Tribhuvan University
Institute of Science and Technology

Comparative Study of Back - Propagation and Monte - Carlo Artificial Neural Network for Plagiarism Detection in Nepali Language

Dissertation
Submitted to

Central Department of Computer Science and Information Technology
Tribhuvan University, Kirtipur
Kathmandu, Nepal

In partial fulfillment of the requirements
for the Master's Degree in Computer Science and Information Technology

By
Rakesh Kumar Bachchan

September 7, 2017

Tribhuvan University
Institute of Science and Technology

Comparative Study of Back - Propagation and Monte - Carlo Artificial Neural Network for Plagiarism Detection in Nepali Language

Dissertation
Submitted to

Central Department of Computer Science and Information Technology
Tribhuvan University, Kirtipur
Kathmandu, Nepal

In partial fulfilment of the requirements
for the Master's Degree in Computer Science and Information Technology

By
Rakesh Kumar Bachchan
September 7, 2017

Supervisor
Arun K. Timalsina, Ph.D

Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science and Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work. No part of this thesis is reproducible without the written authority of the author.

.....
Rakesh Kumar Bachchan
Date: September 7, 2017

Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science and Information Technology

Supervisor's Recommendation

I hereby recommend that this dissertation prepared under my supervision by **Mr. Rakesh Kumar Bachchan** entitled **"Comparative Study of Back - Propagation and Monte - Carlo Artificial Neural Network for Plagiarism Detection in Nepali Language"** in partial fulfilment of the requirements for the degree of M.Sc. in Computer Science and Information Technology be processed for the evaluation.

.....
Arun K. Timalsina, Ph.D
Assistant Professor
Department of Electronics and Computer Engineering
Pulchowk Campus, Institute of Engineering
Tribhuvan University, Nepal
Date: September 7, 2017

Tribhuvan University
Institute of Science and Technology
Central Department of Computer Science and Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master's Degree in Computer Science and Information Technology.

Evaluation Committee

Mr. Nawaraj Poudel
Head of Department
Central Department of Computer Science and Information Technology
Tribhuvan University, Nepal

Arun K. Timalsina, Ph.D
Supervisor
Department of Electronics and Computer Engineering
Pulchowk Campus, Institute of Engineering
Tribhuvan University, Nepal

External Examiner

Internal Examiner

Date: September 7, 2017

ACKNOWLEDGEMENTS

As life is becoming easier with the advent of technology, people are becoming lazy. Because of this people are fond of copying other's creation which is called as Plagiarism. Mostly students are affected by the problem of plagiarism. Students usually copy their homework and assignments. Not only students, case of plagiarising documents and idea by some scholars are also available. Because plagiarism prevent people to exploit their creativity, which will be a serious damage to the future society, this research work is evolved.

No work would be possible without the proper guidance, it would be very difficult for me to complete this research without the valuable comments and motivation of my supervisor, Asst. Prof. Arun K. Timalsina, Ph.D. I express my sincere gratitude to him.

I like to thank Mr. Nawaraj Poudel, Head of Central Department of Computer Science and Information Technology, who always motivated me to complete the work.

I express sincere thanks to all the faculty members and staffs of Central Department of Computer Science and Information Technology who helped me in any possible way to complete the work.

I acknowledge Tribhuvan University Central Library for providing the datasets.

Life is sorrowful without friends, without friends no one is complete. I wish to thank all my friends, seniors and juniors for their cooperation. Special thanks go to Ms. Shristi Baral for her help and motivation. Deep Raj Sharma always motivated and encouraged me during the course. I could not remain silent without mentioning Mr. Ashok Pant for all his help and suggestion.

Finally, to my everything, without whom I would not have appeared to this world and would be in this place today, my parents and my family.

Rakesh Kumar Bachchan

September 7, 2017

ABSTRACT

This research work develops two frameworks for detecting plagiarism of Nepali language literatures incorporating Monte Carlo based Artificial Neural Network (MCANN) and Backpropagation (BP) neural network, which was applied for the plagiarism detection on certain document type segment. Neural Network training is considered using Monte Carlo based family of algorithms as of these algorithms superiority and robustness. Both the frameworks are tested on two different datasets and results were analysed and discussed. Convergence of MCANN is faster in comparison to traditional BP algorithm. MCANN algorithm achieve a convergence in the range of 10^{-2} to 10^{-7} for the training error in 40 epochs while general BP algorithm is unable to achieve such a convergence even in 400 epochs. Also, the mean accuracy of BP and MCANN are respectively found to be in the range of 98.657 and 99.864 during paragraph based and line based comparison of the documents. Thus, MCANN is efficient for plagiarism detection in comparison to BP for Nepali language documents.

Keywords: *Plagiarism; Monte Carlo Method; Artificial Neural Network; Backpropagation;*

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
BP	Backpropagation
MCANN	Monte Carlo based Artificial Neural Network
NUTS	No-U-Turn Sampler
PCA	Principal Component Analysis

Chapter 1

INTRODUCTION

1.1 Introduction

Using documents of others without any reference or violating the copyright rules making the document as our own, is said to be plagiarism. Plagiarism detection is the act of finding the originality of a document i.e. whether a document or idea is of the same person who is claiming about it. Question regarding the act of plagiarism is obvious, i.e., why do anyone copy others idea or creation without giving any reference to the original creator? What made people to copy others? Obviously, because of laziness and human nature of not trying to use their mind. Since, copying another work without any reference and making it as own is "theft of other property", Plagiarism is a big crime and it must be stopped. Although it is very difficult to completely wipe out the act of plagiarism because of its relation to human morality, it could be controlled by act of detection. Because of avalanche of electronic documents over the internet, contents about any topic could be easily found which is the main reason behind plagiarism. Plagiarism not only means using other's document but using ideas, concepts, thought of others without their consent. The only way to stop the act of plagiarism is to change the human morality which is very difficult but it could be controlled by detection which is the main aim of this research.

Detecting Plagiarism is necessary for several reasons. Some of them could be:

- For motivating the original author.
- To check if the person who is claiming about a document is his original idea or he is just lying.
- To Protect the copyright act.
- To allow students use their own idea rather than using others idea, which will definitely leads to new ideas in the concerned field.

In this research work Artificial Neural Network which is the most promising model simulating

the biological neural network is combined with one of the most famous class of randomized algorithm, Monte Carlo Method, and is then trained for stepping towards detecting plagiarism.

Research shows that the act of plagiarism is very common in schools and universities because of student's negligence towards doing assignment and homework. And it is most challenging task to control such activity. Thus, tools for checking plagiarism in the written documents is important.

A lot of research work has been carried out for detecting plagiarism in English documents and some other language documents like Arabic, Chinese and others. No any research work for detecting plagiarism in Nepali language documents is ever found. This research work focuses on detecting plagiarism in Nepali language documents. Also, several works using monte carlo method and artificial neural network have been carried but none of the works related to plagiarism is found using artificial neural network based monte carlo method.

In order to complete this research work, corpus was built and preprocessing was carried out on collected data before training the neural network. The processed data became input for neural network training. Finally, the forged document was tested with the trained neural network.

This research work is beneficial for the university faculties where document plagiarism is very common. Similarly, it will be advantageous to journal publication houses, document reviewers and editors. The main advantage of this research work is to motivate students and researchers to conduct research work without stealing other ideas or documents.

1.2 Aims and Objectives

The thesis aims in using Neural network in combination with Monte Carlo algorithm to investigate reuse of text without any reference in Nepali documents. The motivation behind the work lies in the fact of excessive reuse of other's documents without any reference. The idea underlying the fact is that, the original document and plagiarised document differs in various ways. To fulfil the idea, a framework is developed which incorporates Backpropagation neural network which utilizes Monte Carlo method for updating its weight, which is proposed to be improving the act of detecting text reuse without reference.

The principle objective of this research work is to develop the framework using Monte - Carlo

based artificial Neural Network. In particular the following objectives are considered:

1. To develop the framework for plagiarism detection of Nepali language based documents.
2. To investigate the performance of the framework using the MC based ANN.

1.3 Scope

The thesis attempts to study the act of plagiarism in various datasets including different thesis of different Nepali language documents. Monte Carlo method is used for generating random numbers which is used for adjusting the weights in the neural network. This research work is limited to extrinsic detection of plagiarised text.

1.4 Structure of the Thesis

First chapter introduces the thesis along with its scope and applications. Complete roadmap of the thesis is detailed below.

Chapter 2 gives clear view of what have been carried out in the area of detecting plagiarism and the limitations with the methodology involved. Different types of plagiarism detection methodologies are also discussed. Theoretical background of the research is introduced and discussed in this chapter. The chapter concludes with a general description of evaluation approaches used in automatic plagiarism detection.

Experimental setup for the research is discussed in chapter 3. Details of the corpus and the proposed framework along with the tools used for implementation is discussed and understood in this chapter.

Chapter 4 details the result of the experiment in different datasets. Results are also discussed for their correctness and acceptance.

The concluding chapter entitled Conclusion concludes the chapter explaining what have been done in the thesis and how it could be used and finishes by suggesting future works.

Chapter 2

LITERATURE REVIEW

2.1 Previous works

It is not the case that plagiarism detection tools are not available in the market. There are lots of plagiarism checker tools (e.g., Turnitin, Eve2, CopyCathGold, etc.), still plagiarism detection is a difficult task because of huge amount of information available online [2].

In the study done by Lukashenko et. al. [3], different ways of reducing plagiarism along with widely used detection tools are discussed. Similarity measurement has been done in several works for which different metrics are used. Some metrics are Corpal metric which operates on entire corpus of documents; Superficial metrics is a measure of similarity that can be gauged simply by looking at one or more documents and it does not require the knowledge of the linguistic features; Structural metric is a measure of similarity that requires knowledge of the structure of one or more documents [3]. Similarly, statistical methods are also implemented. In many cases similarity scores between two documents is calculated as Euclidean distance between document vectors [3].

Two types of plagiarism detection method have been investigated in literatures: *Intrinsic and External Plagiarism detection*. In *Intrinsic plagiarism detection* method, identification of the document is done by checking its writing pattern, i.e., whether a document is written by a single author or not, if not which part of it is plagiarised. It is not compared with other document. In *External plagiarism detection* method document is compared with other documents for checking the document similarity.

Dara Curran [4], combined genetic algorithm with neural network for intrinsic plagiarism detection. Approach of Dara Curran [4] consists of two parts: document pre-processing and neural network evolution. During **pre-processing** several stylometric features such as Number

of punctuation marks, Sentence length (number of characters), Sentence word frequency class, etc., are extracted and an average document wide value for each feature are calculated. The differences between each sentence's stylometric features and the average document-wide value is calculated for each stylometric feature and stored as a vector for each sentence. The resulting difference vector gives an indication of the divergence of a particular sentence from the average and is employed as the input for the neural network. Each difference measure is normalised to between 0 and 1. During **neural network evolution** stage Neuro-evolution of Augmenting Topologies (NEAT) encoding is employed. The neural network consists of 10 input nodes (one for each of the stylometric measures) and one output node (where an output of 0 indicates no plagiarism and 1 indicates plagiarism). The intermediary connections and hidden nodes are determined by the evolutionary process. An initial population of random neural networks is generated and for each individual neural network is presented with a number of plagiarised and non-plagiarised difference vectors taken from the pre-processed corpus. The fitness function of the neural network examines the output of the network and calculates the mean square error.

Salunkhe and Gawali in their research work [5], have used Temporal Difference (TD) algorithm of reinforcement learning for detecting plagiarism among documents. It improves the system speed of data retrival from database and also the plagiarism detection accuracy.

Salha Mohammed Alzahrani and Naomie Salim [6], in their research work have proposed statement based approach for detecting plagiarism in Arabic scripts using Fuzzy set information retrieval method. Here fuzzy-set IR model is adapted and used with Arabic language for detecting plagiarized statements based on the degree of membership between words.

Shanmugasundaram Hariharan [2], carried out plagiarism detection using similarity analysis where similarity is estimated using several measures like cosine, dice, jaccard, hellinger and harmonic. In this paper solution for "copy paste" and "paraphrasing" type of plagiarisms is identified.

In the research work considered by Efstathios Stamatatos [7], Plagiarism detection is done without removing the stop words. This method is based on structural information rather than content information. Stopword n-grams are able to capture syntactic similarities between suspicious and original documents and they can be used to detect the plagiarized passage boundaries is shown.

Freitas et. al, in their work discussed a novel strategy to train neural network using sequential Monte Carlo methods where they have used sampling techniques and illustrate their performance on some problems. More precisely, they address the problem of pricing option contracts, traded in financial markets. A new algorithm named Hybrid SIR (hybrid gradient descent/sampling importance resampling algorithm) was also proposed in the same work [8].

Man Yan Miranda Chong [9], in his doctoral thesis, considered natural language processing techniques and deep learning scenario for external plagiarism detection. In the research, a framework is proposed in which the role of machine learning is investigated. Also, the effect of applying the proposed framework in small and large-scale corpus is explored. Further experiments show that combining shallow and deep techniques helps improve the classification of plagiarised texts by reducing the number of false negatives.

2.1.1 PAN workshop and competition

Workshop entitled **"Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection"**, in conjunction with the 30th Annual International ACM SIGIR conference was organised in 2007. During the workshop only one paper discussed on the intrinsic plagiarism case while other discusses on authorship identification and near-duplicate music documents. The workshop concluded that it is necessary to segment long texts in a document to chunks, and raised two main issues [9]:

- the lack of a benchmark corpus to evaluate plagiarism detection systems.
- the lack of an effective plagiarism detection tool that does not trade off computational cost with performance.

This workshop plays a keyrole in first PAN plagiarism detection competition organised in 2009 and it focuses on the following tasks: [9].

- Plagiarism analysis
- Authorship identification
- Near-duplicate detection.

Continuity to the workshop in conjunction with the 18th European Conference on Artificial Intelligence is given in 2008 by organizing another workshop entitled **"Uncovering Plagiarism,**

Authorship and Social Software Misuse" which focuses on the first two points of previous workshop and adding one another point of *social software misuse*.

Similarly, the third PAN workshop on **"Uncovering Plagiarism, Authorship and Social Software Misuse"** was held in conjunction with the 25th Annual Conference of the Spanish Society for Natural Language Processing in 2009. The aims of the workshop remained the same as the 2008 workshop. The workshop was co-organised with the first International Competition on Plagiarism Detection. The focus was shifted from bringing together theoretical research in the field to a more competitive development workshop. The competition consisted of two subtasks: external plagiarism detection and intrinsic plagiarism detection. There was a total of 13 groups participating in the competition. The competition was based on a large-scale artificially created plagiarism corpus and provided an evaluation framework for plagiarism detection. Nine groups entered in the external plagiarism detection task and three groups entered in the intrinsic plagiarism detection task, with one group entering in both tasks [9].

More research on the topic of plagiarism detection is done between 2010 and 2013 [9]. Several research works were submitted to the PAN workshop held in respective years.

2.2 Neural Network

Since learning is the result of communication between several neurons which is actually because of interconnection of a large number of neurons. Because of the highly inter - connected neurons learning seems to be feasible in human. Neural Network although does not completely mimic the biological neural architecture but it resembles with the biological neural network to some extent. Also, it is an attempt to mirror the biological neural network, hence it is used for detecting plagiarism during the work. Actually, it is an information processing paradigm which is inspired by the way biological nervous system process information [10].

