

**Data Leakage Prevention Framework Through Information Sensitivity Classification**

**A dissertation submitted for the Degree of Master of Information Security**

**M. S. Chinthaka Lakshman**

**University of Colombo School of Computing**

**2019**



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: M. S. Chinthaka Lakshman

Registration Number: 2016/MIS/018

Index Number: 16770187

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: Date:

This is to certify that this thesis is based on the work of Mr. M. S. Chinthaka Lakshman under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: Date:

Abstract

Data Leakage Prevention System is one of the core elements in Information Security Tools Framework among other utilities such as Intrusion Detection and Prevention Systems, Firewalls, Security Incident & Event Management Systems, Spam Filters etc. In today’s context, many security incidents occur by the insiders via intentional or unintentional information sharing with unauthorized personals or systems. Thus, a Data Leakage Prevention System plays a key role in securing information assets and it works on three principle domains, namely, Information Asset Discovery, Monitoring and Prevention.

‘Discovery’ stage should identify the available Information Assets within an organization while discovering the sensitivity levels associated with respective assets. Today, this step is either a manual process where information asset owner is responsible for assigning the classification label for the asset or an automated process where various classification mechanisms are applied on the assets. Automated Classification is not yet fully adopted in to the ‘Commercial Data Leakage Prevention Systems’ due to the unpredictable ‘Accuracy Levels’.

This experiment was done for identifying a better technique for classifying information assets of a Domain Specific Data Set with an increased accuracy level. Multi-Layer Perceptron Neural Network was identified as ~98% accurate in classification for the considered data set. ~97% and ~96% was the highest accuracy level observed for Random Forest and Convolution Neural Network techniques respectively. Even though the experiment was performed on another non-standard model which combines the Random Forest with Convolution Neural Network, 60% was the maximum accuracy level achieved. The proposed Multi-Layer Perceptron Neural Network technique achieved ~1% accuracy improvement over Random Forest while Random Forest was the well-accepted algorithm for a Data Set classification.

A realistic data set was prepared as part of this experiment where the Systems Integrator Industry was the target domain. Prepared data set comprised of Legal Documents, HR Documents, Data Sheets, Solution Documents, Agreements, Policy Documents and White Papers. Data Set was finally classified in to four different classes based on the industry acceptance. The different classes are based on sensitivity levels, namely, High Sensitive, Sensitive, Sensitive, Non-Sensitive and Open.

Acknowledgement

First, I am sincerely grateful to my supervisor, Dr. Kasun De Zoysa of University of Colombo, School of Computing for his immense support, motivation, knowledge and guidance given to me throughout my studies. He consistently allowed this research to be my own work and guided me in the right direction from the proposal stage itself until the thesis writing. It was always an open door for me whenever I needed his assistance.

This work could not have been completed without the support and encouragement of my loving family members. I take this opportunity to express my gratitude for my Parents and my beloved wife who actively encouraged me during my studies for twelve months consecutively. And my two loving kids always helped me by accepting the time I allocated for this work.

I would like to thank all my friends for their support and guidance. Also, I would like to thank all the staff members of University of Colombo, School of Computing for making a supportive and nice environment for this kind of work.

Contents

[**Chapter 1: Introduction to DLP** 1](#_Toc18319025)

[1.1 Motivation 1](#_Toc18319026)

[1.2 Content Discovery and Techniques for Analysis 4](#_Toc18319027)

[1.2.1 Rule-Based/Regular Expressions 4](#_Toc18319028)

[1.2.2 Database Fingerprinting 4](#_Toc18319029)

[1.2.3 Exact File Matching 5](#_Toc18319030)

[1.2.4 Partial Document Matching 5](#_Toc18319031)

[1.2.5 Statistical Analysis 6](#_Toc18319032)

[1.2.6 Conceptual/Lexicon 7](#_Toc18319033)

[1.2.7 Categories based Classification 7](#_Toc18319034)

[1.3 Accuracy of Content Discovery and Analysis 8](#_Toc18319035)

[1.4 Objectives 8](#_Toc18319036)

[1.5 Document Structure 9](#_Toc18319037)

[**Chapter 2: Background, Literature Review, Related Work** 10](#_Toc18319038)

[2.1 Importance of Feature Derivation and Reduction 10](#_Toc18319039)

[2.2 Data Set and Experiments 11](#_Toc18319040)

[2.3 Using Association Rule with Naïve Bayes Classifier 11](#_Toc18319041)

[2.4 Classification with Support Vector Machines 12](#_Toc18319042)

[2.5 SVM Over Artificial Neural Network (ANN) 13](#_Toc18319043)

[2.6 Document Classification for Focused Topics 14](#_Toc18319044)

[**Chapter 3: Design of the Solution** 16](#_Toc18319045)

[3.1 Input Data Identification and Preprocessing 20](#_Toc18319046)

[3.2 Classification Algorithms and Training 22](#_Toc18319047)

[3.2.1 Random Forest Classifier 23](#_Toc18319048)

[3.2.2 Artificial Neural Network: 27](#_Toc18319049)

[3.2.3 Convolution Neural Network 31](#_Toc18319050)

[3.3 Testing against a known dataset 33](#_Toc18319051)

[3.4 Performance Evaluation 33](#_Toc18319052)

[**Chapter 4: Implementation** 35](#_Toc18319053)

[4.1 Data Set used in analysis 35](#_Toc18319054)

[4.1.1 Data Set Identification 35](#_Toc18319055)

[4.1.2 Data set preparation 37](#_Toc18319056)

[4.1.3 Data Set Segregation 38](#_Toc18319057)

[4.2 Classification Model Implementation 39](#_Toc18319058)

[4.2.1 Random Forest 39](#_Toc18319059)

[4.2.2 Artificial Neural Network (ANN) 42](#_Toc18319060)

[4.2.3 Convolution Neural Network (CNN) 42](#_Toc18319061)

[4.3 Technologies Used 43](#_Toc18319062)

[**Chapter 5: Results and Evaluation** 44](#_Toc18319063)

[5.1 Random Forest Model 44](#_Toc18319064)

[5.2 MLP Neural Network 46](#_Toc18319065)

[5.3 CNN Model 47](#_Toc18319066)

[5.4 RF + MLP Model 48](#_Toc18319067)

[**Chapter 6: Conclusion and Future Work** 50](#_Toc18319068)

[**References** 51](#_Toc18319069)

[**Annexure I** 53](#_Toc18319070)

[Bag of Words 53](#_Toc18319071)

[**Annexure II** 57](#_Toc18319072)

[ANN Implementation - Pseudocode 57](#_Toc18319073)

List of Figures

[Figure 1. 1: DLP Point of Interests 2](#_Toc9380648)

[Figure 1. 2: Working Order of a Current DLP System 3](#_Toc9380649)

[Figure 3. 1: Proposed Classification Architecture 20](#_Toc9380734)

[Figure 3. 2: Steps Involved in (Systems Integrator) Data Preparation 22](#_Toc9380735)

[Figure 3. 3: Steps Involved in Classification 24](#_Toc9380736)

[Figure 3. 4: Graphical Representation of a Decision Tree 25](#_Toc9380737)

[Figure 3. 5: decision Tree created in Classification 27](#_Toc9380738)

[Figure 3. 6: Standard Representation of the Implemented Neural Network 30](#_Toc9380739)

[Figure 3. 7: Feature Selection from Random Forest Classification 31](#_Toc9380740)

[Figure 3. 8: ReLU Graphical Representation 33](#_Toc9380741)

[Figure 3. 9: Testing Against a Pre-Labeled Data Set 34](#_Toc9380742)

[Figure 4. 1: Flow of Data Preparation 37](#_Toc9380743)

[Figure 4. 2: Data Set Segregation 39](#_Toc9380744)

[Figure 4. 3: Prepared SI Data Set 40](#_Toc9380745)

[Figure 4. 4: Implementation of Convolution Neural Network 44](#_Toc9380746)

[Figure 5. 1: Representation of a Sample Random tree 45](#_Toc9381521)

[Figure 5. 2: Representation of 20 Important Words in Random Forest 46](#_Toc9381522)

[Figure 5. 3: Accuracy at 70% Train:Test Ratio 46](#_Toc9381523)

[Figure 5. 4: Loss and Accuracy for Training and Validation 47](#_Toc9381524)

[Figure 5. 5: Accuracy at 70% (left) & 60% (right) Train:Test 48](#_Toc9381525)

[Figure 5. 7: Training and Validation Loss at 70% Train:Test Ratio 48](#_Toc9381526)

[Figure 5. 8: Accuracy at 70% Train:Test Ratio 49](#_Toc9381527)

[Figure 6. 1: Accuracy Levels of Four Different Techniques against various Train:Test Ratios 50](#_Toc9380754)

[Figure 6. 2: Accuracy Variation against various Train:Test Ratios 51](#_Toc9380755)

[Figure 7. 1: Bag Of Words Used in Document Interpretation 53](#_Toc9380758)

[Figure 7. 2: Different Vocabularies for Different Document Classes 55](#_Toc9380759)

List of Tables

[Table 2. 1: Classification Accuracy of Standard Algorithms 15](#_Toc9381592)

[Table 3. 1: Summary of Various Text Classification Techniques 17](#_Toc9381618)

[Table 3. 2: Organization of Neural Networks Based on Their Functional Characteristics 28](#_Toc9381619)

[Table 4. 1: Data Formats and Associated Security Labels in SI Data Set 36](#_Toc9381622)

[Table 4. 2: Excluded Document Formats in SI Data Set 36](#_Toc9381623)

[Table 5. 1: Accuracy Figures at Multiple Iterations of Random Forest Implementation 45](#_Toc9381626)

[Table 5. 2: Accuracy Figures at Multiple Iterations of MLP Implementation 47](#_Toc9381627)

[Table 5. 3: Accuracy Figures at Multiple Iterations of 2-Dimensional CNN Implementation 48](#_Toc9381628)

[Table 5. 4: Accuracy Figures at Multiple Iterations of Random Forest+MLP Implementation 49](#_Toc9381629)

[Table 6. 1: Summary of Observations for Various Implementations 50](#_Toc9381631)

[Table 7. 1: Term Frequency Matrix for Example Documents 54](#_Toc9381633)

[Table 7. 2: Term frequency matrix of documents for mathematical and sports BoW vocabularies 55](#_Toc9381634)

List of Abbreviations

**HR** Human Resource

**DLP** Data Leakage Prevention

**SVM** Support Vector Machines

**ANN** Artificial Neural Network

**CNN** Convolution Neural Network

**MLP** Multi-Layer Perceptron

**RF** Random Forest

**SI** Systems Integration

**PCI** Payment Card Industry

**USB** Universal Serial Bus

**CAD** Computer Aided Designing

**PCA** Principal Component Analysis

**KNN** K-Nearest Neighbor

**RBF** Radial Basis Functions

**NN** Neural Network

**SGML** Standard Generalized Markup Language

**XML** Extensible Markup Language

**KSS** Knowledge System Server

**LDC** Linguistic Data Consortium

**SVD** Singular Value Decomposition

**ART** Adaptive Resonance Theory

**FLN** Fast Learning Network

**CMAC** Cerebellar Model Articulation Controller

**RFPM** Random Forest Probability Machine

**PNN** Probabilistic Neural Network

**RMSE** Root Mean Square Error

**HTML** Hyper Text Markup Language

**ML** Machine Learning

**IDE** Integrated Development Environment

# **Chapter 1: Introduction to DLP**

## 1.1 Motivation

In todays’ context, Data is more accessible and transferable between parties than ever before, and the majority of data is sensitive at various levels. Certain data is confidential because it is part of the organization and those are not meant to be available to general public. Certain other data is sensitive because of national policies such as General Data Protection Regulation by European Union, corporate requirements and international regulations. However, the sensitivity and the value of the considered data asset has a direct relationship to its context and content.

