural Language Processing: Trends and Applications, Studies in Computational Intelligence 740, https://doi.org/10.1007/978-3-319-67056-0_18.
2. Xuelian Deng, Yuqing Li1, Jian Weng, Jilian Zhang Feature selection for text classification: A review, Received: 12 March 2018 / Revised: 14 April 2018 / Accepted: 30 April 2018 © Springer Science+Business Media, LLC, part of Springer Nature 2018.
3. Tomas Mikolov, Kai Chen Efficient Estimation of Word Representations in Vector Space, arXiv:1301.3781v3 cs.CL. 7 Sep 2013.
4. Navonil Majumder et. al. Deep Learning-Based Document Modeling for Personality Detection from Text IEEE 2017.
5. Erik cambria et.al. Affective Computing and Sentiment Analysis IEEE 2016
6. Giulio Angiani et. al. A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter Springer 2016
7. I.Hemalatha et. al. Preprocessing the Informal Text for efficient Sentiment Analysis IJETTCS 2012
8. Ahmed Hassan et. al. Sentiment analysis algorithms and applications: A survey Elsevier 2014
9. Daoud Clarke et. al. Developing Robust Models for Favourability Analysis ACLWEB 2012
10. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passoneau, R.: Sentiment Analysis of Twitter Data. Computer Science - Columbia University (New York, USA) (2011).
11. Balahur, A.: Sentiment Analysis in Social Media Texts. European Commission Joint Research Center (Varese, Italy) (2013).
12. Bao, Y., Quan, C., Wang, L., Ren, F.: The role of pre-processing in twitter sentiment analysis. In: International Conference on Intelligent Computing. pp. 615{624. Springer (2014).
13. Cambria, E., Olsher, D., Rajagopal, D.: Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In: Twenty-eighth AAAI conference on artificial intelligence (2014) 5. Duncan.
14. B., Zhang, Y.: Neural networks for sentiment analysis on twitter. In: Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on. pp. 275{278. IEEE (2015)

AUTHORS PROFILE

Mr Gurpreet Singh Chhabra having Master degree in Computer Science & Engineering and he is pursuing PhD. He has a life member of ISTE (Indian Society for Technical Education) and a member of CSI (Computer Society of India) and also has a credit of many national and international papers. His qualifications are fortified with a great deal of creativity and problem-solving skills. He has also a proven record of teaching for the last 15 years.

Dr. Anurag Sharma, currently working as Professor in the department of Computer Science & Engineering at MATS School of Engineering & IT, MATS University, Raipur, having Ph.D. in Computer Science & Engineering. He has a life member of Quality Circle Forum of India and a member of IEEE and also has a credit of many national and international papers. His qualifications are fortified with a great deal of creativity and problem-solving skills. He has also a proven record of teaching for the last 15 years. He also has a many Patents Filed and received an Award in many different filled.