Neural network is of two types:

(a) Feed forward Neural Network: This type of neural network consists of a layer of processing units, each layer of which forward its input to the next layer with the help of connection strength also called as weight. No backward propagation of the input is allowed. Architecture of such type of network is shown in figure 2.1.

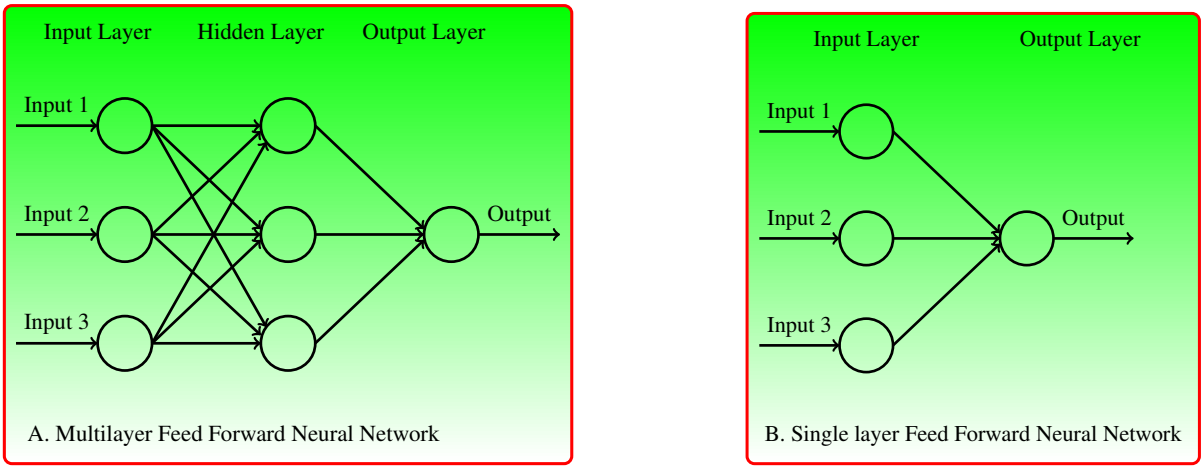

Figure 2.1: Feed Forward Neural Network

(b) Feed backward Neural Network (Recurrent Neural Network): In recurrent neural network, output of one node could be input to the same node as well as to other nodes. Architecture of such type of network is shown in figure 2.2.

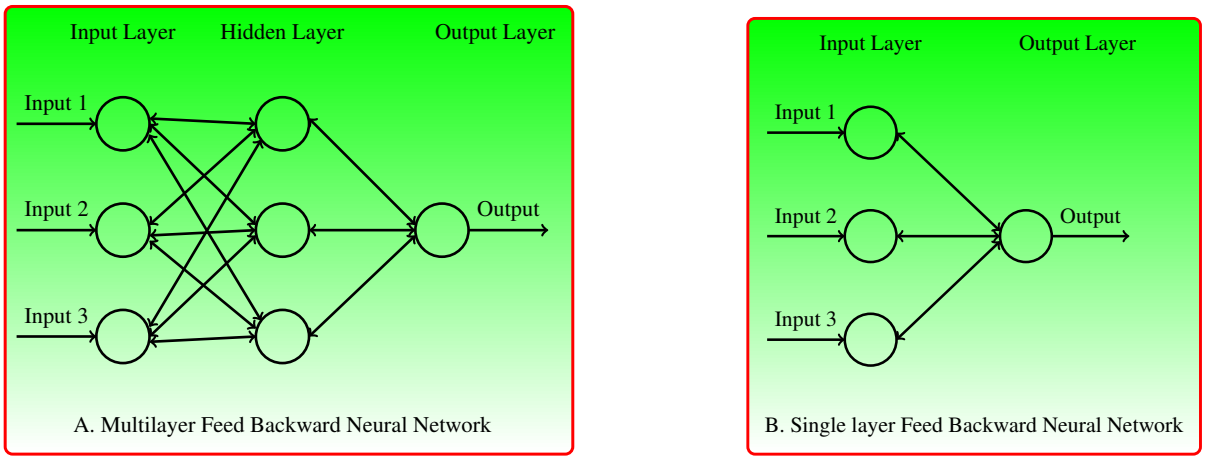

Figure 2.2: Feed Backward (Recurrent) Neural Network

For this thesis, feedforward neural network is used which was trained using Monte Carlo algorithm.

2.2.1 Backpropagation Neural Network

The Backpropagation algorithm is famous for training neural network. It uses gradient descent method for minimizing the error. Error is the difference between target output and actual output. The algorithm for backpropagation can be categorized into four phases:

1. Initialization of Weights and biases (Initialize network)
2. Feed Forward
3. Backpropagation of errors
4. Update weight and biases

The above phases are explained below:

1. **Initialization of Weights and biases (Initialize network):** Each neuron has a set of weights that need to be maintained. One weight for each input connection and an additional weight for the bias. These weights and biases are chosen randomly to small random values. This is the initialization of the network.
2. **Feed Forward:** During this phase, output of the network is calculated by transmitting input signal to each layer until the output layer. This is called as forward propagation.
3. **Backpropagation of errors:** The deviation of the actual output from the target output is calculated in this phase and the deviation which is actually the error is propagated back to the previous layer. The error is back propagated by weighting it by its weight in the previous layer and the gradient of the associated activation function.
4. **Update weight and biases:** Finally, the weights and biases associated with each unit are updated using the δ factor obtained from the previous stage and the activation. This update process continues till the threshold value or until the error is sufficiently low.

2.3 Monte Carlo Method

Monte Carlo Method, a randomized algorithm, was used for updating the weights during the network training. For the purpose, some samples are drawn from the posterior distribution of cosine and jaccard similarity vectors. Generally, the method is used for generating samples from the state space in such a way that the samples resembles the target distribution. The posterior is calculated using $NUTS$ sampler as discussed in [11]. The random samples are drawn from the posterior distribution of parameters during “learning phase”.

Chapter 3

METHODOLOGY

3.1 Dataset

The corpus for Nepali documents used in this thesis was prepared by Bam [1], that was collected from various daily, weekly and monthly Nepali newspapers like Kantipur, Gorkhapatra, and so on during 2012-2013. The dataset consists of different political, educational, biography, sports, stock exchange news. The dataset statistic is given in table 3.1.

Another corpus of Nepali language used in this thesis consists of 11 different theses collected from the Central Library Database. The statistic for the corpus is shown in table 3.2. Both of the datasets consists of copy paste type of plagiarism.

Table 3.1: Statistics of dataset by Bam [1]

Filename	Number of Paragraphs	Number of words
Train1.txt	3	1225
Train2.txt	3	661
Train3.txt	7	2416
Train4.txt	4	775
Train5.txt	9	4177
Train6.txt	6	2728
Train7.txt	10	1480
Test1.txt	6	301
Test2.txt	27	1426
Test3.txt	72	8294
Test4.txt	36	7404
Test5.txt	36	10858
Test6.txt	76	10519
Test7.txt	63	14114
Test8.txt	6	5571
Test9.txt	15	6194
Test10.txt	3	12092

Table 3.2: Statistics of Nepali Language Thesis Dataset

Filename	Number of Paragraphs	Number of words
लमजुङ भोर्लेटार क्षेत्रमा प्रचलित लोकगीतको सङ्कलन वर्गीकरण र विश्लेषण	668	14113
नेपाली उपन्यासको सङ्कलनमा त्रिभुवन विश्वविद्यालय केन्द्रीय पुस्तकालयको योगदान	1453	27329
शान्तिकुमारी राईको जीवनी, व्यक्तित्व र कृतित्व	1059	37673
फ़णीन्द्रराज खेतालाको जीवनी, व्यक्तित्व र कृतित्व	856	21319
सेतो बाघ उपन्यासको पात्रविधान	507	13498
नेपाली नाट्यविद्याका कृति र पत्रपत्रिकाको संरक्षणमा त्रिभुवन विश्वविद्यालय केन्द्रीय पुस्तकालयको योगदान	1839	19750
पश्चिमाञ्चल क्षेत्रमा प्रचलित नेपाली लोकगीतहरुको अध्ययन	3670	152085
माधवप्रसाद पोखरेल: जीवनी, व्यक्तित्व र कृतित्वको अध्ययन	957	35957
नुवाकोट जिल्लामा प्रचलित लोकगीतको अध्ययन	1190	28148
पुण्य निरौलाका उपन्यासमा पात्र विधान	1136	27286
सल्यानको पुर्वीक्षेत्रमा प्रचलित लोकगीतहरुको अध्ययन	1387	27610

3.2 Data Preprocessing

Preprocessing This includes paragraph segmentation, Punctuation removal, lowercasing, number removal, and stopword removal.

Paragraph Segmentation This splits the text in the documents into paragraph, allowing paragraph processing in the following stages.

Punctuation Removal This removes the punctuation symbols from the text. It helps in generalising the text for matching.

Number Replacement This replace the numbers from the text using some dummy symbol. It helps in generalising the text for matching.

Lowercasing This replace the uppercase letters in the text with corresponding lowercase character. It helps in generalising the text for matching.

Stopword Removal This replace the stopwords like “a”, “an”, “the”, “is”, “are”, “am” and so on from the text for generalising the text for matching.

The stop words of Nepali language used are: छ, र, पनि, छन्, लागि, भएको, गरेको, भने, गर्न, गर्ने, हो, तथा, यो, रहेको, उनले, थियो, हुने, गरेका, थिए, गर्दै, तर, नै, को, मा, हुन्, भन्ने, हुन, गरी, त, हुन्छ, अब, के, रहेका, गरेर, छैन, दिए, भए, यस, ले, गर्नु, औं, सो, त्यो, कि, जुन, यी, का, गरि, ती, न, छु, छौं, लाई, नि. The punctuation marks used in Nepali language are same as that used in English language except one additional “।” which is used for terminating the sentence. Table 3.3 shows result of several preprocessing tasks on an example Nepali text.

Table 3.3: Example of several preprocessing tasks on an example Nepali text

Original Text	२०१६ सालमा त्रिभुवन विश्वविद्यालय केन्द्रीय पुस्तकालयको स्थापना काठमाण्डौंको त्रिपुरेश्वरमा भएको हो। स्थापनाकालमा जम्मा १२ सय पुस्तकबाट सेवा दिन थालेको यस पुस्तकालयमा हाल विभिन्न भाषामा लेखिएका विभिन्न विषयमा केन्द्रित गरी जम्मा २ लाख ८० हजार पुस्तक र १ लाख थान विदेशी जर्नल र नेपाली अखवार तथा पत्रिकाहरु छन्।
Paragraph Segmentation	(**Paragraph 1**) (२०१६ सालमा त्रिभुवन विश्वविद्यालय केन्द्रीय पुस्तकालयको स्थापना काठमाण्डौंको त्रिपुरेश्वरमा भएको हो।) (**Paragraph 2**) (स्थापनाकालमा जम्मा १२ सय पुस्तकबाट सेवा दिन थालेको यस पुस्तकालयमा हाल विभिन्न भाषामा लेखिएका विभिन्न विषयमा केन्द्रित गरी जम्मा २ लाख ८० हजार पुस्तक र १ लाख थान विदेशी जर्नल र नेपाली अखवार तथा पत्रिकाहरु छन्।)
Punctuation Removal	२०१६ सालमा त्रिभुवन विश्वविद्यालय केन्द्रीय पुस्तकालयको स्थापना काठमाण्डौंको त्रिपुरेश्वरमा भएको हो स्थापनाकालमा जम्मा १२ सय पुस्तकबाट सेवा दिन थालेको यस पुस्तकालयमा हाल विभिन्न भाषामा लेखिएका विभिन्न विषयमा केन्द्रित गरी जम्मा २ लाख ८० हजार पुस्तक र १ लाख थान विदेशी जर्नल र नेपाली अखवार तथा पत्रिकाहरु छन्
Number Replacement	[#] सालमा त्रिभुवन विश्वविद्यालय केन्द्रीय पुस्तकालयको स्थापना काठमाण्डौंको त्रिपुरेश्वरमा भएको हो स्थापनाकालमा जम्मा [#] सय पुस्तकबाट सेवा दिन थालेको यस पुस्तकालयमा हाल विभिन्न भाषामा लेखिएका विभिन्न विषयमा केन्द्रित गरी जम्मा [#] लाख [#] हजार पुस्तक र [#] लाख थान विदेशी जर्नल र नेपाली अखवार तथा पत्रिकाहरु छन्
Stopword Removal	२०१६साल त्रिभुवन विश्वविद्यालय केन्द्रीय पुस्तकालय स्थापना काठमाण्डौं त्रिपुरेश्वर। स्थापनाकाल जम्मा १२ सय पुस्तकबाट सेवा दिन थाले यस पुस्तकालय हाल विभिन्न भाषा लेखिए विभिन्न विषय केन्द्रित जम्मा २ लाख ८० हजार पुस्तक १ लाख थान विदेशी जर्नल नेपाली अखवार पत्रिका।

3.3 Vector Processing and Dimensionality Reduction

After Preprocessing the dataset was ready for the further processing. The data returned by the preprocessing stage was then vectorized using Term Frequency - Inverse Document frequency (TF-IDF).

TF - IDF Term Frequency is defined as the number of times a particular term appears in the document. Inverse Document Frequency is used for measuring the importance of a term in document. TF-IDF weight is a statistical measure used to evaluate importance of word to document in a corpus. The importance of the word in the document is directly proportional to the frequency of the word in the corpus.

$$\text{TF}(t) = \frac{\text{Number of times term t appears in a document}}{\text{Total number of terms in the document}} \tag{3.1}$$

$$\text{IDF}(t) = \log_e \frac{\text{Total number of documents}}{\text{Number of documents with term t in it}} \tag{3.2}$$

The TF-IDF vectorizer converts the document from text to number, which is what is required for further computation including neural network training.

After vectorizing the document, it's dimensionality was reduced using Principal Component Analysis (PCA) for reducing the processing complexity.

PCA The Principal Component Analysis is a method used for analysing the dimension of the data because all the dimension may not be relevant for computation and only increase the computational complexity. The algorithm for PCA can be written out as [13]:

- At first the N datapoint are written as row vectors
- Then the vectors are kept into a matrix of size $M \times N$
- Mean of each column is then subtracted from the column elements and kept in separate matrix B
- Covariance matrix of the above vector is then computed using formula $C = \frac{1}{N} B^T B$
- Eigenvalues and Eigenvectors of C is then computed. Those values are then sorted in descending order

- Finally, eigenvalues with values less than some threshold values are rejected, and other are kept as the dimension.

After performing all the preprocessing, vectorizing and dimensionality reduction task, the data was ready for similarity checking.

3.4 Similarity Calculation

Cosine Similarity and Jaccard Similarity between each paragraph vector from the source data and suspicious data was then calculated.

For calculating the Cosine similarity between the vectors, the following formula is used:

$$\cos\theta = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|\|\vec{b}\|} \tag{3.3}$$

Where,

a and b are the vectors of *suspicious paragraph* and *source paragraph* respectively.