[Intellectual](https://whatis.techtarget.com/definition/intellectual-property-IP) Property, National Identity Card Numbers of customers and employees, [personally identifiable information](https://searchfinancialsecurity.techtarget.com/definition/personally-identifiable-information), [personal health information](https://searchhealthit.techtarget.com/definition/personal-health-information), financial data such as bank account numbers and [Payment Card Information](https://searchcompliance.techtarget.com/definition/PCI-compliance)(PCI) can be given as examples of sensitive information. As per Symantec estimation about the data sensitivity in a todays’ organizational context [1]

* One in every 400 emails will contain confidential information
* One in every 50 network files will contain confidential data
* 80% of times, a company looses confidential data if a laptop gets lost
* A USB drive contains confidential information at 50% of times
* The increase in customer turnover is 11% in case a company goes through data breach incident

Leakage of such sensitive information could be embarrassing or worse, cost the organization’s industrial edge or loss of accounts. Hence an organization must take necessary precautions and actions for securing such sensitive data. Among many other mechanisms to maintain the confidentiality of data, one basic methodology is to deploy Data Loss/Leakage Prevention Systems (DLP) [2]. Data Leakage Prevention Systems are all about

1. Identify which data is sensitive
2. How the unintentional data leakages can be detected and prevented.

DLP is generally defined as, “*Products that, based on central policies, identify, monitor, and protect data at rest, in motion, and in use, through deep content analysis*.” [3]

*Data loss prevention (DLP) is the practice of detecting and preventing confidential data from being “leaked” out of an organization’s boundaries for unauthorized use.* [4]

As depicted in Figure 1.1, a DLP system can protect the data from unwanted and un-intentional sharing in three different stages of the life cycle of data, namely ‘Data at rest’, ‘Data in motion’ and ‘Data in use’.

Laptops

Workstations

Firewall

Databases

App Servers

Data in Motion

Data in Use

Data at Rest

Storage

Internet

Figure 1. 1: DLP Point of Interests

* **Data at Rest:** The stage of ‘Data at Rest’ includes scanning of all storages and other content repositories with the purpose of identifying where sensitive content is located and more over what those assets are. This step is called ***‘content discovery’***. As an example, a security administrator should be able to use a DLP system to scan all the server contents and identify specific set of documents with payment card information. In case if this server is not authorized for keeping this type of data set in its possession, then the file can be encrypted or removed, or a warning can be sent to the original owner of the file.
* **Data in Motion:** This stage discusses about passive sniffing of motion traffic on the network with the purpose of identifying contents that will be sent across specific communications channels. More often this will happen through a implementation of proxy inside the network.
* **Data in Use:** This stage discussed about monitoring the data as and when the user operationally interacts with documents. Ideally a security administrator can detect when an intruder attempts to transfer a sensitive document over unauthorized channel and block that action as a security measure.

DLP systems’ main building blocks include Discovery, Monitor and Prevent. Under Discovery, the system will search through the network and storage locations for the information assets which should be protected. The discovered assets will be automatically or manually classified and labeled accordingly for future Monitoring. Monitoring step will enforce the policies and check for any policy violation that may occur. The final step is Prevent where it will either take actions for preventing the information disclosure or alert the authorized personal on the security event.

My research area is working on the ‘Discovery’ step and more precisely on an improved way of ‘Information Assets Classification’

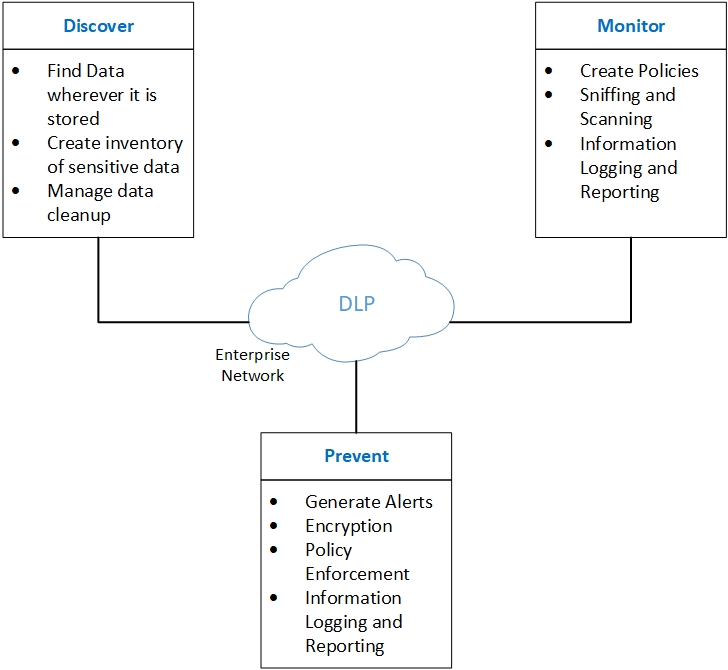


Figure 1. 2: Working Order of a Current DLP System

## 1.2 Content Discovery and Techniques for Analysis

Once the content accessing and discovery is completed, there are common techniques which can be used to ﬁnd importance of the assets and policy violations of those, each with its own strengths and weaknesses which are discussed below.

### 1.2.1 Rule-Based/Regular Expressions

Rule-based or Regular expressions based methodology is one of the popular analysis techniques that is used in both DLP products and such tools which are having DLP features. This analyzes the content for speciﬁc predefined rules, as example, 16-digit codes that meet electronic payment checksum requirements, any invoice or billing code, other textual analyses etc. Most available DLP systems have taken extra steps to improve these basic regular expressions with some additional rules so that the hidden information also can be uncovered. As an example, they can uncover a name in proximity to an address closer to a credit card number can be found.

This method could be used as a ﬁrst-pass ﬁlter and will work towards detecting easily identiﬁed pieces of structured data. As an example, National Identity Card Numbers (NIC), electronics payment card information and records of personal healthcare.

**Strengths:** This method is straight forward, can be easily configured and also Processes quickly. Most of available DLP systems come with initial common set of rules. This well understood technology is easy to easy to incorporate into a variety of other products.

**Weaknesses:** This methodology has probability of giving high false positive rates. This means this offers a little or no protection for unstructured contents.

1.2.2 Database Fingerprinting

This is also introduced as ‘Exact Data Matching’. The way this technique operates is, first it takes a database dump or live data stream of the considered database. Secondly, it will look for exact matches. As an example, one DLP system could set a policy to check only for electronic payment card numbers in a customer base. This could ignore the employees who buy online. More advanced DLP tools will search for various combinations of information, such as the combination of ﬁrst name, with last name, with NIC Number or with payment Card information, which could trigger an alert to the security administrator or data owner.

This method best works for unstructured data formats.

**Strengths:** This technique generate low false positives, or it would be nearly zero most of the times. Allows to better protect sensitive information while other similar types of information used by the employees is ignored.

**Weaknesses:** Having a live connection can affect the performance of the database. On the other hand, scheduled dumps would not contain up to date transaction data and it will contain data only up to the last extract. And finally the larger databases could affect the DLP system performance.

1.2.3 Exact File Matching

This technique calculates a hash of a particular ﬁle and then monitor it for any exact matching is available between the fingerprint and new file. This could considered to be a ‘contextual analysis technique’ as the ﬁle internal contents will never be read nor analyzed.

This method best fits for media ﬁles as textual analysis is not possible all the times.

**Strengths:** This method will work on any ﬁle type while generating low false positives rate. But for ensuring such low rate, a large enough hash values should be available.

**Weaknesses:** If the considered file content is frequently getting edited, this technique will not fit. Media files, ofﬁce documents and financial documents are examples of content being frequently edited.

### 1.2.4 Partial Document Matching

This technique will search for a complete or partial match on content which need to be protected. Hence, a DLP system could build a policy to protect a document with sensitive information, and the system will search for complete text of the document. Or this search will go in to granular levels such as number of sentences. Example; this technique may take a business plan for a new product in to consideration and the DLP system would generate alert if an employee extracted and pasted a single paragraph into an E-Mail or Instant Message. More importantly, most solutions are based on cyclical hashing technique, where it will first take the hash of a portion of the content, and then offset a predetermined number of characters before creating another hash. This partial hashing is continued until the document is completely passed through as a series of overlapping hash values. One entire content including outbound content is gone through this hash technique, the hash values compared for matches. In addition to cyclic hashing values, these products may use more advanced analysis techniques on top of.

This method is suitable for protecting sensitive documents or similar text contents. Examples could be given as CAD ﬁles with text labels, Source Codes and any other content in unstructured format which are identified as sensitive.

**Strengths:** Capability of unstructured data protection and will generate low false positives. Since this technique doesn't depend on complete matching of large documents, can ﬁnd policy violations even on partial basis.

**Weaknesses:** Common phrases presented in a protected document may trigger false positives. There could be performance limitation arising based on the total size of the content being analyzed. Exact document should be known in advanced.

### 1.2.5 Statistical Analysis

Statistical techniques and Machine Learning algorithms are used to analyze the content and then to ﬁnd policy violations in content which resembles the protected content. This technique comprises of number of statistical techniques which may defer from each other based on effectiveness and the way of implementation.