Similarly, the formula used for calculating the Jaccard similarity is:

$$J(A,B) = \frac{\|\text{Intersection}(A,B)\|}{\|\text{Union}(A,B)\|} \tag{3.4}$$

Where,

A and B are the vectors of *suspicious paragraph* and *source paragraph* respectively.

Using equations (3.3) and (3.4), the different similarity scores between each suspicious and source paragraph are calculated, which was used for training the neural network.

3.5 Processing for Learning

Two learning algorithms, namely, Backpropagation and Monte Carlo Artificial based Neural Network were used for training purpose. Input to the neural network are cosine similarity and Jaccard similarity scores. The output from the network is either 0 or 1, where 0 represents

the plagiarised case and 1 represents the non-plagiarised cases. The threshold value taken for indicating the document as plagiarised was ten percent.

1. **Backpropagation (BP):** The Backpropagation algorithm is used for training the neural network. Different phases of the algorithm was discussed in chapter 2 subsection 2.2.1. The different equations used during each phases are discussed below [10]:

   Network Initialization Phase

   The weights and biases used in this thesis was assigned randomly using the *random library function*.

   Feed Forward Phase

   Total input to the hidden unit was calculated using equation 3.5

$$z_{inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij} \tag{3.5}$$

   Total output from the hidden unit is calculated by applying an activation function given by equation 3.6

$$Z_j = f(z_{inj}) \tag{3.6}$$

   Output from above equation 3.6 was sent to next layer forward. Similarly, the total input and output of the output layer was calculated using equations 3.7 and 3.8

$$y_{ink} = w_{ok} + \sum_{i=1}^{n} z_j v_{jk} \tag{3.7}$$

$$Y_k = f(y_{ink}) \tag{3.8}$$

   Backpropagation Phase

   For backward propagation of error, the error term was calculated using equation 3.9:

$$\delta_k = (t_k - y_k) f(y_{ink}) \tag{3.9}$$

   each hidden layer z_j sums its delta inputs from the above layer using equation 3.10

$$\delta_{inj} = \sum_{k=1}^{m} \delta_j w_{jk} \tag{3.10}$$

The error information term in the hidden layer is calculated using equation 3.11

$$\delta_j = \delta_{inj} f(z_{inj}) \tag{3.11}$$

Weight and Bias Update Phase

Weight and bias correction term for the output unit is given by equation 3.12 and 3.13 respectively.

$$\Delta W_{jk} = \alpha \delta_k z_j \tag{3.12}$$

$$\Delta W_{ok} = \alpha \delta_k \tag{3.13}$$

And, the new weight and bias is given by equations 3.14 and 3.15 respectively.

$$W_{jk}(new) = W_{jk}(old) + \Delta W_{jk} \tag{3.14}$$

$$W_{ok}(new) = W_{ok}(old) + \Delta W_{ok} \tag{3.15}$$

Similarly, the weight and correction term for the hiddden units is given by equations 3.16 and 3.17 respectively.

$$\Delta V_{ij} = \alpha \delta_j x_i \tag{3.16}$$

$$\Delta V_{oj} = \alpha \delta_j \tag{3.17}$$

And, the new weight and bias is given by equations 3.18 and 3.19 respectively.

$$V_{ij}(new) = V_{ij}(old) + \Delta V_{ij} \tag{3.18}$$

$$V_{oj}(new) = V_{oj}(old) + \Delta V_{oj} \tag{3.19}$$

2. **Monte Carlo based Artificial Neural Network (MCANN):** Monte Carlo Method, a randomized algorithm, was used for updating the weights during the network training. For the purpose, some samples are drawn from the posterior distribution of cosine and jaccard similarity vectors. The posterior is calculated using $NUTS$ sampler discussed in [11]. The random samples are drawn from the posterior distribution of parameters during "learning phase".

The implementation was done using Python 3.6.0. The modules provided by sklearn, numpy, itertools, nltk were utilized for document preprocessing, vectorizing and similarity calculation task. Similarly, LaTeX was used for document preparation.

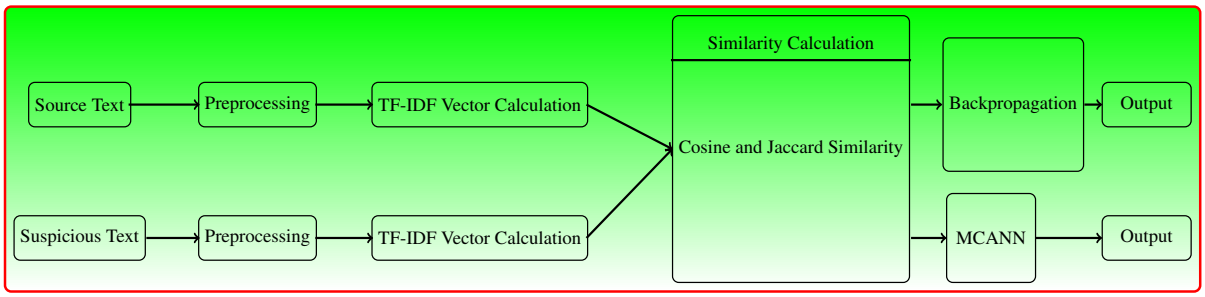

Figure 3.1: General Processing Framework

The MCANN was a three layered architecture with two hidden layers (two nodes in each layer), one input layer (two nodes) and one output layer (one node). The input layer takes the cosine and jaccard similarity vectors as input and the output layers produces either 1 or 0 as output. 1 was used for representing plagiarised case and 0 for non plagiarised case. The architecture for MCANN was shown in figure 3.2. Architecture used for backpropagation was as same as that of MCANN.

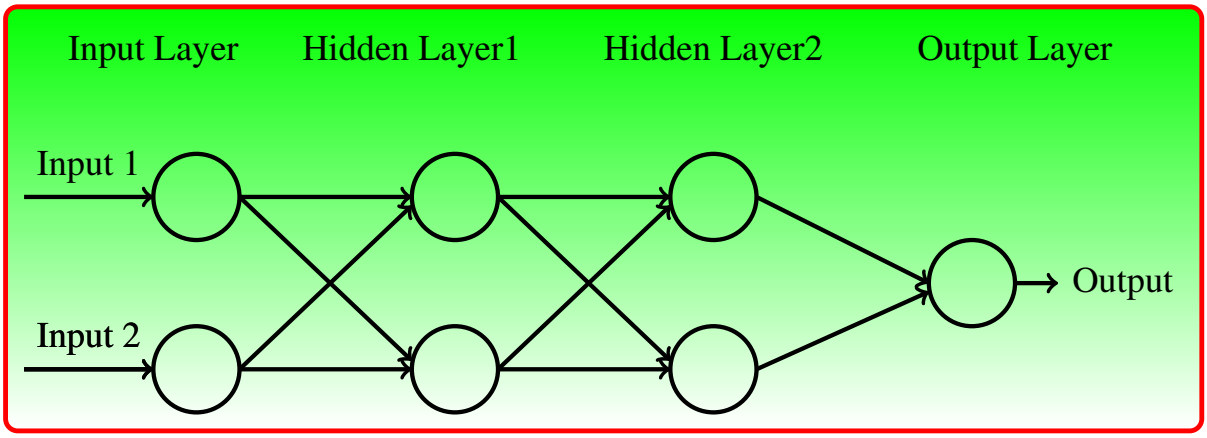

Figure 3.2: MCANN Architecture

Chapter 4

RESULTS AND ANALYSIS

4.1 Results and Analysis

In this research work, Artificial neural network (ANN) model and Monte Carlo based Artificial neural network model were developed for detecting the plagiarism of Nepali documents. Both the models were tested on several dissertations carried out in Nepali. Eleven dissertations of Nepali language were collected for the research. Similarly, testing was also carried out on the data prepared by Bam [1]. The purpose of testing was to check the accuracy of the model. The learning rate used for each experiment was 0.3.

4.2 Results of Paragraph based Comparison

4.2.1 Experiment with Nepali Thesis using BP

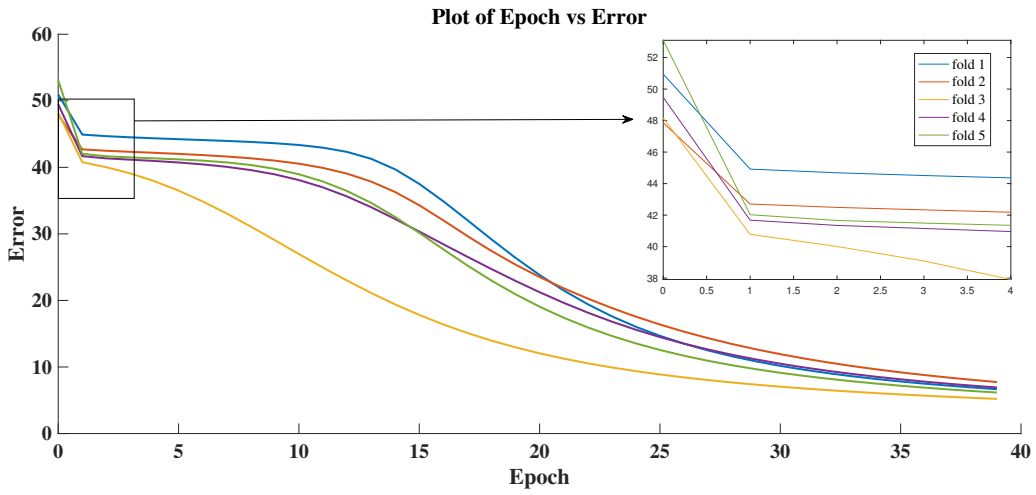

Figure 4.1: Error Vs Epoch for Nepali Thesis using BP for 40 epochs. It is the case of 5-fold cross validation.

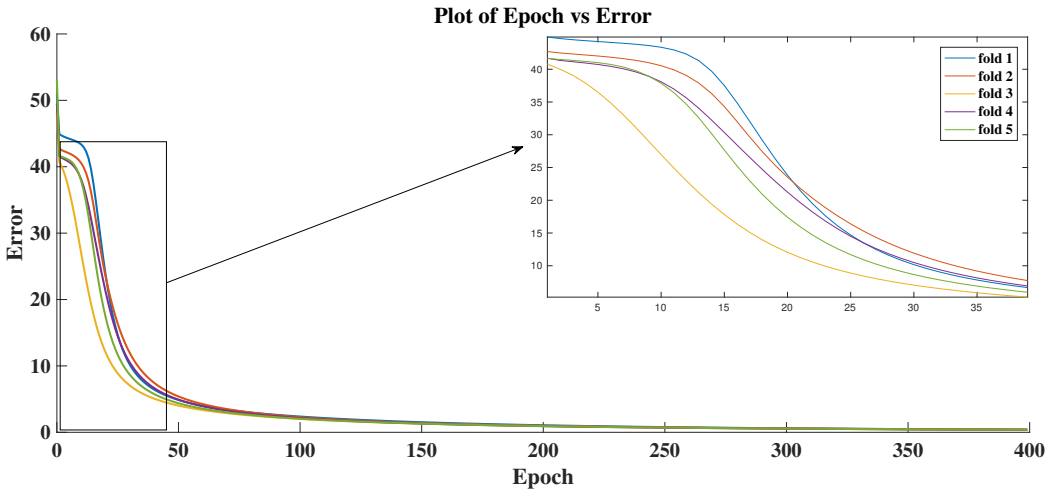

Figure 4.2: Error Vs Epoch for Nepali Thesis using BP for 400 Epochs. It is the case of 5-fold cross validation.

Figures 4.1 and 4.2 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of several thesis of Nepali languages.

The training error obtained for Nepali thesis with BP with two hidden layers and 5-fold cross validation was 6.163 (in 40 iterations) and 0.385 in 400 iterations.

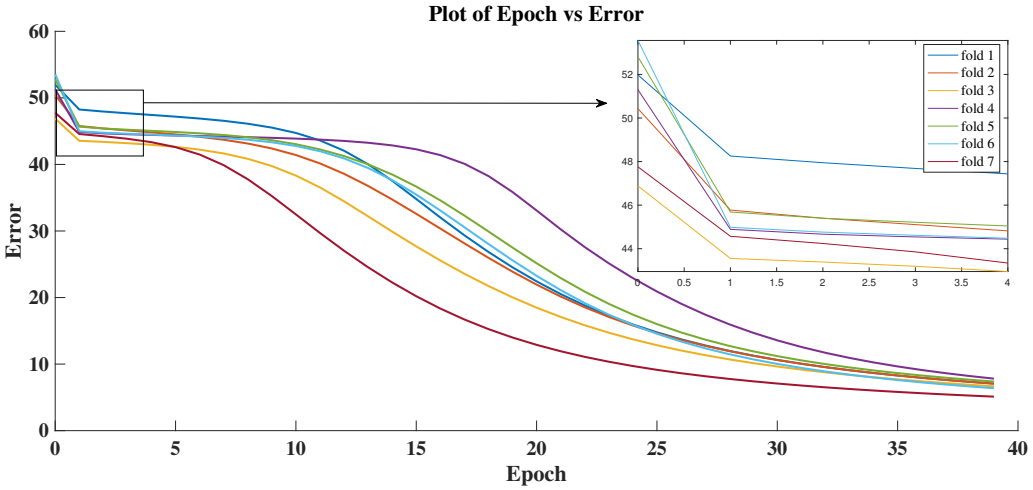

Figure 4.3: Error Vs Epoch for Nepali Thesis using BP for 40 Epochs. It is the case of 7-fold cross validation.

Figure 4.3 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of several thesis of Nepali languages.

The training error obtained for Nepali thesis with BP with two hidden layers and 7-fold cross validation was 5.111 in 40 iterations.

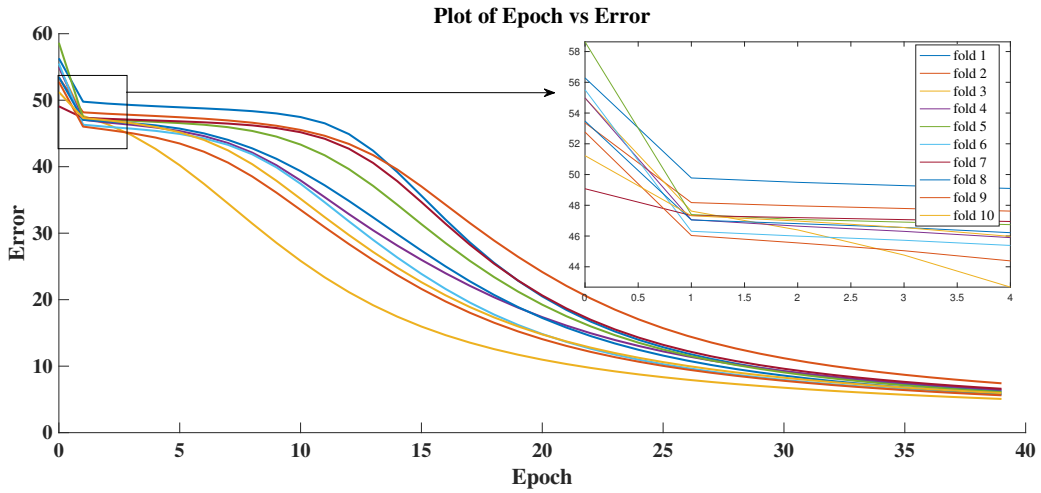

Figure 4.4: Error Vs Epoch for Nepali Thesis using BP for 40 epochs. It is the case of 10-fold cross validation.