This technique is suitable for unstructured contents where deterministic techniques such as partial document matching will fail at. An example could be, a repository of Engineering Designs and Plans can be impractical to load for partial document matching due to its high volume and high volatility.

**Strengths:** This technique may work with more vague contents where it’s difficult to isolate exact documents for matching. Further, this technique can enforce policies such as "create an alert on anything outbound which resembles the documents in a particular folder".

**Weaknesses:** This technique requires a large corpus of source content in order to achieve a better result. Further this will generate more false positives and false negatives.

### 1.2.6 Conceptual/Lexicon

Conceptual/Lexicon technique uses a rules combination, dictionaries and other analyses to protect vague content. The requirement here is to protect the content which resembles a particular "idea". As an example, a specific policy that alerts on trafﬁc which resembles insider trading, which may be using word counts, key phrases or positions to ﬁnd violations. Running a private business from office account could be another example of Conceptual/Lexicon.

For the completely unstructured ideas that doesn’t adopt for any simple categorization may be a best use case for Conceptual/Lexicon technique.

**Strengths:** In general, Conceptual analysis is capable of ﬁnding out loosely deﬁned policy violations.

**Weaknesses:** This technique can generate false positives and false negatives because of the loose nature of the rules. Another weakness is that, the DLP vendor must make effort on building and defining a comprehensive set of rules for operation and that could add cost to the product.

### 1.2.7 Categories based Classification

This defines pre-built categories with rules and dictionaries for common types of data which could be identified as sensitive. As an example, credit card numbers and PCI protection data.

This technique is successful for contents that neatly ﬁts a provided category. Contents related to privacy or industry-speciﬁc guidelines is easy to describe here

**Strengths:** Configuration is extremely simple and signiﬁcant policy generation time can be saved. More advanced, enterprise speciﬁc policies can be formed by the Category Policies. Organizations can fulfill their data protection needs by using categories.

**Weaknesses:** This technique is good only for easily categorized rules.

In today’s context, above discussed techniques form the fundamentals for most of the DLP systems on the market. One product may use one or few of above discussed techniques though there can be signiﬁcant differences between implementations. Some products are chaining various techniques in a way those will build complex policies so that comprehensive protection is achieved.

## 1.3 Accuracy of Content Discovery and Analysis

As per the observation of an informal survey of major DLP vendors, content discovery is about 50-60% of a complete DLP project and content discovery will take first 12 months of the implementation. [5] This express the importance of the content discovery and analysis. A good Content Discovery tool will understand the file context, not only the content.

Accuracy rating achievable by Content Analysis methodology has to be closed to 100% always. It should reduce the number of false positives where such falsely alarmed assets will be monitored unnecessarily and negatively affect the productivity of the organization. On the other hand, it should reduce false negatives where such unattended assets could be a more sensitive asset to the organization which should be protected by all means.

Industry today is working on multiple mechanisms to increase the content analysis accuracy. These approaches are found and discussed under the search topic of “Information Assets Classification”.

## 1.4 Objectives

The Objective of this research is to develop an information asset classification model which gives higher accuracy level in classification. Instead of currently utilized techniques in DLP context, my approach will be on Machine Learning and Artificial Neural Network based information asset classification model. Furthermore, a special focus was given in for creating a workable model which best suits for Systems Integration Industry information assets in Sri Lanka. This is because, even though there have been many research works done around document classification, the accuracies and performance of those techniques depend heavily on the type of data under consideration. The same model applied on an entirely different dataset would not give the same accuracy or the performance.

Result of my work will make sure the derived workable model will fit for any typical Systems Integrator’s data set who operates in Technology Domain. Further, the obtained accuracy ratings will be discussed in detail for understanding the work and more importantly for anyone to study and continue the work for a much better accuracy and performance.

## 1.5 Document Structure

The content of this thesis is organized as follows:

Chapter 2 consists of a Literature Survey on existing Data Classification Techniques, Experiments and Associated Results. Also, this chapter discusses in detail on the background information such as Data set used, Observations and Conclusions.

Chapter 3 of this thesis discusses about the Design of this work. More details can be found here on the approach for the solution with four main techniques; Multi-Layer Perceptron Neural Network, Random Forest, Convolution Neural Network and Combined Technique separately. Expectations of each technique is also discussed. The industry specific data set preparation is also elaborated here.

Chapter 4 is provided with the implementation details of individual approaches discussed in previous Chapter. Pseudocodes and the exact parameters used will be presented in this Chapter.

Next Chapter (Chapter 5) is dedicated for discussing the results obtained for above techniques. Tabular formats, Heat Maps and Graphs will present the result in more understandable way to the reader.

Final Chapter of this document will summarize the observations and results while briefing out future possible enhancements for the work.

# **Chapter 2: Background, Literature Review, Related Work**

Previous researchers have shown a number of dependable techniques for document classification. During this research, the most recently established efficient approaches were considered that was found as a result of the search. This chapter will discuss on these findings while focusing more attention on below two aspects;

* A study done on some research papers which gives a view on different techniques to classify text will
* Important Factors to consider while Document Classifying

## 2.1 Importance of Feature Derivation and Reduction

In most of the previous researches, one of the common design steps is the feature derivation and feature set reduction of documents. In the work [6], Rudolf Hanka and Karel Fuka discusses on how Feature set reduction for document classification becomes important and how it can improve the accuracy of the model in consideration.

Feature reduction is performed in either of below two ways

* Feature selection: A subset of original features is retained while the rest of the features are discarded in this approach. Then the classification model is built using previously selected features. Determining a subset of ‘d’ features from the set of ‘m’, is the aim of the feature selection method.

* Feature extraction: In this technique, the original vector space is transformed to form a new feature vector space and its done with some special characteristics. Reduction is then performed on the newly derived vector space. Compared to the technique of Feature selection, this method will use all the features available. A linear or non- linear transformation is used to transform the original features into a smaller set of transformed features. This comprises of original human understandable features even though not meaningful to humans.

An optimization of some criterion function ‘J’ is required for both above approaches, which is usually a measure of distance or dissimilarity between distributions.

2.2 Data Set and Experiments:

The researchers here have used ‘Reuters-21578’ dataset for the purpose of training and testing of the classifier. Chi-squared statistic and PCA are some of the feature reduction techniques employed by the researchers. They originally started with 3822 original terms at the beginning. The authors used χ2 statistical data for feature selection where it gave 81% of approximate accuracy. The researchers used PCA to test the feature extraction. When PCA is applied to features obtained through χ 2 statistic, it gave an accuracy of 86%. In the same way, when PCA was applied over the complete feature set, it provided a 95% of accuracy.

As a conclusion of this experiment, all feature set reduction algorithms perform better compared to no feature reduction. Furthermore, selection of an appropriate feature extraction algorithm may perform better compared to the feature selection algorithm.

## 2.3 Using Association Rule with Naïve Bayes Classifier

Research on “Document Classification Using the Concept of Association Rule of Data Mining” has been done while using Naive Bayes Classifier for classifying the text. That research has shown the dependability of the Naïve Bayes Classifier with Associated Rules [7].

A set of example documents is given in the proposed system. They have preprocessed the text documents by removing stop words at the beginning. Then from each document, the frequently occurring words set has been extracted and each document has been treated as a transaction. Then the frequently occurring word sets has been viewed as a set of items in that transaction. As the next step, association mining method has been applied in order for discovering the sets of association words in that documents. These set of associated words will then act as features. Finally, new documents will be classified using Naïve Bayes approach using previously derived set of features.

But in some cases, the accuracy may fail because this method may ignore the negative calculations for a specific class. As an example, in order to classify a text, this method will calculate the probability of different classes with the probability values of the matched set and in the mean time by ignoring the unmatched sets. Result of that is, if test set matches with a set of rules and if it has weak probability to the actual class, then it might cause a wrong classification.

In this research, the number of documents in the training data set plays a vital role in generating the word sets which will be used to determine the class of a new document. Because, the task of new document classification completely depends on the associated word sets generated from training documents. It reduces the possibility of failure to classify a new document correctly, if there are higher number of word sets in training documents.

## 2.4 Classification with Support Vector Machines

**Introduction:** In this paper [8] the author Thorsten Joachims identified and discussed the benefits of using Support Vector Machines (SVMs) for the purpose of text categorization.

**Feature Representation:** As part of the preprocessing before creating the feature vectors, they have performed stemming. In order to generate feature vectors, the authors have used word counts. Therefore, each of the considered document was represented as an integer vector while each integer was representing how many times a corresponding word occurred inside that document. For avoiding large sized feature vectors, the researchers considered only those words which took place more than three times inside document as features. In making these feature vectors, the authors also eliminated stop words.

Above discussed representation scheme leads to very high-dimensional feature spaces which contains about 10000 dimensions or more. Furthermore, for the purpose of reducing the number of features and overfitting, Information gain criterion was used. Therefore, final subset of features was formed based on the Information gain.

**Data Set and Experiments:** Thorsten Joachims used two data sets for this model. ModApte split of the Reuters-21578 dataset was the first data set used by the author. This was compiled by David Lewis and the dataset contained 9603 training documents and 3299 test documents. There were 135 categories and out of that only 90 were used in the experiment. Reason was that only those 90 categories had at least one training and test sample.

As the second data set for model creation, the Ohsumed corpus was used where William Herse compiled. Here, the researcher used 10000 documents for training purposes. And different set of 10000 documents were used for testing purposes. There are 50000 documents in total. There are 23 diseases categories in total and the final goal of classification was to assign each document to one of those categories.

In this work, the researcher compares the performance of Rocchio, C4.5, SVMs with Naïve Bayes, and KNN for text categorization. Polynomial and RBF kernels were used for SVM. As a measure of performance, the Precision/Recall Breakeven Point is used. In the same time, for obtaining a single value of performance for all classification tasks, micro-averaging was performed. The author here also makes sure that the results are not biased towards any of above methods. Hence, the researcher implemented all the four methods with different number of selected features or all features and it was always based on the Information gain.

**Conclusion:** Conclusion of this work is that, KNN performed the best on Reuters data set compared to the other conventional methods. In the same time, SVM technique also achieved a better classification results with a good margin when its compared with other conventional methods. Also the SVM technique was proven to perform better in high dimensional space and therefore, that technique did not mandate the requirement feature selection such as other methods. And also, the author concludes that SVMs are observed to be robust and performed well in almost all the experiments.

It was concluded that the similar results were obtained for the Ohsumed collection data set. One of the facts that the results demonstrated was, k-NN performed the best among other conventional methods while SVM outperformed almost all other classifiers.