Figure 4.4 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of several thesis of Nepali language.

The training error obtained for Nepali thesis with BP with two hidden layers and 10-fold cross validation was 5.952 in 40 iterations.

4.2.2 Experiment with Nepali Thesis using MCANN

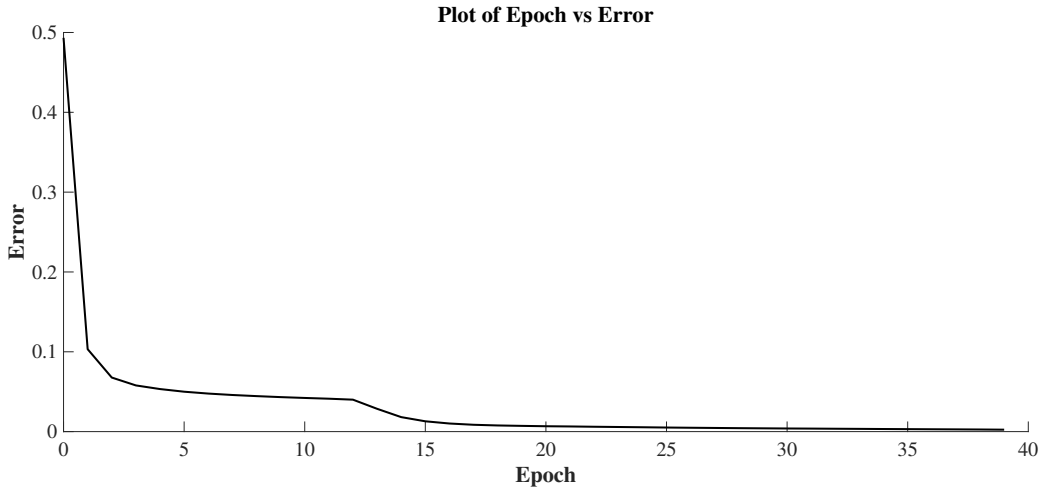

Figure 4.5: Error Vs Epoch for Nepali Thesis using MCANN in 40 epochs. Ninety percent data was used as training data and ten percent as test data.

The error obtained for Nepali thesis using 90% training data and 10% testing data with MCANN was 2.4219e-03 in 40 iterations as shown in figure 4.5 and 1.3963e-03 in 1400 iterations.

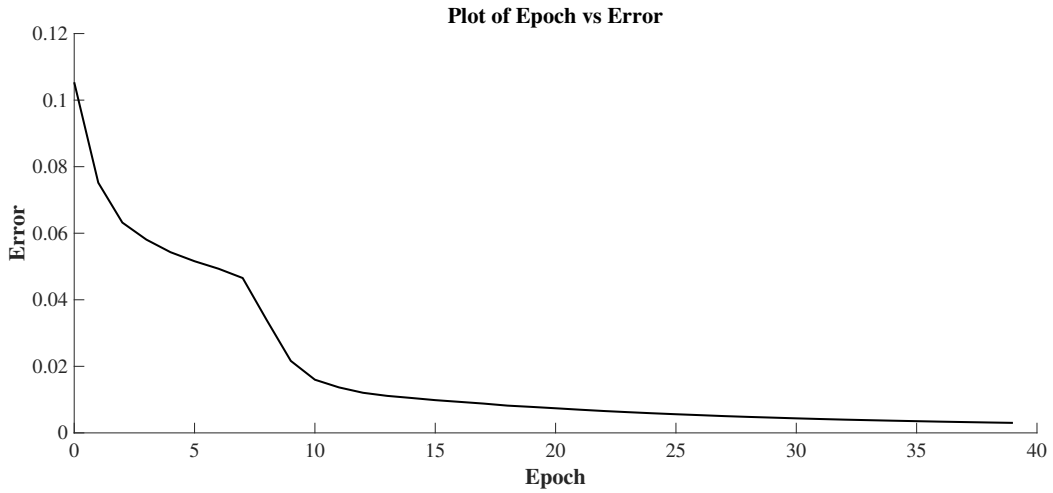

Figure 4.6: Error Vs Epoch for Nepali Thesis using MCANN in 40 epochs. Eighty percent of data was used as train data and twenty percent data as test data.

Error obtained for Nepali thesis using 80% of data as training data and 20% of data as testing data with MCANN was 3.0096e-03 in 40 iterations as shown in figure 4.6.

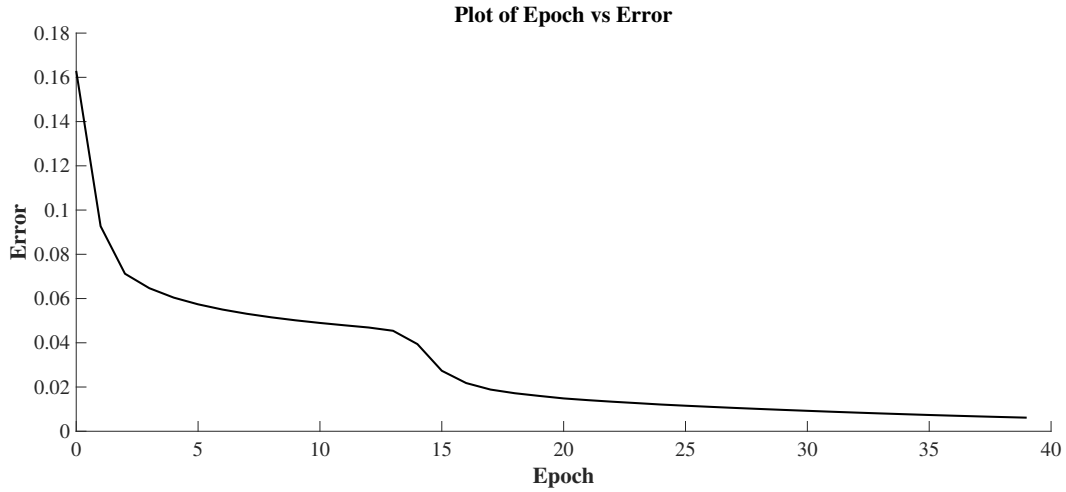

Figure 4.7: Error Vs Epoch for Nepali Thesis using MCANN in 40 epochs. Sixty percent of data was used as train data and forty percent data as test data.

Error obtained for Nepali thesis using 60% of data as training data and 40% of data as testing data with MCANN was 6.1455e-03 in 40 iterations as shown in figure 4.7.

4.2.3 Experiment with Bam data using BP

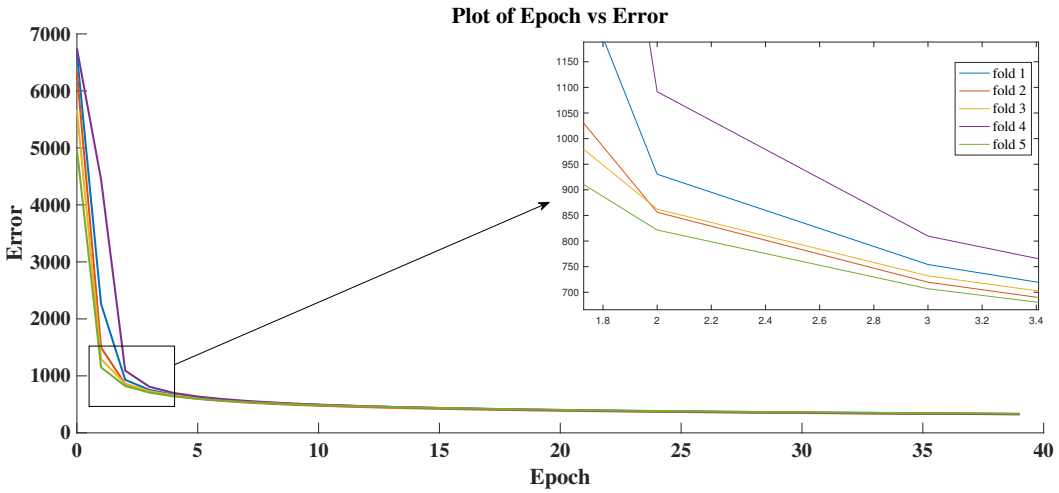

Figure 4.8: Error Vs Epoch for Bam [1] data using BP (40 epochs). It is the case of 5-fold cross validation.

Figure 4.8 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of data in Nepali language by Bam [1].

The training error obtained for above experiment with BP with two hidden layers and 5-fold cross validation was 335.854 (in 40 iterations) and 98.185 (in 1400 iterations).

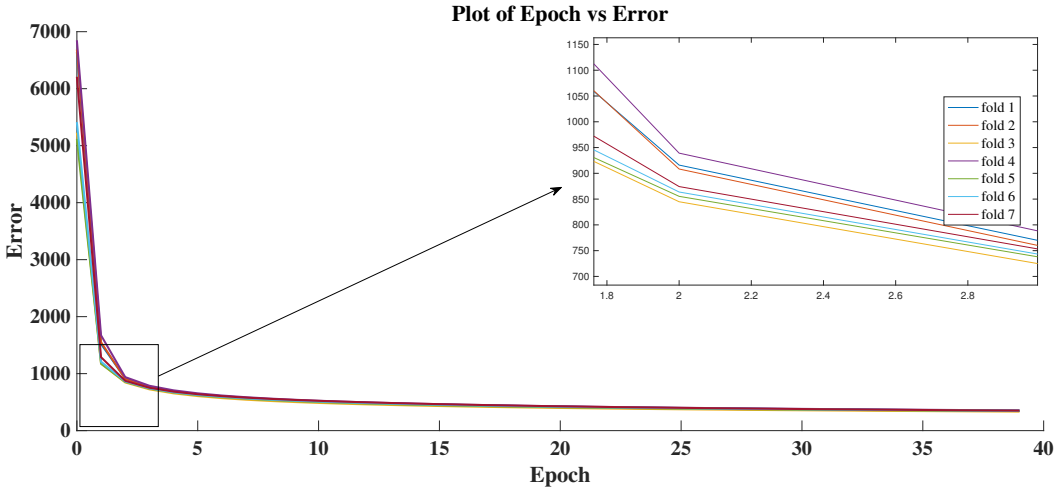

Figure 4.9: Error Vs Epoch for Bam [1] data using BP (40 epochs). It is the case of 7-fold cross validation.

Figure 4.9 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of data in Nepali language by Bam [1].

The training error obtained for above experiment with BP with two hidden layers and 7-fold cross validation was 350.929 (in 40 iterations).

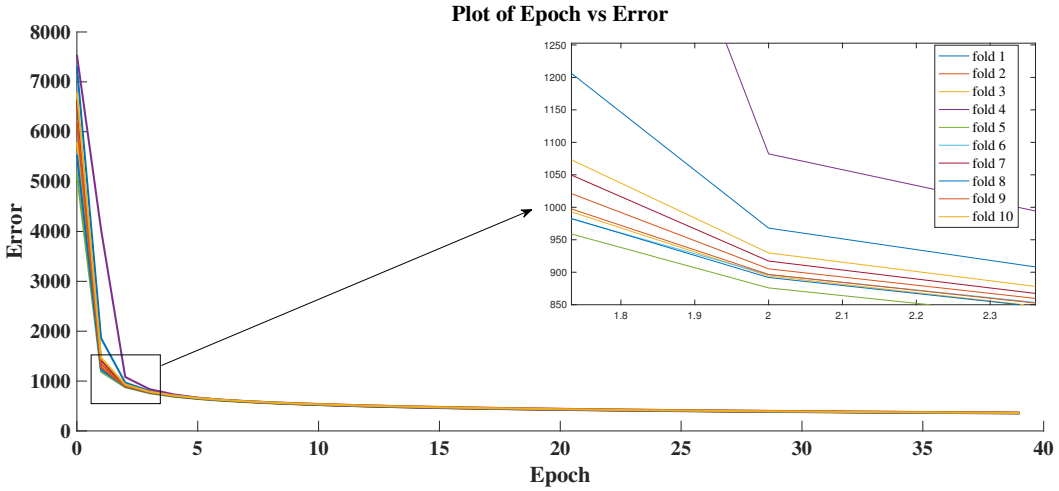

Figure 4.10: Error Vs Epoch for Bam [1] data using BP (40 epochs). It is the case of 10-fold cross validation.

Figure 4.10 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of data in Nepali language by Bam [1].

The training error obtained for above experiment with BP with two hidden layers and 10-fold cross validation was 360.370 (in 40 iterations).

4.2.4 Experiment with Nepali data collected by Bam using MCANN

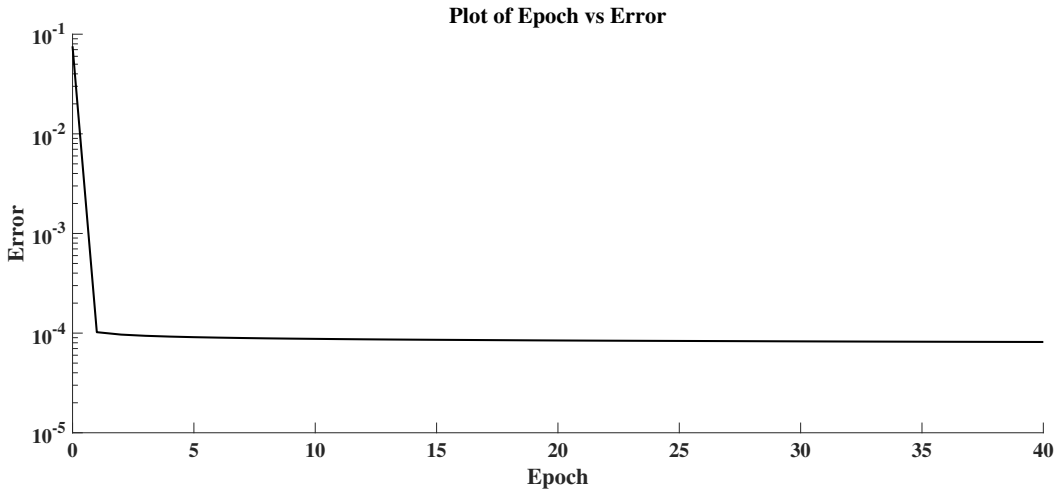

Figure 4.11: Error Vs Epoch for Bam [1] using MCANN (40 epochs). Ninety percent data was used as training data and ten percent as test data.

The error obtained for above experiment with MCANN was 8.1471e-05 (in 40 iterations) as shown in figure 4.11. In this experiment ninety percent data was used for training and ten percent was used for testing.