## 2.5 SVM Over Artificial Neural Network (ANN)

There is another study done by C. Watters, A. Basu, and M. Shepherd [9] where they compare support vector machine with an artificial neural network. The purpose of this work is to classify the content of news items.

Data Set: For their comparative study they used the Reuters News data set where that dataset contained a collection of 21,578 news items. These news items are divided across 118 categories.

Data Preprocessing:

The researchers first converted the SGML documents into XML documents using a separate tool. Then the documents which belonged to multiple categories or no category was removed from the set of XML documents. As the next step, they removed the categories which are having less than 15 documents left in it and this elimination process derived 63 categories which contains 11,327 documents finally.

As the next step, an extensive vocabulary was generated which comprises of 102,283 terms and KSS (Knowledge System Server) was used for performing this task. The authors used two different IQ values in order to reduce the complexity and limit the size of the vocabulary. KSS with 87 as the IQ value created a vocabulary of 62,106 terms and 57 as IQ value created a vocabulary of 78,165 terms. These terms were further reduced to 33,191 with the removal of terms and abbreviations that are not understandable by the KSS.

Experiment was performed to test both the classifiers by choosing 600 documents from the pool randomly. This random selection of documents resulted in leaving with a set of documents that are having too few or no documents from a particular category in many cases. Hence the researchers also performed testing for only those categories that had more than ten documents within the 600 random documents.

After performing the experiments, the researchers concluded that SVM performed better compared to Artificial Neural Network. For a set of documents with fewer categories and with shorter documents, they recommended SVM over ANN as SVM is less computationally expensive.

## 2.6 Document Classification for Focused Topics

Jay Chen, Russell Power, Trishank Karthik and Lakshminarayanan Subramanian, propose a combination of classification algorithms and feature extraction for the purpose of document classification in their paper [10]. Researchers propose an algorithm for simple feature extraction for domain specific dataset (domain specific data set). This study also yields that if the algorithm is coupled with conventional classifiers, a high classification accuracy could be achieved.

Their algorithm for features extraction comprised of a combination of two potentially opposing and different metrics. The purpose of having different metrics was to extract textual features for a considered topic, namely i) popularity and ii) rarity.

The set of words’ popularity could be demonstrated as per the level of popularity of an specific word for given category of documents. A list of more frequently occurring words in the document could be determined by this metric. Also, this list becomes closely related to the considered topic.

The metric of ‘rarity’ could capture the list of words which are less frequently used in relation to a considered topic. In order to learn the occurrence frequency of any n-gram on Web, the researchers leveraged the Linguistic Data Consortium (LDC) data set and this helps to measure rarity of any given term. Even though they used a slightly old LDC data set, there was an important observation that the rarity of terms in relation to any category is preserved and more importantly the specific relation does not become obsolete.

The experiment was performed using a data set of “4 Universoty” which could be extracted from WebKB. It comprised of multiple pages from different universities and those were grouped in to seven categories. Categories includes student, course, staff, faculty, department, project and few others. The experiment was performed only on a sub set of documents namely student, staff and course groups. This helped them to clear the ambiguity among some of the document categories.

In order to achieve a better accuracy in classification, they worked on both popularity and rarity metric in feature set extraction. But these metrics did not prove a better accuracy when compared with previous studies by other researchers. As a result of the research, They could minimize the noise of the feature set, in addition to limiting the size of the feature set by using the two metrics. As an example, for a larger document, there will be a large set of features and that will enable the document to be classified in to multiple classes. Hence, it was observed that classification process is largely benefited by limiting the feature set and thereby limiting the noise.

As a conclusion of this work, they achieved above 99% precision for rejecting the documents which are unrelated. Another observation was that there is 95% recall for related documents selection. When these feature extraction algorithms were used to implement standard classifiers, it gave some interesting results. Refer the figure below for information on classification accuracy of standard classifier algorithms.

Table 2. 1: Classification Accuracy of Standard Algorithms

|  |  |
| --- | --- |
| SVM (Original) | 89.8% |
| SVM (Filtered) | 80.2% |
| Naïve Bayes (Original) | 81.7% |
| Naïve Bayes (Filtered) | 90.7% |

# **Chapter 3: Design of the Solution**

Documents can be considered as one of the richest sources of data for any businesses irrespective of whether those are in the shape of customer support documents, Technical whitepapers, emails, User Acceptance Tests, Legal Documents, Contracts between parties, Policies, Non-Disclosure Agreements. Most of these documents contain valuable information that can be used for regular operations of the businesses. However, these documents comprise of information at various sensitivity levels for the businesses such as trade secrets, Product Information, Financial Information etc. It’s much important to identify these sensitivity levels of documents so that they can be monitored for unauthorized access or exchange between un-intended parties.

Segregating such documents in to different classes is called Classification. There are methodologies commonly used in classification. However, such algorithms and techniques struggle when processing heavily unstructured documents. Specially it’s a fact that these models work only for certain data sets but not for all. Document set and its domain heavily affects the accuracy achieved by the deployed model. Several previous studies have shown the comparison of commonly used document classification accuracies and performance against publicly available datasets [11].

Further, various existing Document Classification Methodologies can be briefed with their own advantages, disadvantages and which datasets they are good at as below.

Table 3. 1: Summary of Various Text Classification Techniques [12]

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Advantages | Disadvantages | Applications |
| Logistic Regression | Simple parameter estimation, works well for categorical predictions | Requires larger sample size, not suitable for non-linear problems, vulnerable to over-confidence | Financial forecasting, Software cost prediction, software effort prediction software quality assurance, Crime data mining |
| Naïve Bayes | Fast classifier, converges earlier than discriminative models like logistic regression, requires less training, applies for both binary and multi-class problems | Interactions between the features cannot be achieved. The probabilities calculated are not mathematically accurate, but relative probabilities. | To mark email as spam/ham, classify articles based on content, sentiment/emotion analysis. |
| SVM | Regularization parameter avoids over-fitting. Kernel engineering helps to incorporate expert knowledge. | Selecting the best kernel and time consumed for training and testing. | Good for biological datasets, hypertext categorization, etc., |
| Decision Trees | Simple to understand after providing explanation. Insights based on expert knowledge and dynamic. | Not suitable for multilevel categorical variables, biased information gain, complex for uncertain and multiple valued attributes. | Marketing data and customer intelligence |
| Rule Induction | Optimized rules are built based on lexical patterns of the domain | Inter-dependency among rules and sequential rule learning slows learning process for new class. | Healthcare Systems |
| K-NN | Simpler implementation, Flexible feature selection, good for multiclass problem | Searching nearest neighbors and estimating optimal k value | Recommender Systems |
| Artificial Neural Networks | Easier to use, approximates any kind of function, and almost matches human brain | Requires large training and test data, much of the operations are hidden and difficult to increase accuracy. | Sales forecast, data validation, risk management and target marketing |
| K- Means | Easy to implement, faster than hierarchical clustering and easy to interpret results. | Not good for global clusters and sensitive to outliers | Customer service segmentation, health care, fraud detection and Segmentation |
| Hebbian Algorithm | Suitable for multi-class models in neural networks. Easy to interpret layer-wise operations. | It could take only orthogonal inputs that are not correlated. | Suitable for Image and Speech recognition in artificial intelligence models. |
| Anomaly Detection | Interdependency between variables and prediction is clearly encoded, can integrate both historical information and current data | Difficulty in framing rules, sometimes outliers occur almost similar to original patterns. | Fraud detection, faults reporting, healthcare systems and networks |
| Expectation Maximization | Better suitable for heterogeneous datasets and simple to implement | It takes longer duration to converge | Image reconstruction, Probabilistic context free grammars and risk management in item response theory. |
| Singular Value Decomposition (SVD) | Robust to numerical errors, Reduces data dimensionality | Data has to be detrended before applying SVD and it must contain outliers / anomalies. | Digital signal and image processing applications. Recommender systems to predict ratings. |

As briefly described in Chapter 1, the focus of this research work is to formulate a proper document classification model using a different approach. However, in order to compare the accuracy levels, few standard implementations of existing models were also evaluated against the formulated dataset. (formulation of data set is described in Section 3.1.1) The design approach has several key steps and overall system architecture is given in below figure.

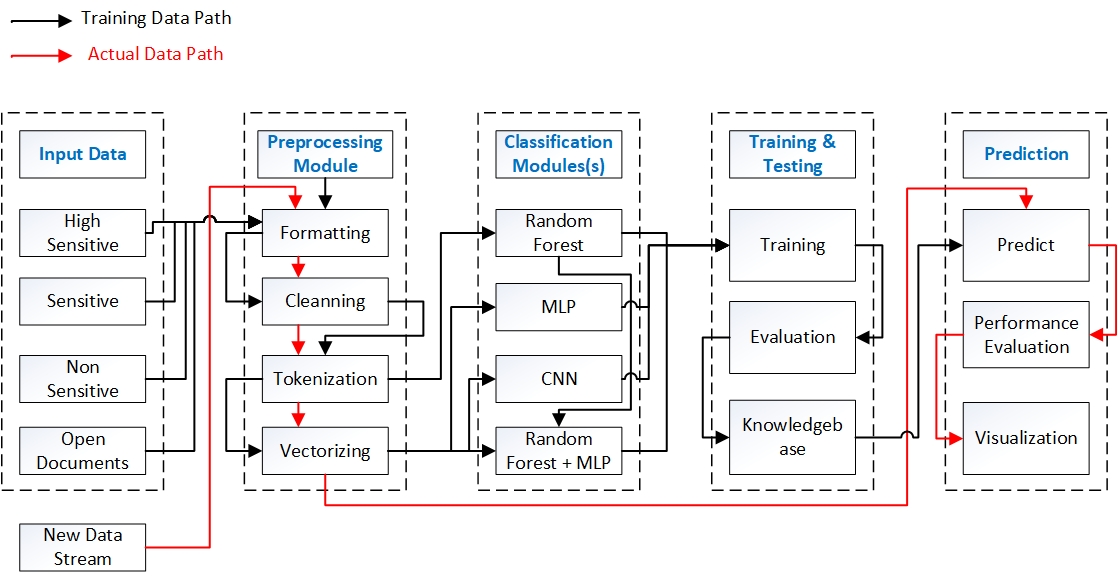


Figure 3. 1: Proposed Classification Architecture

The design of this work has several goals.

1. Input Data Identification and Preprocessing
2. Classification Algorithm and Training Phase
3. Testing against a known, labeled dataset
4. Performance Evaluation

## 3.1 Input Data Identification and Preprocessing

Input data identification and preprocessing involves the initial data set finding and its pre processing tasks before used in classification algorithm.