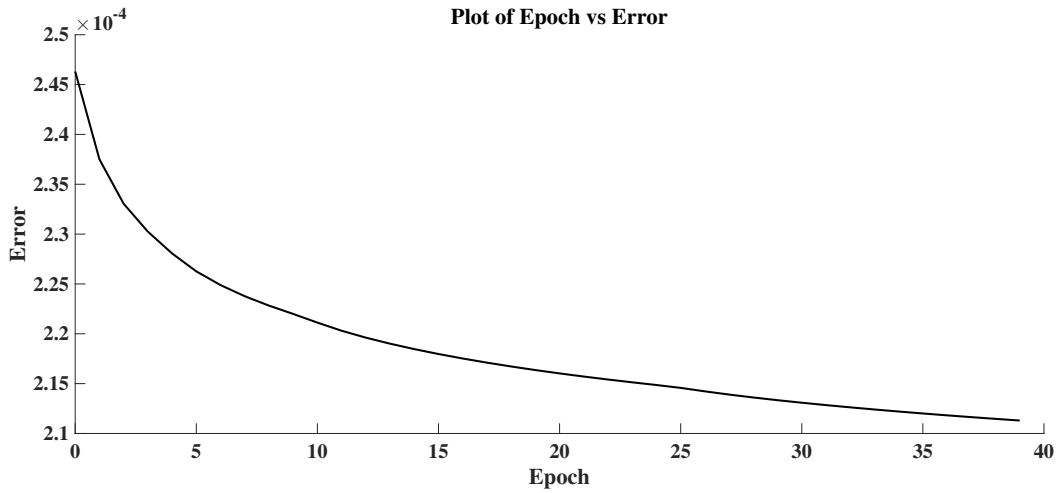

Figure 4.12: Error Vs Epoch for Bam [1] using MCANN (40 epochs). Eighty percent data was used for training and twenty percent for testing.

The error obtained while experimenting with Bam data [1] using using MCANN was 2.1130e-04 in 40 iterations as shown in figure 4.12. In this experiment eighty percent data was used for training and twenty percent for testing.

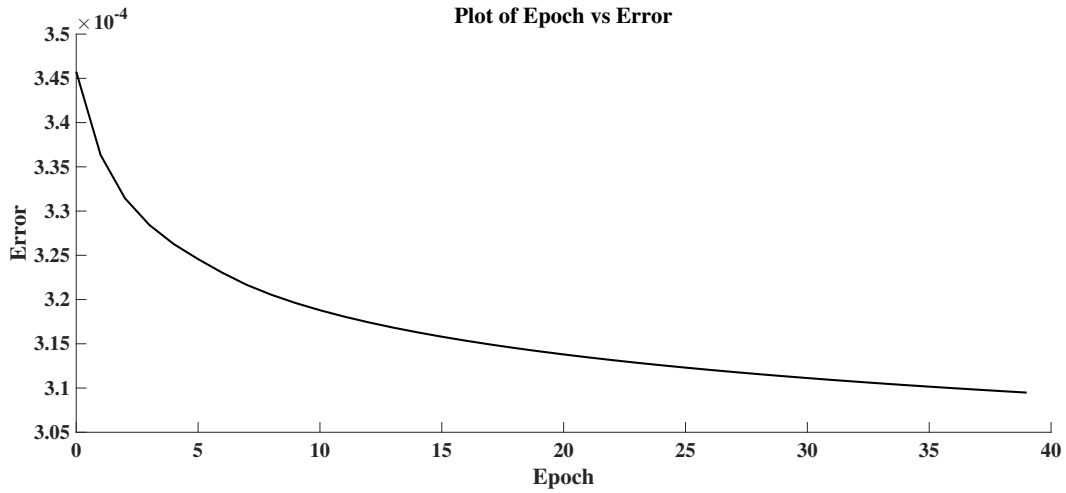

Figure 4.13: Error Vs Epoch for Bam [1] using MCANN (40 epochs). Sixty percent data was used as training data and forty percent as test data.

The error obtained for above experiment with MCANN was 3.0948e-04 in 40 iterations as shown in figure 4.13. In this experiment sixty percent of the data was used for training and forty percent was used for testing.

Table 4.1: Result of Backpropagation on Nepali Thesis and Bam Data during Paragraph based Comparison

Dataset Used	Algorithms Applied	Error obtained on different experiments		
		5-fold Cross Validation	7-fold Cross Validation	10-fold Cross Validation
Nepali Thesis	BP	6.163	5.111	5.952
Bam data	BP	335.854	350.929	360.370

The above table 4.1 lists the result of Backpropagation algorithm on different datasets. The

Table 4.2: Result of MCANN on Nepali Thesis and Bam Data during Paragraph based Comparison

Dataset Used	Algorithms Applied	Error obtained on different experiments		
		60% train & 40% test data	80% train & 20% test data	90% train & 10% test data
Nepali Thesis	MCANN	6.1455e-03	3.0096e-03	2.4219e-03.
Bam data	MCANN	3.0948e-04	2.1130e-04	8.1471e-05

above table 4.2 lists the result of MCANN on different datasets.

4.3 Results of Line Based Comparison

4.3.1 Experiment with Bam data using BP

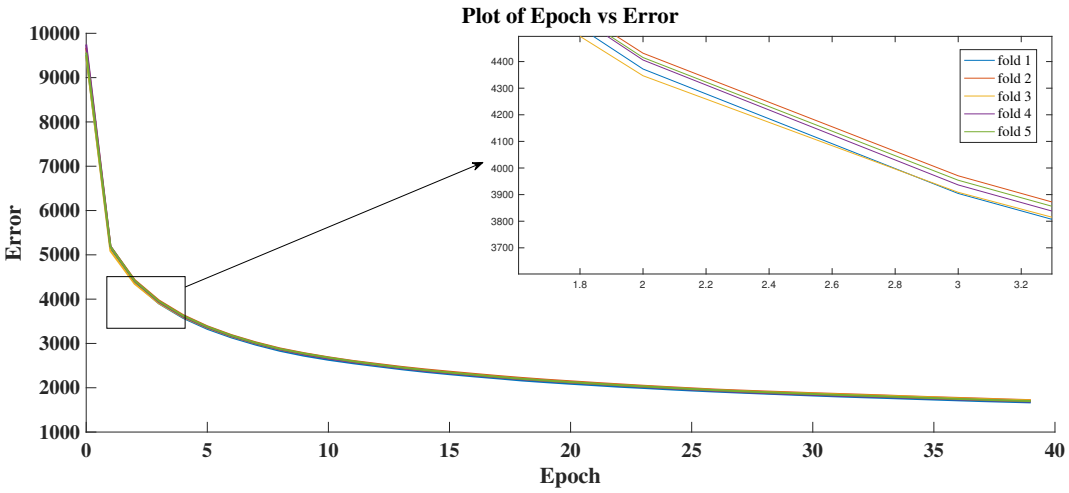

Figure 4.14: Error Vs Epoch for Bam [1] data using BP (40 epochs). It is the case of 5-fold cross validation.

Figure 4.14 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of data in Nepali language by Bam [1].

The training error obtained for above experiment with BP with two hidden layers and 5-fold cross validation was 1699.706 (in 40 iterations).

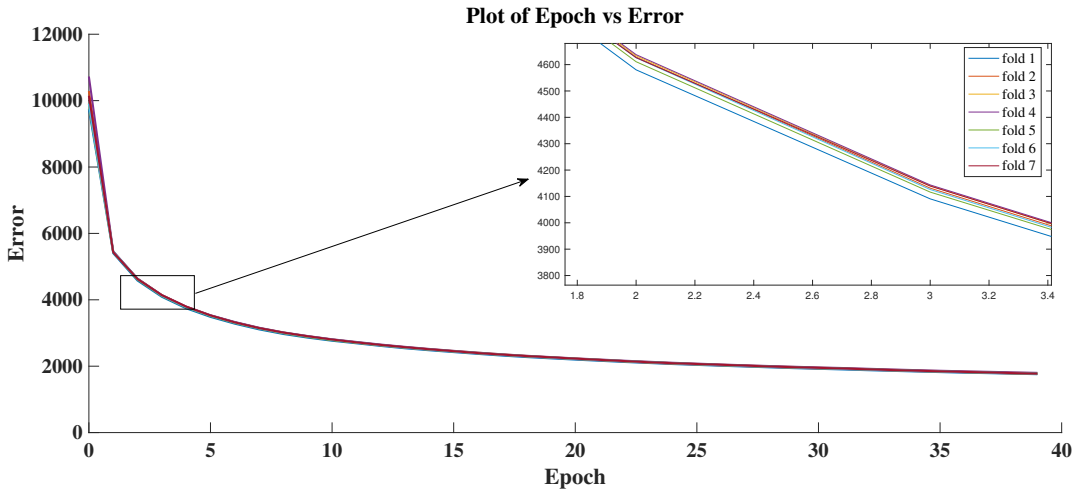

Figure 4.15: Error Vs Epoch for Bam [1] data using BP (40 epochs). It is the case of 7-fold cross validation.

Figure 4.15 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of data in Nepali language by Bam [1].

The training error obtained for above experiment with BP with two hidden layers and 7-fold cross validation was 1775.384 (in 40 iterations).

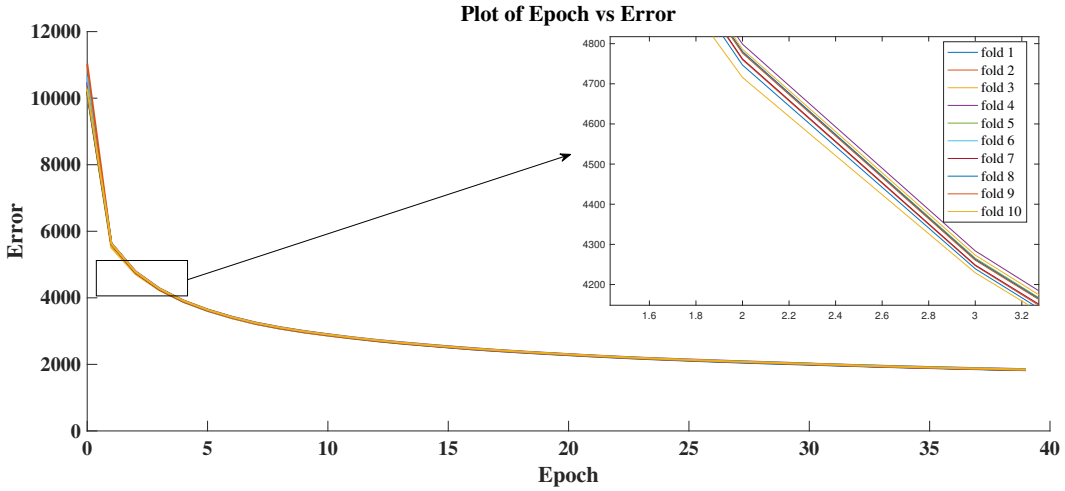

Figure 4.16: Error Vs Epoch for Bam [1] data using BP (40 epochs). It is the case of 10-fold cross validation.

Figure 4.16 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of data in Nepali language by Bam [1].

The training error obtained for above experiment with BP with two hidden layers and 10-fold cross validation was 1843.961 (in 40 iterations).

4.3.2 Experiment with Nepali data collected by Bam using MCANN

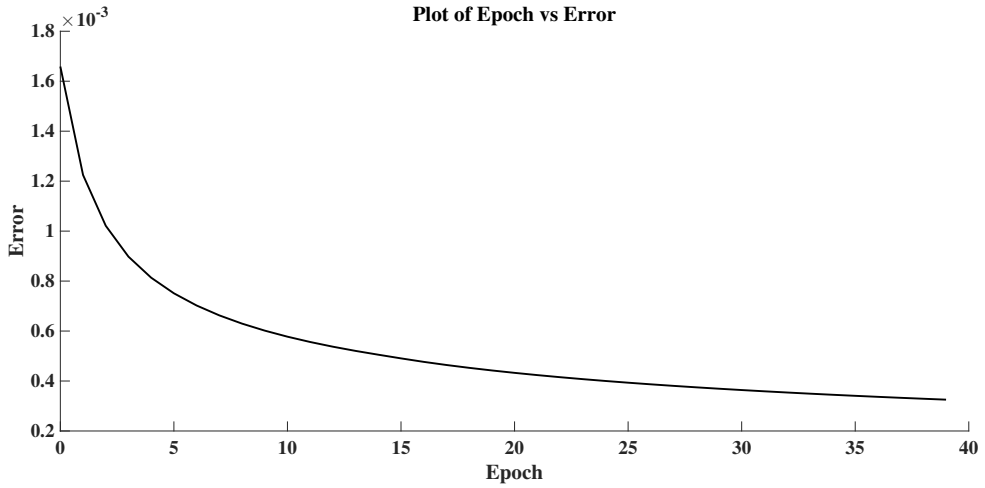

Figure 4.17: Error Vs Epoch for Bam [1] using MCANN (40 epochs). Ninety percent data was used as training data and ten percent as test data.

The error obtained for above experiment with MCANN was 3.2539e-04 (in 40 iterations) as shown in figure 4.17. In this experiment ninety percent data was used for training and ten percent was used for testing.

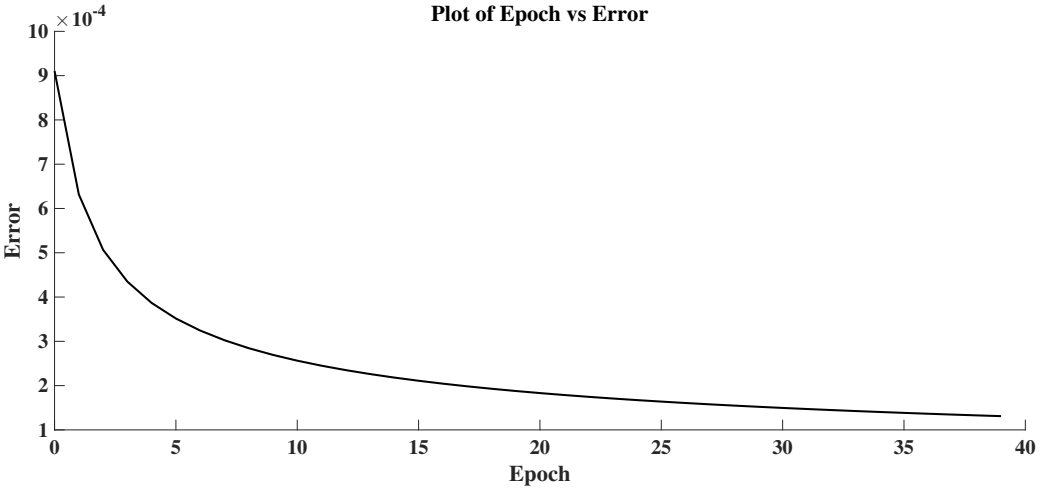

Figure 4.18: Error Vs Epoch for Bam [1] using MCANN (40 epochs). Eighty percent data was used for training and twenty percent for testing.

Error while experimenting with Bam data [1] using MCANN was 1.3106e-04 in 40 iterations

as shown in figure 4.18. In this experiment eighty percent data was used for training and twenty percent for testing.

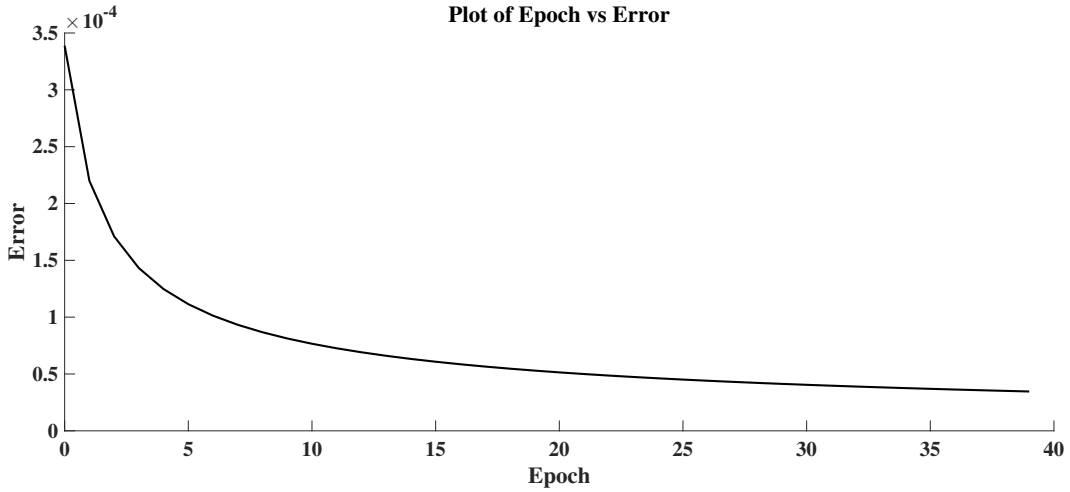

Figure 4.19: Error Vs Epoch for Bam [1] using MCANN (40 epochs). Sixty percent data was used as training data and forty percent as test data.

The error obtained for above experiment with MCANN was 3.4599e-05 in 40 iterations as shown in figure 4.19. In this experiment sixty percent of the data was used for training and forty percent was used for testing.

Table 4.3: Result of Backpropagation on Bam Data during Line based Comparison

Dataset Used	Algorithms Applied	Error obtained on different experiments		
		5-fold Cross Validation	7-fold Cross Validation	10-fold Cross Validation
Bam data	BP	1699.706	1775.384	1843.961

The above table 4.3 lists the result of Backpropagation algorithm on different datasets.

Table 4.4: Result of MCANN on Bam Data during Line based Comparison

Dataset Used	Algorithms Applied	Error obtained on different experiments		
		60% train & 40% test data	80% train & 20% test data	90% train & 10% test data
Bam data	MCANN	3.4599e-05	1.3106e-04	3.2539e-04

The above table 4.4 lists the result of MCANN algorithm on different datasets.

4.4 Results of Cluster based Analysis

To further check the performance of the algorithms, most similar looking documents of nepali thesis were selected and paragraphs based comparison were done on those documents.

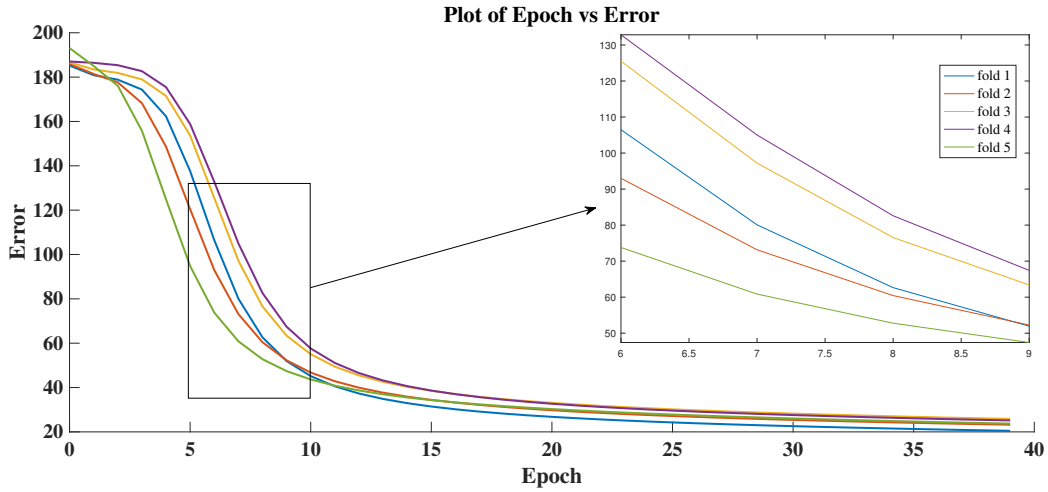

Figure 4.20: Error Vs Epoch for selected four documents using BP (40 epochs). It is the case of 5-fold cross validation.

Figure 4.20 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of selected documents of Nepali thesis. Error obtained in this case was 23.743.

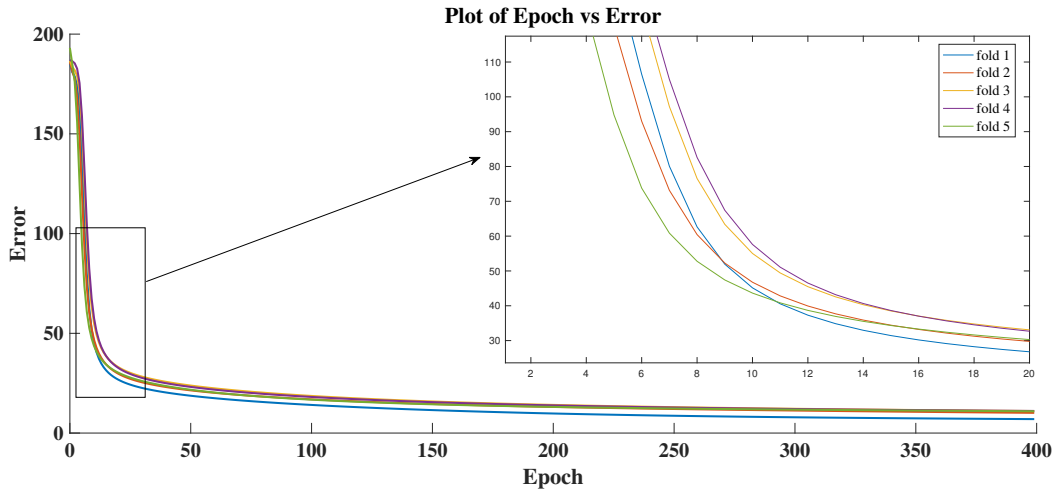

Figure 4.21: Error Vs Epoch for selected four documents using BP (400 epochs). It is the case of 5-fold cross validation.

Figure 4.21 represents the plot of error against number of epochs when BP with two hidden

layers were used for detecting the similarity of selected documents of Nepali thesis. Error obtained in this case was 10.925

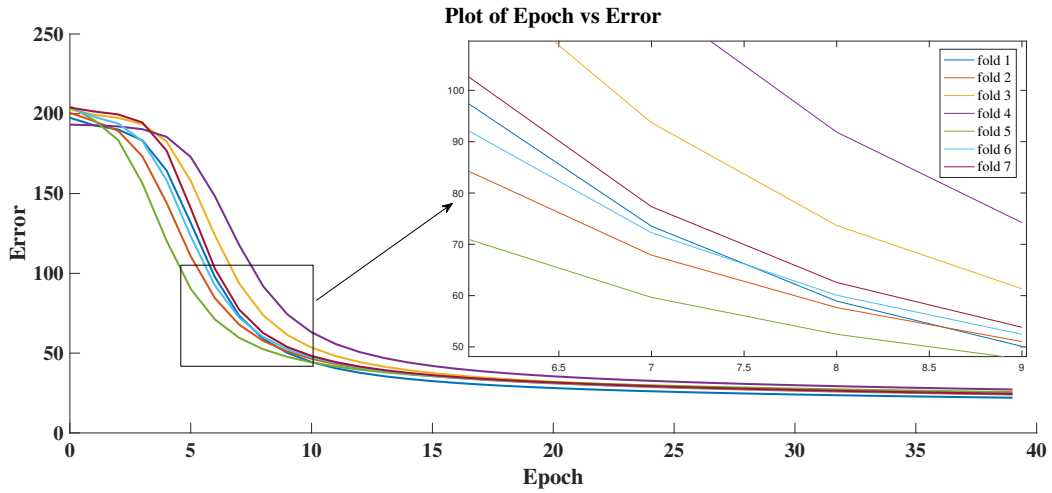

Figure 4.22: Error Vs Epoch for selected four documents using BP (40 epochs). It is the case of 7-fold cross validation.

Figure 4.22 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of selected documents of Nepali thesis. Error obtained in this case was 24.496.

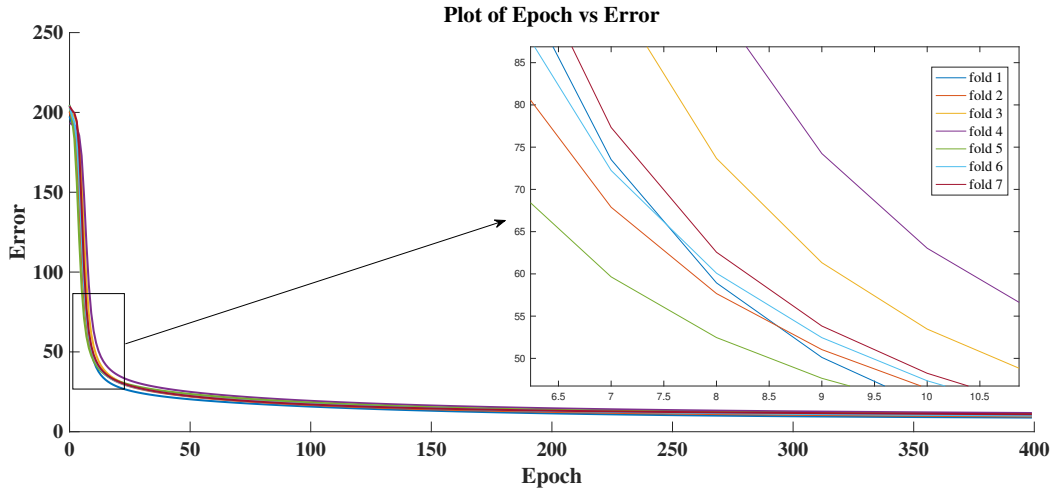

Figure 4.23: Error Vs Epoch for selected four documents using BP (400 epochs). It is the case of 7-fold cross validation.

Figure 4.23 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of selected documents of Nepali thesis. Error obtained in this case was 10.926.

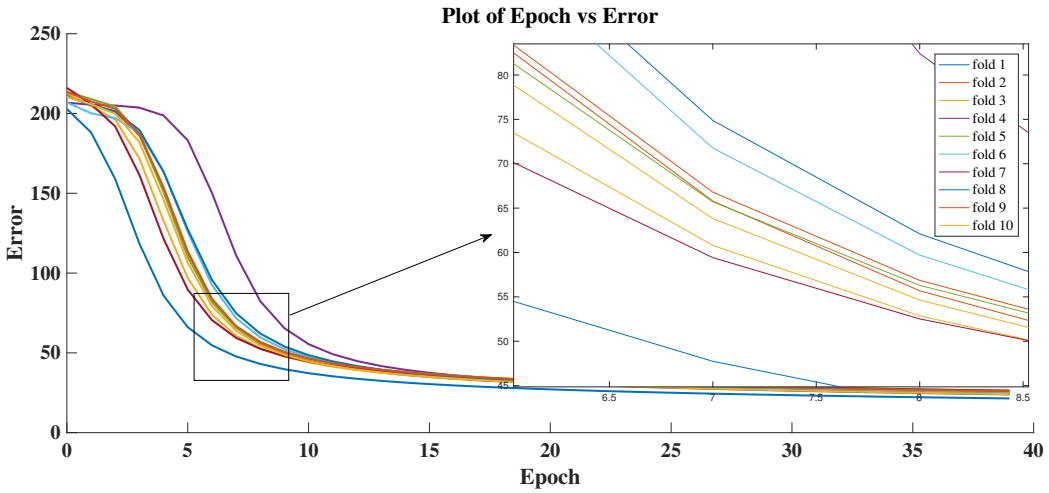

Figure 4.24: Error Vs Epoch for selected four documents using BP (40 epochs). It is the case of 10-fold cross validation.

Figure 4.24 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of selected documents of Nepali thesis. Error obtained in this case was 24.137.

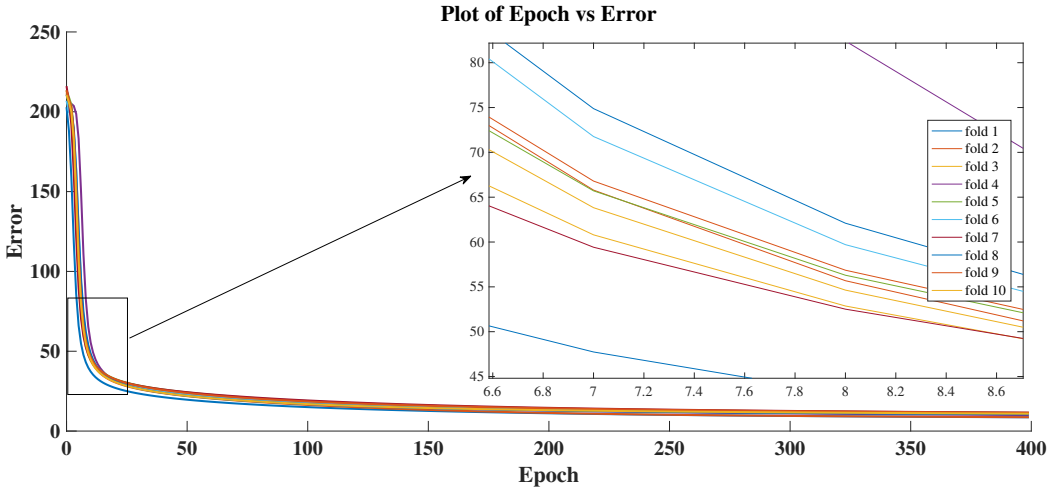

Figure 4.25: Error Vs Epoch for selected four documents using BP (400 epochs). It is the case of 10-fold cross validation.

Figure 4.25 represents the plot of error against number of epochs when BP with two hidden layers were used for detecting the similarity of selected documents of Nepali thesis. Error obtained in this case was 10.880.

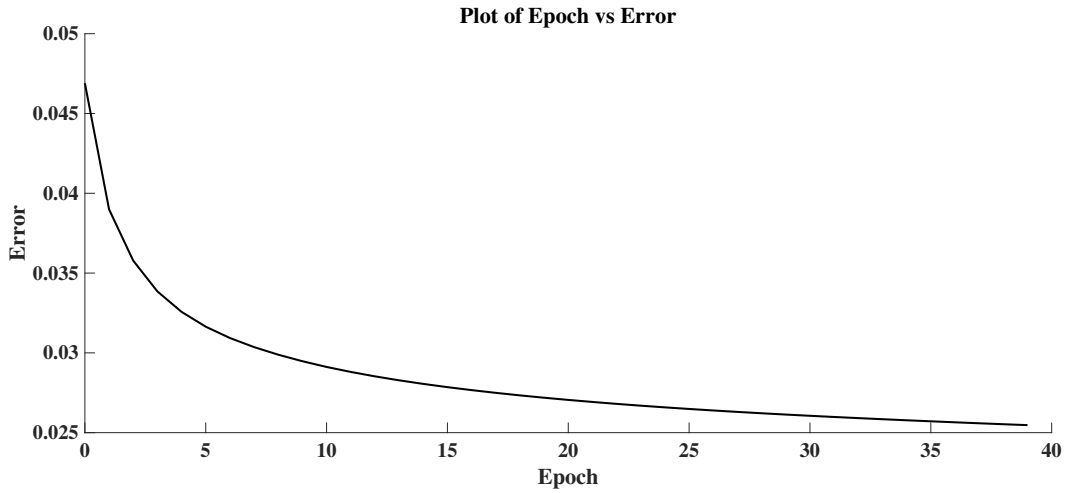

Figure 4.26: Error Vs Epoch for selected four documents using MCANN (40 epochs). Ninety percent data was used as training data and ten percent as test data.