#### 3.1.1 Data Set

This research work is particularly targeting on a common document set available in Systems Integration industry. There will be no limitation applying the derived results in any other domain which is having a considerable document collection, but the accuracy of results will be not the same as what is discussed in Chapter 5.

Data Set plays a vital role in this type of classification exercise. In order to derive a more accurate model for classifying domain specific documents, it was much needed to have a proper sample document repository (Data Set). The selected documents collection should be quantitatively sufficient for the selected model to operate properly. Also, the collection should be qualitative enough so that it comprehensively represents the domain targeted.

In this research work, I used two data sets, one Data Set purposely built for this research work and one set extracted from the public domain;

1. Domain Specific Document set which represents System Integrators Industry
2. Publicly Available data set

##### Domain Specific Document Set and Preprocessing

Figure 3. 2: Steps Involved in (Systems Integrator) Data Preparation

More details on above data preprocessing architecture is discussed in Chapter 4. Briefly, the main components are;

Row Data: A set of documents collected in flavors of Legal Documents, Technical

Documents, HR related Documents and Policy Documents etc.

Type Conversion: Convert the document formats in to one text-based documents repository

Cleaning: Filter the words to be fed in to the classification algorithm

Vectorizing: Format the representation of words so that Classification algorithm can

perform calculation on it and derive decisions

##### Publicly Available Dataset

There are common, freely available data set repositories available in public domain. [13] [14]. These data sets are categorized in to various categories so that either set is selectable based on the study being done. The datasets are available in multiple categories such as Public & Government data sets, Housing Data sets, Finance & Economics data sets, Imaging Data sets and Clinical Data sets.

In this work, I used **Reuter’s data set** [15], by Martin Thoma in 2017, which is a multi-class and multi-label dataset. One document could be part of multiple classes. The requirement of this data set was to test the classification model for Convolution Neural Network (CNN) which will be discussed later in this chapter. Below are the detailed information on the Reuter’s data set.

* Classes: 90
* 10788 documents (80% was used as Training\_Data and 20% as Test\_Data in Convolution Neural Network)
* Mean number of words per document: Between 93 and 1263
* Vocabulary Size: 35247
* Number of words which appears at least 5 times: 12017 Words

Public data set was used for classification model implementations and testing purposes only. The actual experiments were done on the SI Data Set and results given in Chapter 5 is shared on SI data Set.

## 3.2 Classification Algorithms and Training

Machine learning based document classification can learn to make classification based in previous observation, instead of relying on manually defined rules. A machine learning algorithm can learn the different combinations and associations between text content and the relevant output classes by using pre labeled training and test data set.

As the first step of training a classifier with machine learning can be given as feature extraction. Feature extraction is a method used for [transforming each text into a numerical representation](https://monkeylearn.com/blog/beginners-guide-text-vectorization/) in the vector form. In this work I used [bag of words](https://machinelearningmastery.com/gentle-introduction-bag-words-model/) approach as the first step for creating the Vocabulary. (Discussed in Annexure I). Target here is to have a vector which could be a representation of the frequency of a word in a predefined words dictionary.

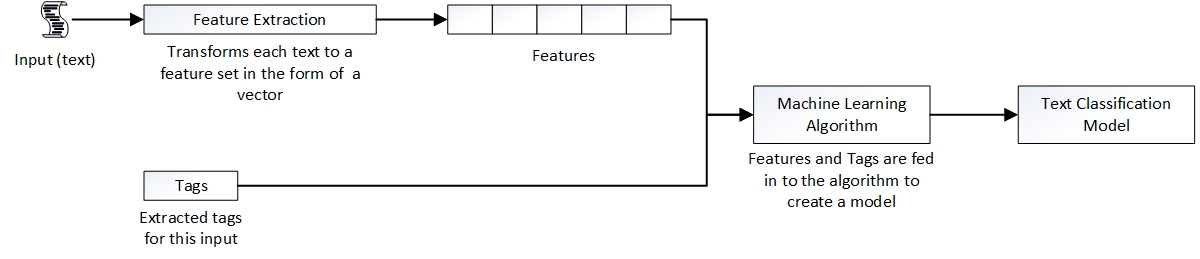
As the next step, the machine learning algorithm is fed with a set of pre labeled data which consists of pairs of feature sets (each text sample as a vector) and labels (classification level such as High Sensitive, Sensitive, Non-sensitive and Open) to produce a classification model. This initial pre labeling was done with the industry’s standard acceptance of sensitivity levels.

Figure 3. 3: Steps Involved in Classification

Some of the popular machine learning algorithms for document classification was studied and briefed in Section 2 of this document. Naïve Bays Classifier, K-Means Clustering, K Nearest Neighbor Classifier, Support Vector Machines and Decision Tree Algorithm were among those discussed.

In this work, I used four different classification techniques against the same dataset. Standard implementations of Classification algorithms as well as combined techniques were deployed. For an example, one model was, Radom Forest Classifier combined with Multi-Layer Perceptron Neural Network.

1. Random Forest Classifier
2. Neural Network Method
   1. Standard Implementation - Multi-Layer Perceptron Neural Network
   2. Altered Implementation - MLP + Random Forest
3. Convolution Neural Network

### 3.2.1 Random Forest Classifier

Random forests can be defined as a combination of decision tree predictors where each of these tree depends on the values of a random vector sampled independently and with the same distribution for all the identified trees in the forest [16]. Random Forest can be used as an ensemble learning method for regression and classification and problems. Same can be described as a combination of Decision Trees which are creates by randomly selecting set of vectors (Word Vectors in this work). For a Random Forest which consists of N trees, following equation can be used to predict the class label ***‘l’*** of a case ***‘y’*** through majority voting:

*l(y) =* *argmaxc*(

#### Mathematical Explanation of Random Forest

Assume there are ***n*** documents (samples), and feature vectors **Xi** (encoded documents with word dictionary) with outcomes ***y*i** (labels)

Data:

*D =* {(X1,*y1*),….,(X*n,yn*)}

Each Feature Vector

*D =* {(X1,*y1*),….,(X*n,yn*)}

Create a Decision Tree ***(h(X))*** where each node has a binary decision based on whether **X*i<a*** or not for a fixed ***a***

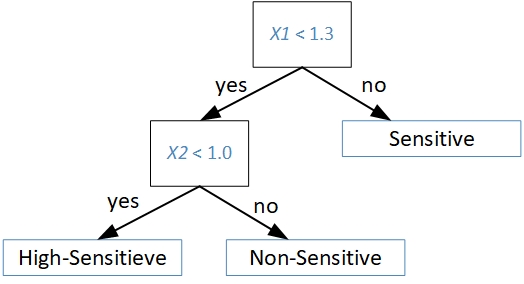


Figure 3. 4: Graphical Representation of a Decision Tree

The top node consists of all the sample in consideration (which were Randomly selected) **(X*k,yk*)**, and the set of examples is then subdivided among the children of each node. That is performed according to the classification at that node.

This process of subdivision is continued until every node at the bottom has examples which can belong in one class only. (High-Sensitive, Sensitive, Non-Sensitive or Open)

At each node, feature and threshold is chosen to minimize resulting *'diversity'* in the children nodes where this diversity is measured by *G,* the which is known as ‘Gini’ criterion.

**Gini Criterion:** Define class **C1** = High-Sensitive and **C2** = Non-Sensitive. The method to measure variation of samples in a node with respect to these two classes

As example, if there are 2 classes **C1, C2** and availability of examples in set S at current node.

To create child nodes, partition **S = S1 U S2**

(Note each sample **S1, S2** is now partitioned into the two classes called **C1, C2**)

**|S|** = # objects in set **S**

**=** proportion of

= **=** proportion of

Define the variation *g(Sj)* in set *Sj* to be:

*g(Sj) =*

Note: variation *g(Sj)*is largest if set *Sj* is equally divided among *Ci.* It’s smallest when all of *Sj* is just one of the *Ci.*

Define the variation of this full subdivision of the *Sj* as the Gini Index = G if;

*G = (S1)g(S1) + (S2)g(S2) =* weighted sum of variations *g(S1), g(S2)*

Assuming that the above Decision Tree was based on randomly selected samples of entire data D;

Then; ensemble of (random) decision trees can be denotes as below;

*h* = {*h1*(X),….,*hK*(X)} ; Where *h* is defined as the Random Forest

Define the parameters of the decision tree for classifier as

Same can be represented as;

*hK*(X) = *h(X|*

To derive the final classification *f*(X), which combines the classifiers {*hK*(X)}, each tree votes for the most popular class at input X. Then the class with the most votes wins.

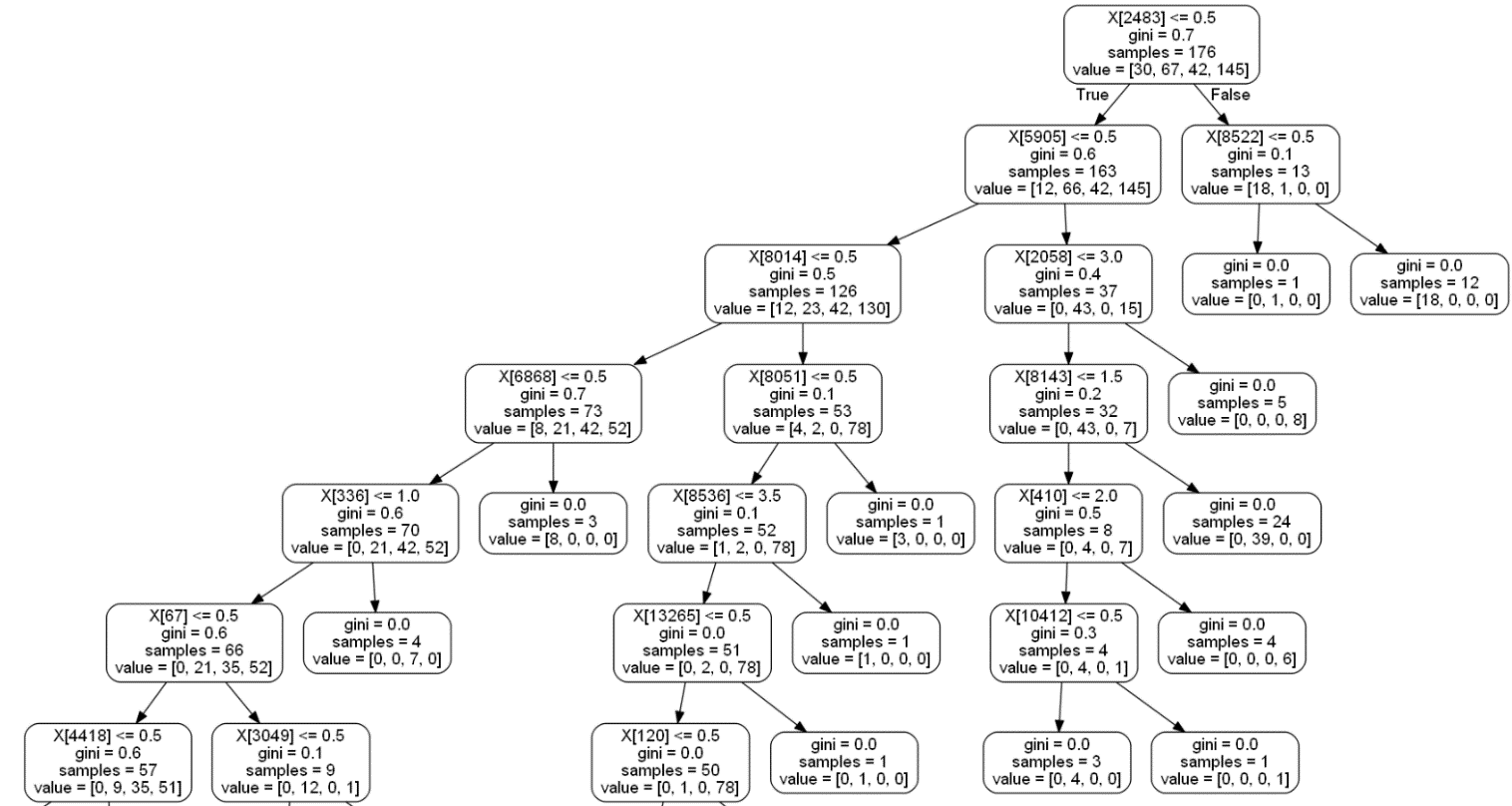
In this work I trained the labeled data set using Random Forest Classifier for 100 Decision trees per estimate. Following figure shows one of the Random Trees related to this work.

Figure 3. 5: decision Tree created in Classification

More details of the Implementation of Random Forest and achieved result is discussed in Chapter 4 and Chapter 5.

Functions used: Python Library ‘*sklearn*’ was used with sub library called ‘*ensemble*’.

*< from sklearn.ensemble import RandomForestClassifier>*

### 3.2.2 Artificial Neural Network:

Artificial neural networks (ANNs) work in the way that human brain works when it’s arriving at a decision. It works on the method of learning and evolution with no human intervention or at very low level of human intervention. There are multiple Neural Network models that can be applied in various scenarios expecting various results.

“An *artificial neural network* is a network of simple elements called *“*[*artificial neurons*](https://en.wikipedia.org/wiki/Artificial_neurons)*”*, which receive input, change their internal state (*activation*) according to that input, and produce output depending on the input and activation.” [20]

Artificial Neural Networks receive an input which could be represented as a single vector. And then transforms it through a series of “*hidden layers”*. Each of these hidden layers comprise of a set of neurons, where each neuron is fully connected to all neurons of the previous layer. Neurons in a particular layer functions completely independently and it will not share any of the connections. The last fully connected layer is called “*output layer*” where it is the representation of the class scores in classification task.

There are multiple variations of Neural Networks and their own application domains as below [17]

Table 3. 2: Organization of Neural Networks Based on Their Functional Characteristics

|  |  |
| --- | --- |
| Functional Characteristics | Structure |
| Pattern Recognition | MLP, Hopfield, Kohonen, PNN |
| Associative Memory | Hopfield, recurrent MLP, Kohonen |
| Optimization | Hopfield, ART, CNN |
| Function Approximation | MLP, CMAC, RBF |
| Modeling and Control | MLP, recurrent MLP, CMAC, FLN, FPN |
| Image Processing | CNN, Hopfield |
| Classification (including Clustering) | MLP, Kohonen, RBF, ART, PNN |

MLP: Multi-Layer Perceptron Neural Network

PNN: Probabilistic Neural Network

ART: Adaptive Resonance Theory

CNN: Convolution Neural Network

FLN: Fast Learning Network

CMAC: Cerebellar Model Articulation Controller

RFPM: Random Forest Probability Machine

PNN: Probabilistic Neural Network

RBF: Radial Basis Functions

In this work I worked with two models of neural Networks. Objective was to measure the accuracy and check the applicability of different neural networks in text classification context, especially for purpose-built data set for SI Industry. Two Models I worked with are

1. Multi-Layer Perceptron Neural Network (MLP)
2. Convolution Neural Network (CNN)

In both these models, I made alternations in the deployment beyond the standard implementation. Those are briefly discussed below and further details on implementation is discussed in Chapter 4

#### MLP Neural Network

Ordinary Neural Network was implemented in two different ways;

1. Standard Implementation: Applying of the complete SI Data set for Training of the

Model

1. Altered Implementation: Derive the Priority Words for the Classification and use only

that list of words for Training the Model

#### Standard Implementation - MLP

Feedforward Multi-Layer Perceptron (MLP) Neural Network was selected to be implemented in the first phase. The feedforward neural network can be introduced as a simpler implementation type of artificial neural networks. [18]

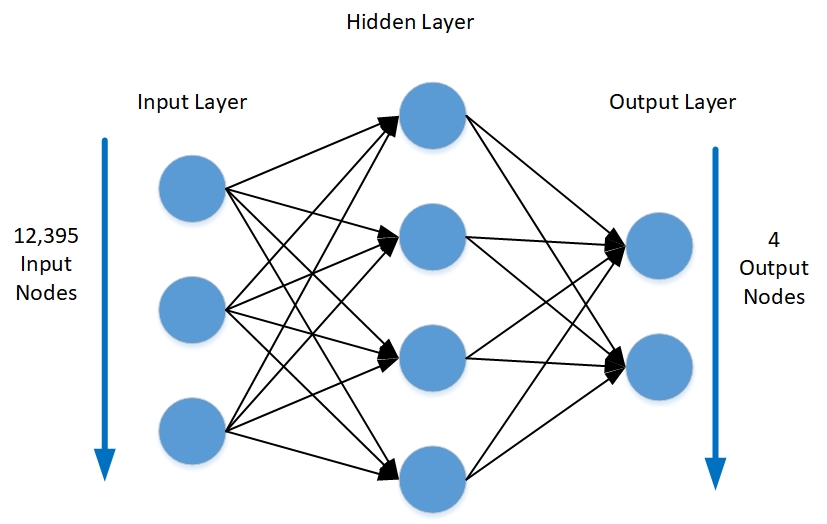


Figure 3. 6: Standard Representation of the Implemented Neural Network

A Multi-Layer Perceptron (MLP) is a category of [deep artificial neural network](https://skymind.ai/wiki/neural-network)s. It consists of more than one perceptron and those are composed of an input layer to receive the signal, an output layer and a random number of hidden layers in between. Function of the output layer is to represent a decision or prediction about the input. Hidden layers represents the computational core of the MLP. If an MLP is available with one hidden layer, then it’s capable of approximating any continuous function. [19]

MLP’s are often applied to supervised learning problems. They learn to model the correlation or dependencies between inputs and outputs by training on a set of input-output pairs.

The training Phase involves adjusting the parameters, or the weights and biases, of the hidden layers for minimizing the errors. Backpropagation is the method used to make the weigh and bias adjustments in relation to the error. The error itself is measured in by means of Root Mean Squared Error (RMSE).

* In this work, 12,395 Features are fed in to the Input layer which represents the individual Key Values of the Vocabulary created.
* Output Layer was implemented with 4 nodes in the layer with One hot encoding
  + [0 0 0 **1**] High Critical
  + [0 0 **1** 0] Critical
  + [0 **1** 0 0] Non-Critical
  + [**1** 0 0 0] Open
* One Hidden Layer was Implemented
* Different Values for Epochs and Batch size was tested.

Results of multiple iterations with different values are discussed in Chapter 5.

#### Altered Implementation – MLP + Random Forest

As discussed in Section 3.2.2 (Standard Implementation), MLP Neural Network was implemented with selected Input Features only. Intention of this experiment was to observe the performance and accuracy of the results when implemented with selection of Input features.

Input Feature Selection Criteria:

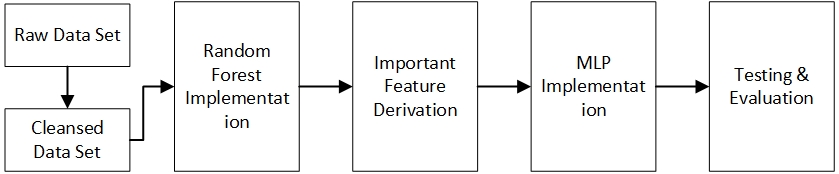


Figure 3. 7: Feature Selection from Random Forest Classification

* As the first step, Random Forest model was implemented and Trained against Training Data Set
* First 1024 words which mostly influenced the Random Forest’s decision was derived.
* The derived words were made as Input for the MLP
* MLP was trained and Results were obtained for Comparison

### 3.2.3 Convolution Neural Network

As the fourth model in this work, I evaluated the possibility of Convolution Neural Network being utilized in Text based Document Classification. Convolution Neural Networks are mostly known for the Image Processing such as Image Classification or Object detection etc.

Apart from Image processing, 1D Convolution Networks have been used in Natural Language Processing and other time series analysis problems. 2D Convolution networks mainly works on color images as those are easily represented as 3D matrices.

The focus area in this work was to study the possibility of representing a text documents as an image, more precisely as a gray scale image. That will allow a set of Training documents to be represented as a 4D array where;

* The first digit will represent the number of Training Documents in rows
* The second and third digits will represent the height and the width of the Image (Image = Document represented as Rows and Columns)
* Last digit will represent the Image considered is a Gray Scale (document is gone through feature reduction and already represented as vectorized array with ‘1’s and ‘0’s)

And once the set of documents are represented as images, the 2 D Convolution Matrix can be applied on that data set. Because 2 D Convolution Matrix would work best for any “spacial” representation of data.

For an example of spacial representation of a text-based document;

Original Document Sample: [*Apart from Image processing, 1D Convolution Networks have been used in Natural Language Processing and other time series analysis problems. 2D Convolution networks mainly works on color images as those are easily represented as 3D metrices* ]

Vectorized Document: [0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0

1 0 1 1 0 1]

200 Columns

Image Representation: [ [0 1 0 1 1 0 1]

[0 1 0 1 1 0 1]

100 Rows

[0 1 0 1 1 0 1]

[0 1 0 1 1 0 1] ]

One main advantage of this representation is, the order of word appearance is also well captured. None of the other techniques selected in this work considered the order of word appearance, hence the result of CNN was expected to be better over others

How the Convolution Neural Network worked compared to other three methodologies will be discussed in Chapter 4.