The error obtained for above experiment with MCANN was 2.5471e-02 (in 40 iterations) as shown in figure 4.26. In this experiment ninety percent data was used for training and ten percent was used for testing.

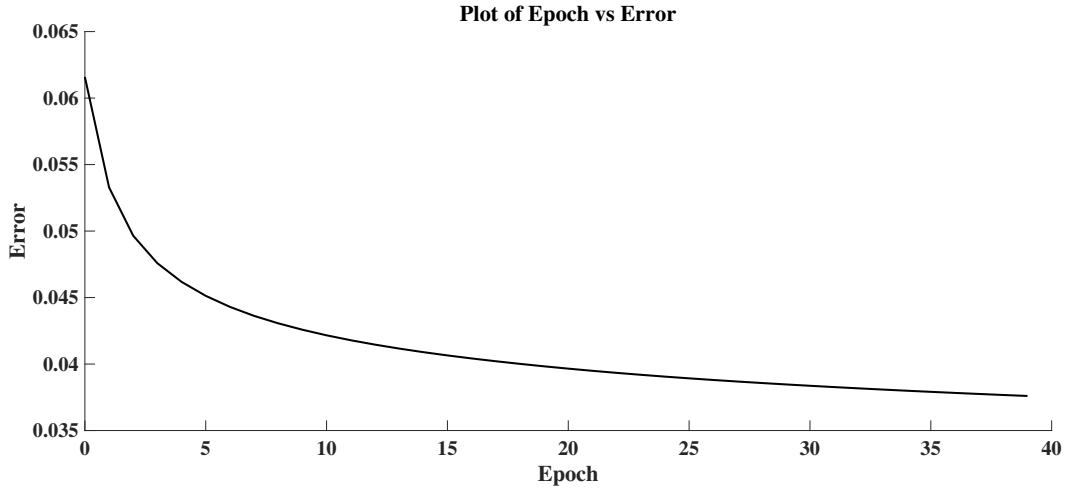

Figure 4.27: Error Vs Epoch for selected four documents using MCANN (40 epochs). Eighty percent data was used as training data and twenty percent as test data.

The error obtained for above experiment with MCANN was 3.7599e-02 (in 40 iterations) as shown in figure 4.27. In this experiment eighty percent data was used for training and twenty percent was used for testing.

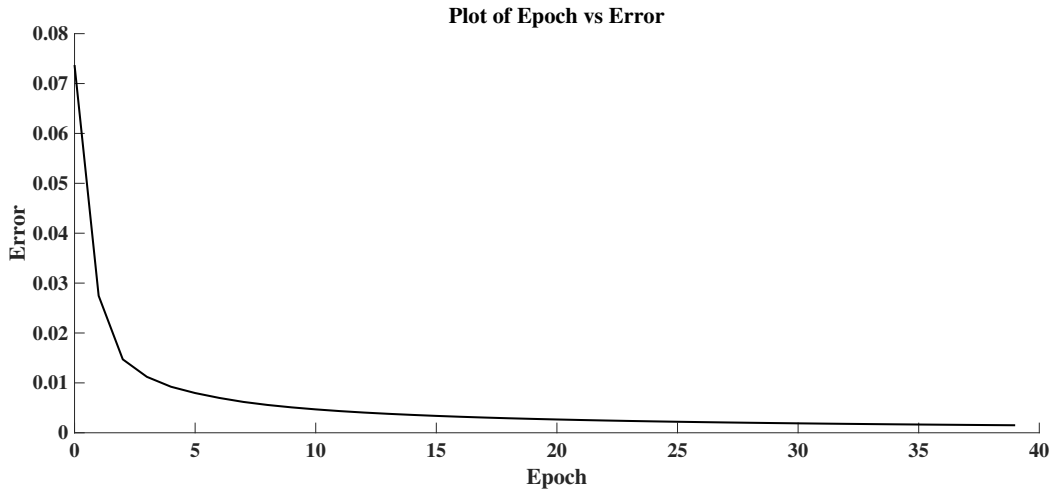

Figure 4.28: Error Vs Epoch for selected four documents using MCANN (40 epochs). Sixty percent data was used as training data and forty percent as test data.

The error obtained for above experiment with MCANN was 1.5065e-03 (in 40 iterations) as shown in figure 4.28. In this experiment sixty percent data was used for training and forty percent was used for testing.

Table 4.5: Result of Backpropagation on selected Nepali thesis

Dataset Used	Algorithms Applied	No. of Epoch	Error obtained on different experiments		
			5-fold Cross Validation	7-fold Cross Validation	10-fold Cross Validation
Selected Nepali Thesis	BP	40	23.743	24.496	24.137
Selected Nepali Thesis	BP	400	10.925	10.926	10.880

The above table 4.5 lists the result of Backpropagation algorithm on selected nepali thesis.

Table 4.6: Result of MCANN on selected Nepali thesis

Dataset Used	Algorithms Applied	Error obtained on different experiments		
		60% train & 40% test data	80% train & 20% test data	90% train & 10% test data
Selected Nepali Thesis	MCANN	1.5065e-03	3.7599e-02	2.5471e-02

The above table 4.6 lists the result of MCANN algorithm on selected nepali thesis.

4.5 Results of Experiments Carried Out with Selected Portion of Selected Nepali Thesis

For more rigorous analysis, Theory and Result section of four selected Nepali thesis are extracted and experiment using MCANN was carried out.

4.5.1 Results of Paragraph based Experiment carried out on Theory section of four documents

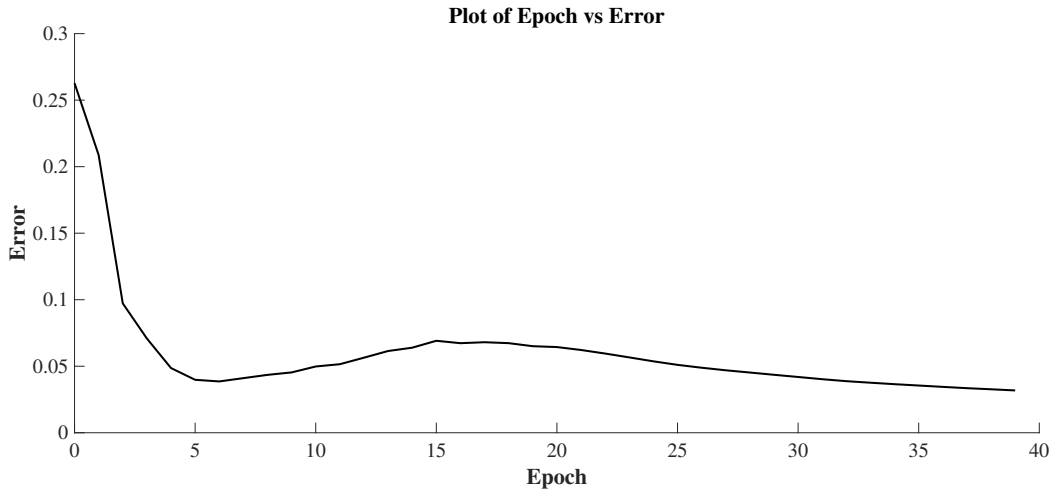

Figure 4.29: Error Vs Epoch for Theory section of documents using MCANN (40 epochs). Ninety percent data was used as training data and Ten percent as test data.

The error obtained for above experiment with MCANN was 3.1812e-02 (in 40 iterations) as shown in figure 4.29. In this experiment Ninety percent data was used for training and Ten percent was used for testing.

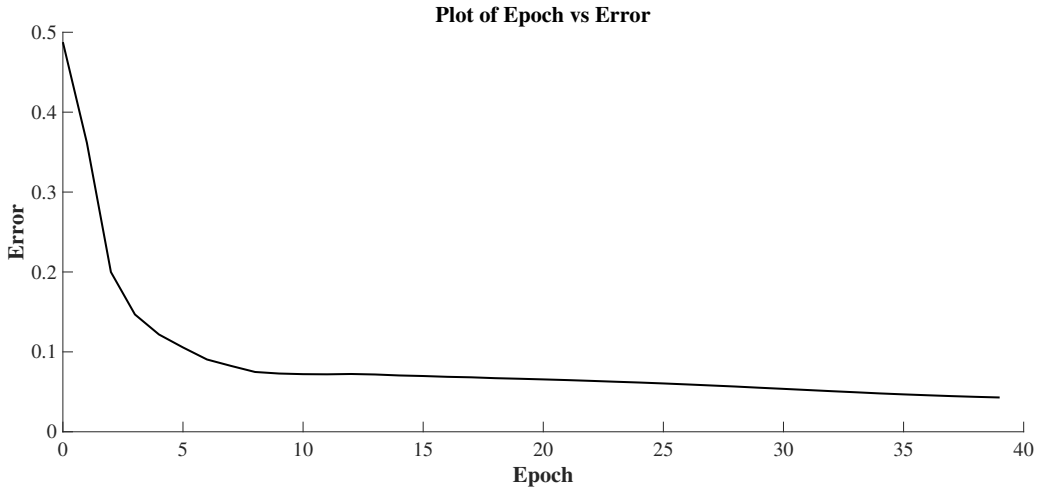

Figure 4.30: Error Vs Epoch for Theory section of documents using MCANN (40 epochs). Eighty percent data was used as training data and Twenty percent as test data.

The error obtained for above experiment with MCANN was 4.2914e-02 (in 40 iterations) as shown in figure 4.30. In this experiment Eighty percent data was used for training and Twenty percent was used for testing.

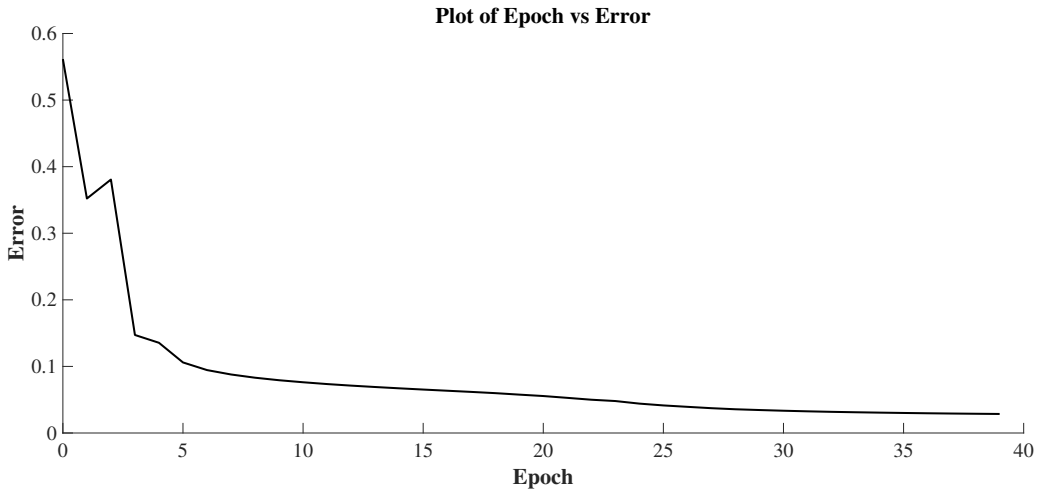

Figure 4.31: Error Vs Epoch for Theory section of documents using MCANN (40 epochs). Sixty percent data was used as training data and Forty percent as test data.

The error obtained for above experiment with MCANN was 2.8589e-02 (in 40 iterations) as shown in figure 4.31. In this experiment Sixty percent data was used for training and Forty percent was used for testing.

4.5.2 Results of Line based Experiment carried out on Theory section of four documents

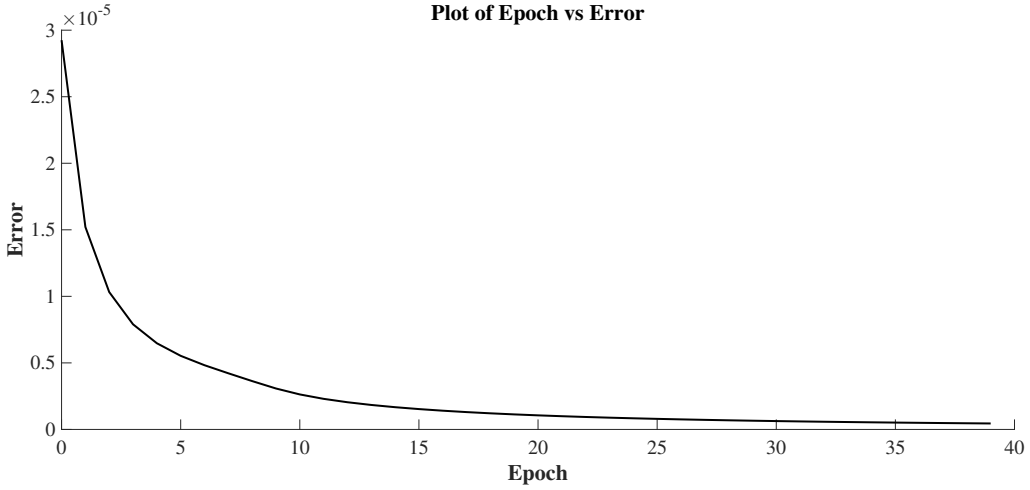

Figure 4.32: Error Vs Epoch for Theory section of documents using MCANN (40 epochs). Ninety percent data was used as training data and Ten percent as test data.

The error obtained for above experiment with MCANN was 4.4805e-07 (in 40 iterations) as shown in figure 4.32. In this experiment Ninety percent data was used for training and Ten percent was used for testing.

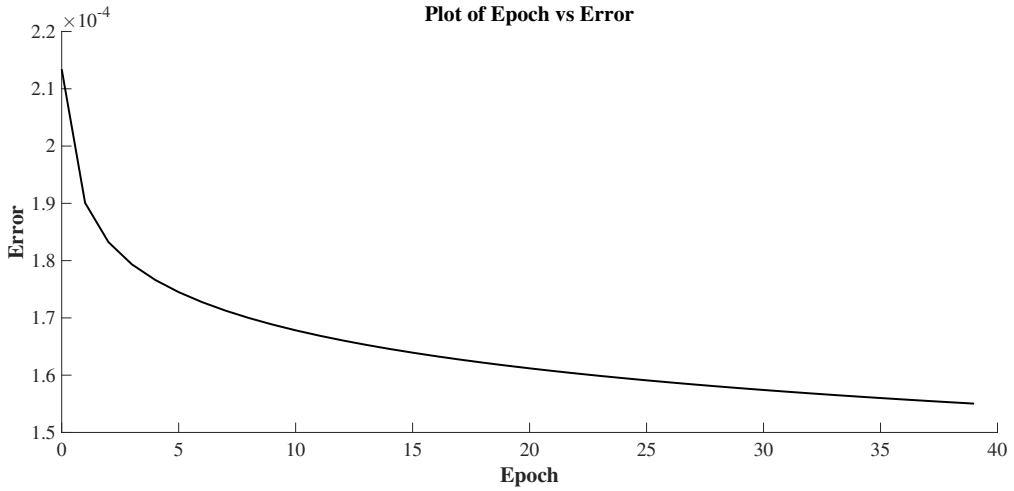

Figure 4.33: Error Vs Epoch for Theory section of documents using MCANN (40 epochs). Eighty percent data was used as training data and Twenty percent as test data.