Activation Function Used in this Work:

ReLU:

Artificial neural networks use [**rectified linear units**](https://github.com/Kulbear/deep-learning-nano-foundation/wiki/ReLU-and-Softmax-Activation-Functions) (ReLUs) for the hidden layers. A ReLU has output 0 if the input is less than 0, and raw output otherwise. If the input is greater than 0, then the output is always equal to the input.

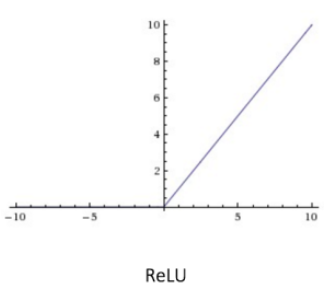
f(x) = max(0, x)

Figure 3. 8: ReLU Graphical Representation

Softmax

The “[*Softmax*” activation function](http://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function/) is generally used for multi-class classification tasks. This function calculates the probabilities distribution of a particular event over ’*n*’ different events. As a summary, this function will calculate the probabilities of each target class over all possible target classes. Finally, the target class for the given inputs could be determined by the the calculated probabilities.

In both Neural Network based methodologies, I used the *softmax* function at the output layer as it always gives probability values for each classification categories.

## 3.3 Testing against a known dataset

Once training is completed with enough training samples, the machine learning model starts making actual predictions. In this process, the same feature extractor is used to transform the unseen text in to feature sets which can be fed into the classification model for obtaining predictions:

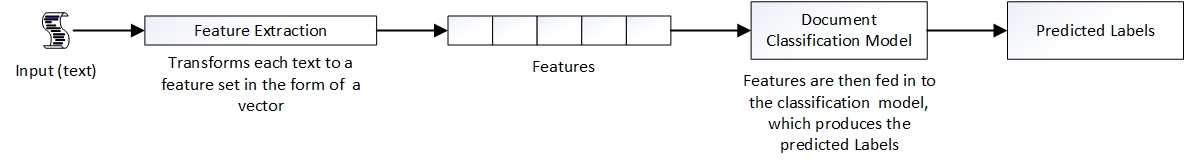


Figure 3. 9: Testing Against a Pre-Labeled Data Set

## 3.4 Performance Evaluation

‘Percentage Error’ was used in all evaluations which can be expressed as below;

M = *(n – ns); where ns is number of successful prediction, n is the number of test cases*

For visualization purposes, Heat Maps were created against each test case which is presented in Chapter 5.

# **Chapter 4: Implementation**

## 4.1 Data Set used in analysis

As briefly discussed in Chapter 3, proper data set is one of the key factors for the success of a Machine Learning exeriment in this nature. There are two factors that directly affects the eligibility of the selected dataset for the desired model and output. [21] [22]

1. Quality of the dataset:
2. Quantity of the dataset:

In this research work, quality of the data set was evaluated on whether the documents have the correct labels. More the similarities in documents which belongs to different classification levels, more the quality of data set. That ensures the robustness of the classification model against any data set in same domain. If the documents in different classed are entirely different on the feature vector, then the trained model (predictive model) will produce false positives and negatives when actual data or test data is fetched.

Having the quality of data set is not sufficient for a machine learning experiment, quantity has to be large enough to properly train the model. The values associated with variables used for predictions has to have more diversification so that learning model can expand in to a much larger scope in actual prediction stage.

### 4.1.1 Data Set Identification

In this work, originally there was no publicly available data set in similar manner. Hence, I started by collecting common data that we see in Systems Integration Industry on regular basis. The main Challenge was to work with many file formats and not having a common repository to collect data from a single source.

The File Formats I considered here is given below with associated data description ass well in below table.

Table 4. 1: Data Formats and Associated Security Labels in SI Data Set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Types of Data** | **Format** | **Security Label** | **Conversion to** |
| 1 | White Papers | pdf/word | 0 | text |
| 2 | Data Sheets | pdf | 0 | text |
| 3 | Specification Sheets | pdf | 0 | text |
| 4 | Company Policies | pdf/word | 1 | text |
| 5 | Memo/Procedures | pdf/word | 1 | text |
| 6 | Contract Documents | pdf/word | 2 | text |
| 7 | Legal Documents | pdf | 2 | text |
| 8 | Non-Disclosure Agreements | pdf | 2 | text |
| 9 | HR Related Documents | pdf/word | 3 | text |
| 10 | Personal Documents | pdf/word | 3 | text |

Exclusions: Few Data source which was identified as possible document formats in Systems Integration Industry, but not taken in to consideration in this scope is given in below table.

Table 4. 2: Excluded Document Formats in SI Data Set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Types of Data** | **Format** | **Security Label** | **Conversion to** |
| 1 | BoM files | excel |  | text |
| 2 | Solution Presentations | ppt |  | text |
| 3 | Presentations | ppt |  | text |
| 4 | Quotations | excel |  | text |

File Format Conversion - Pseudocode:

*For all the file in <dir>;*

*Open the file as whole in to contents;*

*Open a new text file in append and binary mode;*

*Pass the contents in to the new file;*

*Close new file;*

All the pdf files were passed through the python code to convert those in to text format.

### 4.1.2 Data set preparation

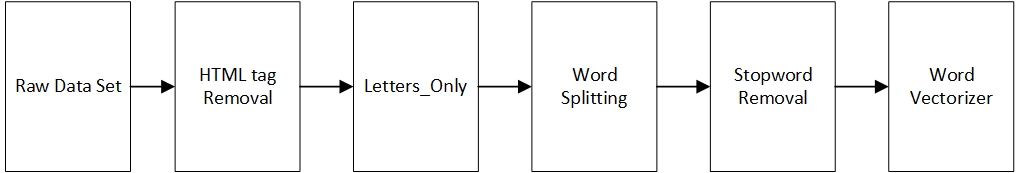
The next step was to clean the text data by removing unnecessary words. Only few words can be termed as *keywords* in a text document to characterize the document. Therefore, a filtering process was adopted in order to remove unnecessary words in all documents. The steps and the performed action under each step is described using below figure.

Figure 4. 1: Flow of Data Preparation

HTML Tag Removal:

In this work I used the python library called *beautifulsoup* for removing tags from HTML and XML files. It was included in the implementation in the initial stage even though the web pages were not considered initially.

Removing Stop-words:

When processing natural language, *stop-words* are often filtered out. *Stop-words* **are the words** commonly inside a document such as “the”, “a”, “an”, “in”. These lists generally refer to most common words in English. Even though that there is no any universal list of such stop-words, there are tools that can generate different stop-words lists. Stop-words are usually deemed irrelevant and are often dropped from the text. In this experiment, I used pyhon provided library called *nltk*:

*from nltk.corpus import stopwords*

Ex of stop-words and related python code:

import nltk

from nltk.corpus import stopwords

set(stopwords.words(‘english’)

*{‘ourselves’, ‘hers’, ‘between’, ‘yourself’, ‘but’, ‘again’, ‘there’, ‘about’, ‘once’, ‘during’, ‘out’, ‘very’, ‘having’, ‘with’, ‘they’, ‘own’, ‘an’, ‘be’, ‘some’, ‘for’, ‘do’, ‘its’, ‘yours’, ‘such’, ‘into’, ‘of’, ‘most’, ‘itself’, ‘other’, ‘off’, ‘is’, ‘s’, ‘am’, ‘or’, ‘who’, ‘as’, ‘from’, ‘him’, ‘each’, ‘the’, ‘themselves’, ‘until’, ‘below’, ‘are’, ‘we’, ‘these’, ‘your’, ‘his’, ‘through’, ‘don’, ‘nor’, ‘me’, ‘were’, ‘her’, ‘more’, ‘himself’, ‘this’, ‘down’, ‘should’, ‘our’, ‘their’, ‘while’, ‘above’, ‘both’, ‘up’, ‘to’, ‘ours’, ‘had’, ‘she’, ‘all’, ‘no’, ‘when’, ‘at’, ‘any’, ‘before’, ‘them’, ‘same’, ‘and’, ‘been’, ‘have’, ‘in’, ‘will’, ‘on’, ‘does’, ‘yourselves’, ‘then’, ‘that’, ‘because’, ‘what’, ‘over’, ‘why’, ‘so’, ‘can’, ‘did’, ‘not’, ‘now’, ‘under’, ‘he’, ‘you’, ‘herself’, ‘has’, ‘just’, ‘where’, ‘too’, ‘only’, ‘myself’, ‘which’, ‘those’, ‘i’, ‘after’, ‘few’, ‘whom’, ‘t’, ‘being’, ‘if’, ‘theirs’, ‘my’, ‘against’, ‘a’, ‘by’, ‘doing’, ‘it’, ‘how’, ‘further’, ‘was’, ‘here’, ‘than’}*

Vectorization:

Word Vectorization was done for interpreting the document as a vector. Machine Learning algorithm and Neural Networks can only work on vectors, not on individual words. How the vectorization was done is elaborated under relevant Machine Learning and Neural Network implementation in this section of the document. Because the vector has to be prepared in slightly different ways for different classification approaches.

### 4.1.3 Data Set Segregation

Training Data Set: The training data set is the one used to train an algorithm to understand how to apply different concepts. In this work four classification models were tested for its accuracy by using training data set. It includes both input data and the expected output which is called the data label.

Test Data set: The test data set was used to evaluate how well the implemented algorithm was trained with the training data set.

Proportion of the Training:Test data always become a key factor for the ultimate accuracy of the result. In this work, I tested the accuracy of each classification methodology against different Training:Test ratios. The relevant results are given in Chapter 5: Results and Evaluation.

Validation Data Set: validation sets are used to select and tune the final ML model.