The error obtained for above experiment with MCANN was 1.5503e-04 (in 40 iterations) as shown in figure 4.33. In this experiment Eighty percent data was used for training and Twenty percent was used for testing.

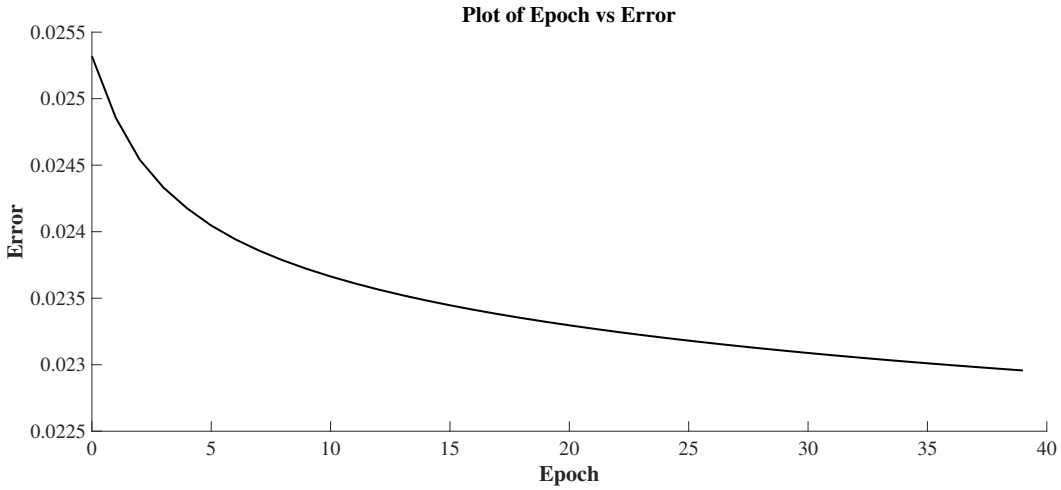

Figure 4.34: Error Vs Epoch for Theory section of documents using MCANN (40 epochs). Sixty percent data was used as training data and Forty percent as test data.

The error obtained for above experiment with MCANN was 2.2957e-02 (in 40 iterations) as shown in figure 4.34. In this experiment Sixty percent data was used for training and Forty percent was used for testing.

4.5.3 Results of Paragraph based Experiment carried out on Result section of four documents

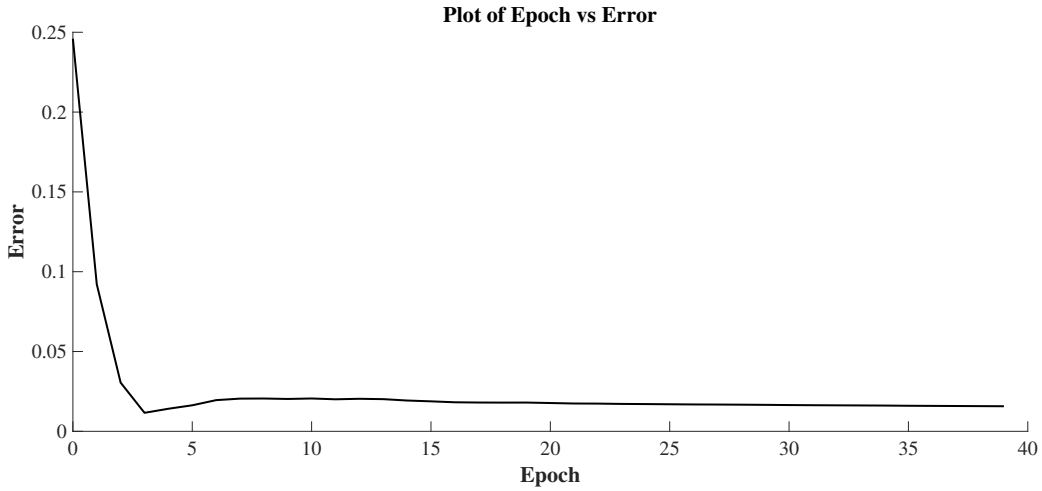

Figure 4.35: Error Vs Epoch for Result section of documents using MCANN (40 epochs). Ninety percent data was used as training data and Ten percent as test data.

The error obtained for above experiment with MCANN was 1.5713e-02 (in 40 iterations) as shown in figure 4.35. In this experiment Ninety percent data was used for training and Ten percent was used for testing.

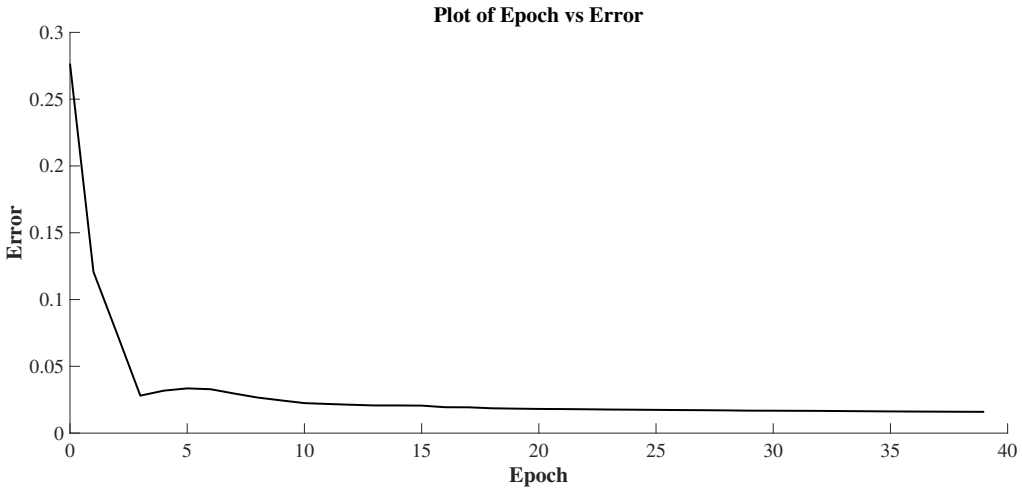

Figure 4.36: Error Vs Epoch for Result section of documents using MCANN (40 epochs). Eighty percent data was used as training data and Twenty percent as test data.

The error obtained for above experiment with MCANN was 1.5928e-02 (in 40 iterations) as shown in figure 4.36. In this experiment Eighty percent data was used for training and Twenty percent was used for testing.

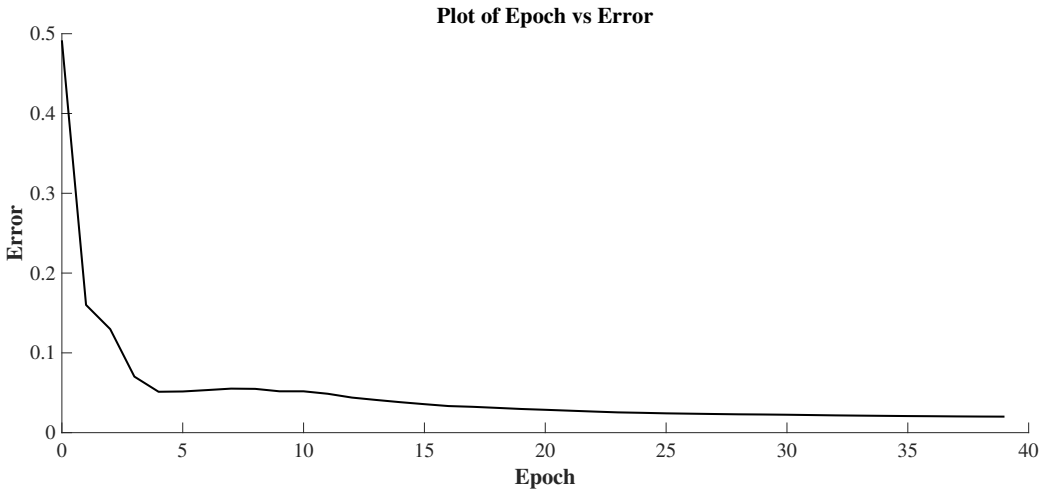

Figure 4.37: Error Vs Epoch for Result section of documents using MCANN (40 epochs). Sixty percent data was used as training data and Forty percent as test data.

The error obtained for above experiment with MCANN was 2.0076e-02 (in 40 iterations) as shown in figure 4.37. In this experiment Sixty percent data was used for training and Forty percent was used for testing.

4.6 Results Summary

All the results obtained are summarized in the table as shown.

Table 4.7: Comparison of result of MCANN and BP model on Bam data

Dataset Used	Algorithms Applied	Analysis Approach	No. of Epoch	Error obtained on different experiments		
				60% train & 40% test data	80% train & 20% test data	90% train & 10% test data
Bam data	MCANN	Paragraph based	40	3.0948e-04	2.1130e-04	8.1471e-05
Bam data	MCANN	Line based	40	3.4599e-05	1.3106e-04	3.2539e-04
Bam data	BP	Paragraph based	40	5-fold cross validation 335.854	7-fold cross validation 350.929	10-fold cross validation 360.370
Bam data	BP	Line based	40	1699.706	1775.384	1843.961

Table 4.7 lists the result of MCANN and BP on Bam data. Table 4.8 lists the result of MCANN and BP on all eleven Nepali thesis. Table 4.9 lists the result of MCANN and BP on selected four Nepali thesis documents. The above table 4.10 lists the result of MCANN algorithm on different portion of selected four Nepali thesis.

Table 4.8: Comparison of result of MCANN and BP model on all eleven Nepali thesis

Dataset Used	Algorithms Applied	Analysis Approach	No. of Epoch	Error obtained on different experiments		
				60% train & 40% test data	80% train & 20% test data	90% train & 10% test data
Nepali Thesis (11 documents)	MCANN	Paragraph based	40	6.1455e-03	3.0096e-03	2.4219e-03
				5-fold cross validation	7-fold cross validation	10-fold cross validation
Nepali Thesis (11 documents)	BP	Paragraph based	40	6.163	5.111	5.952
Nepali Thesis (11 documents)	BP	Line based	400	0.385	0.231	0.131

Table 4.9: Comparison of result of MCANN and BP model on Selected four Nepali thesis

Dataset Used	Algorithms Applied	Analysis Approach	No. of Epoch	Error obtained on different experiments		
				60% train & 40% test data	80% train & 20% test data	90% train & 10% test data
Selected Nepali Thesis (4 documents)	MCANN	Paragraph based	40	1.5065e-03	3.7599e-02	2.5471e-02
				5-fold cross validation	7-fold cross validation	10-fold cross validation
Selected Nepali Thesis (4 documents)	BP	Paragraph based	40	23.743	23.743	24.137
Selected Nepali Thesis (4 documents)	BP	Line based	400	10.925	10.926	10.880

Table 4.10: Comparison of result of MCANN on different portion of selected four Nepali thesis

Dataset Used (Selected Nepali thesis)	Algorithms Applied	Analysis Approach	No. of Epochs	Error obtained on different experiments		
				60% train & 40% test data	80% train & 20% test data	90% train & 10% test data
Theory Section	MCANN	Paragraph based	40	2.8589e-02	4.2914e-02	3.1812e-02
Theory Section	MCANN	Line based	40	2.2957e-02	1.5503e-04	4.4805e-07
Result Section	MCANN	Paragraph based	40	2.0076e-02	1.5928e-02	1.5713e-02

Chapter 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

Plagiarism is the most common problem found in the current world of electronics where data are easily available. So, mechanism for detecting and controlling it is the must. For the task, several methodologies are available for several languages but they are not enough for Nepali language based literature. Also, randomized method for detecting plagiarism is not encountered, which begins the motivation for the research and hence the topic "Comparative Study of Back - Propagation and Monte - Carlo Artificial Neural Network for Plagiarism Detection in Nepali Language". Nepali languages documents collected from different sources are passed in the framework for results. Obtained results are then analyzed for their accuracy. Accuracy of MCANN method seems satisfactory over BP method.

From the results obtained it is concluded that neural network trained with monte carlo method performs better than traditional backpropagation method. Thus, Monte Carlo based Artificial Neural Network is beneficial over general artificial neural network trained using backpropagation learning method for problems related to similarity detection.

5.2 Future Enhancement

This research was focused on extrinsic plagiarism detection of Nepali language based documents. It could be further researched for cross lingual plagiarism detection task. Similarly, performance could be increased by increasing more similarity measures as features. Better analysis could be carried out with datasets of different varieties collected from different fields. Also, effect of Evolutionary algorithms could be studied for detecting the plagiarism on Nepali language documents. Also, this research could be augmented for intrinsic plagiarism detection.

References

[1] S. B. Bam and T. B. Shahi, "Named entity recognition for nepali text using support vector machines," *Intelligent Information Management*, vol. 2014, 2014.

[2] S. Hariharan, "Automatic plagiarism detection using similarity analysis." *International Arab Journal of Information Technology*, vol. 9, no. 4, pp. 322–326, 2012.

[3] R. Lukashenko, V. Graudina, and J. Grundspenkis, "Computer-based plagiarism detection methods and tools: an overview," in *Proceedings of the 2007 international conference on Computer systems and technologies*. ACM, 2007, p. 40.

[4] D. Curran, "An evolutionary neural network approach to intrinsic plagiarism detection," in *Artificial Intelligence and Cognitive Science*. Springer, 2010, pp. 33–40.

[5] S. D. Salunkhe and S. Gawali, "A plagiarism detection mechanism using reinforcement learning," *International Journal*, vol. 1, no. 6, 2013.

[6] S. M. Alzahrani and N. Salim, "Plagiarism detection in arabic scripts using fuzzy information retrieval," in *Student Conference on Research and Development, Johor Bahru, Malaysia*, 2008.

[7] E. Stamatatos, "Plagiarism detection using stopword n-grams," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 12, pp. 2512–2527, 2011.

[8] J. F. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet, "Sequential monte carlo methods to train neural network models," *Neural computation*, vol. 12, no. 4, pp. 955–993, 2000.

[9] M. Y. M. Chong, "A study on plagiarism detection and plagiarism direction identification using natural language processing techniques," 2013.

[10] S. Sivanandam and S. Deepa, *Introduction to neural networks using Matlab 6.0*. Tata McGraw-Hill Education, 2006.

[11] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1593–1623, 2014.

[12] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso, "An Evaluation Framework for Plagiarism Detection," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*. Beijing, China: Association for Computational Linguistics, Aug. 2010.

[13] S. Marsland, *Machine learning: an algorithmic perspective*. CRC press, 2015.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, Massachusetts: London, England: A Bradford Book, The MIT Press, 2012.

[15] M. Potthast, A. Eiselt, A. Barrón-Cedeño, B. Stein, and P. Rosso, "Overview of the 3rd International Competition on Plagiarism Detection," in *Working Notes Papers of the CLEF 2011 Evaluation Labs*, V. Petras, P. Forner, and P. Clough, Eds., Sep. 2011. [Online]. Available: http://www.clef-initiative.eu/publication/working-notes