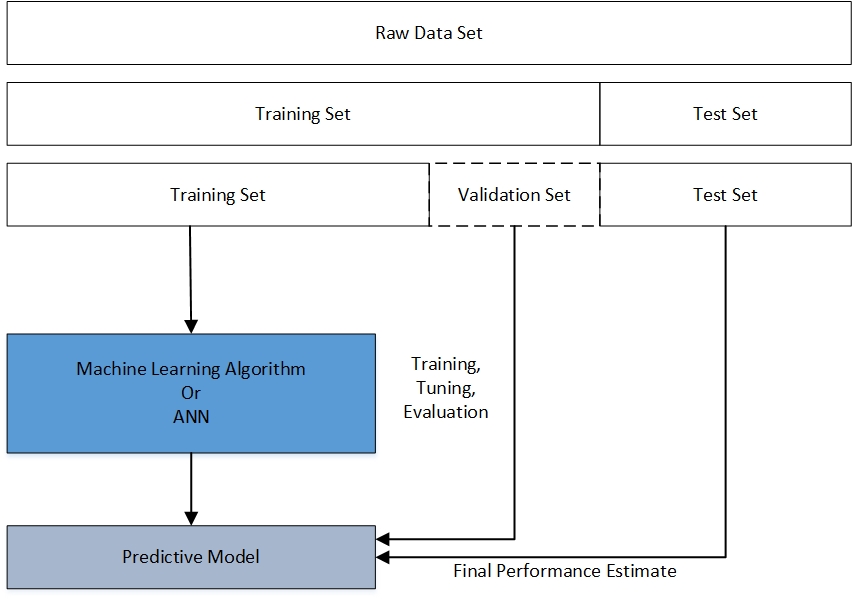


Figure 4. 2: Data Set Segregation

## 4.2 Classification Model Implementation

### 4.2.1 Random Forest

As described in Chapter 3 in detail, a Random Tree model was implemented against the prepared SI dataset.

#### `4.2.1.1 Steps of Implementation

Step 1: Data set Preparation so that clear text is formulated out of documents as below

*acceptance credit debit card payments printemail responsible official university controller chief information security officer responsible office office controller information security policy compliance office effective date december revision date may policy sections requesting new merchant identification number mid service provider selection mid record maintenance training requirements cardholder data ………………………………………………………………………….. within department pci dss compliant cooperating pci administration completion annual attestation process completing monthly reconciliation payment card activity individual employees responsible securing cardholder data processing redacting sensitive information scanning shredding disposing individuals must adhere policies related resources pr payment credit debit card*

Step 2: Formulate a Train and Test Data in Proper Structure so that those can be fed in to the Random Forest Algorithm

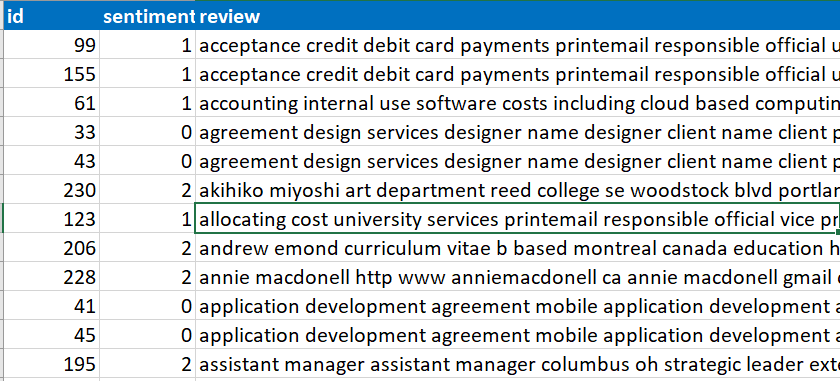


Figure 4. 3: Prepared SI Data Set

Step 3: Vectorize the Documents in Data set and Create a Unique Vocabulary

*vectorizer=CountVectorizer(analyzer='word',tokenizer=None,preprocessor = None, stop\_words = None,max\_features = 20100)*

*train\_data\_features=vectorizer.fit\_transform(clean\_train\_review)*

*train\_data\_features=train\_data\_features.toarray()*

The Function Vectorizer was used to create a set of unique words in to an *numpy array* while representing each document by each row. ‘1’ in a row represents the presence of a specific word in that location for that document.

This entire process is followed in order to reduce the features of the document and represent the documents in vectorized format so that the Random Forest algorithm can understand the documents and train the prediction model.

Sample output of the vectorization process is given below;

[[0 1 0 1 1 0 ………….0 1 0 1 0 1]

[0 0 1 0 0 1 ………….0 1 0 0 1 0]

[0 0 1 0 1 0 ………….1 1 0 0 1 1]

[0 1 0 0 0 1 ………….0 1 1 1 0 0]

………….

………….

[0 0 0 1 1 0 ………….0 1 0 1 1 0]

[0 1 1 0 0 1 ………….0 1 0 0 1 1]

[0 0 0 1 1 0 ………….1 1 0 1 0 0]

Above was a *‘numpy’* (python) array with 406 rows and 13269 columns. These columns represent each unique word in the word Dictionary/Vocabulary.

Step 04: Random Forest Model Implementation

Python library was used to implement the Random Forest Model.

Training:

* The training data set was obtained as a portion of entire data set which was created earlier.
* The results accuracy was observed for different ratios of Training : Test data
* As discussed in Chapter 3 in detail, Random Forest is about having created multiple Decision Trees and selecting the best classification level based on the maximum vote mechanism. In this work, I derived the results for 100 estimators (100 decision trees) all the time as it was giving the maximum accuracy in each scenario.

Testing:

* As discussed in section 4.2.1.1, the training data set was cleansed and vectorized following the same ‘*vocabulary*’ created above.
* Test data was fed in to the trained Random Forest algorithm.
* Results were obtained for multiple ratios of Train : Test data in different format such as Arrays and Visualizations
* The results obtained is detailed discussed in Section 5.

### 4.2.2 Artificial Neural Network (ANN)

MLP Neural Network implementation was done for two different set of vectorized inputs. One Input was prepared with standard feature reduction technologies while other Input was directly extracted from the Random Forest Implementation. (Combination of Random Forest + MLP Neural Network Model)

These two implementations were discussed under ‘Standard Implementation’ and ‘Altered Implementation’ earlier. Results were obtained and analyzed for the best possible accuracy level on each implementation considering the SI Industry Data set. Observations are discussed in detail in the Chapter 5.

The pseudocode for the above implementation is given in Annexure II.

### 4.2.3 Convolution Neural Network (CNN)

Convolution Neural Network was discussed in detail in Chapter 3. CNN is mostly applied in the industry in Analyzing the images. It can work on a data set which is having a special distribution. Hence, in this work I expected to observe the applicability of CNN for the text-based document classification.

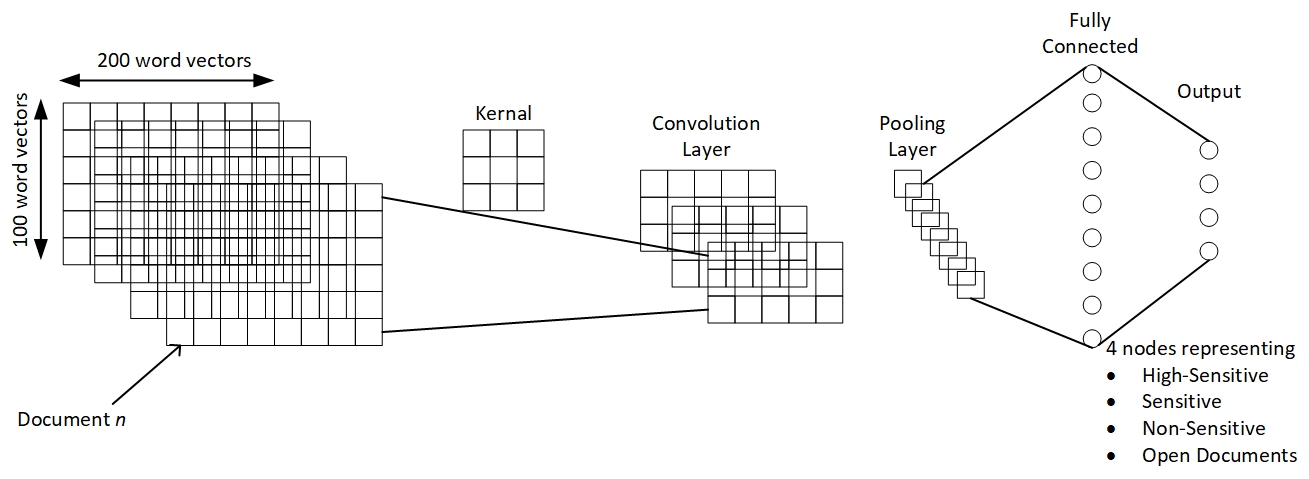


Figure 4. 4: Implementation of Convolution Neural Network

Implementation Paramters

* Each document was represented as a 4D Array before the Training phase as below

Document representation = [<# of Documents>, 100, 200, 1]

Where each document is a 2-dimensional array with 100 rows and 200 columns.

* kernal size (filter) = 3 x 3, same kernel for all layers.
* Activation function for convolution layers= ‘relu’
* Activation function for output layer = ‘softmax’

## 4.3 Technologies Used

Python was used ass the implementation language. Python is a high-level programming language and there are two major versions, Python 2 and Python 3 where I used Python 3 in this work. It has become an easy use language due to higher number of packages available and being continuously developed by the community.

* Libraries Used: <numpy, nltk, pandas, scikit, seaborn, matplotlib, tensorflow, keras>

Integrated Development Environemnt (IDE):

* Spyder Editor and Anaconda Navigator Framework

# **Chapter 5: Results and Evaluation**

Results obtained for four different classification approaches are given in below tables respectively for Random Forest, MLP, CNN and MLP + Random Forest respectively. Multiple round were tested and average accuracy value was taken for Train:Test ratios starting from from 20:80 to 90:10

## 5.1 Random Forest Model

Table 5. 1: Accuracy Figures at Multiple Iterations of Random Forest Implementation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train Percentage | 1st Round | 2nd Round | 3rd Round | 4th Round | Average | Train Samples | Test Samples | False\_Count |
| 90% | 97 | 96 | 95 | 94 | 96.5 | 365.4 | 40.6 | 1.421 |
| 80% | 94 | 94 | 91 | 92 | 94 | 324.8 | 81.2 | 4.872 |
| 70% | 92 | 90 | 90 | 89 | 91 | 284.2 | 121.8 | 10.962 |
| 60% | 89 | 93 | 88 | 88 | 91 | 243.6 | 162.4 | 14.616 |
| 50% | 90 | 90 | 94 | 92 | 90 | 203 | 203 | 20.3 |
| 40% | 91 | 88 | 86 | 84 | 89.5 | 162.4 | 243.6 | 25.578 |
| 30% | 94 | 88 | 90 | 89 | 91 | 121.8 | 284.2 | 25.578 |
| 20% | 91 | 94 | 91 | 90 | 92.5 | 81.2 | 324.8 | 24.36 |

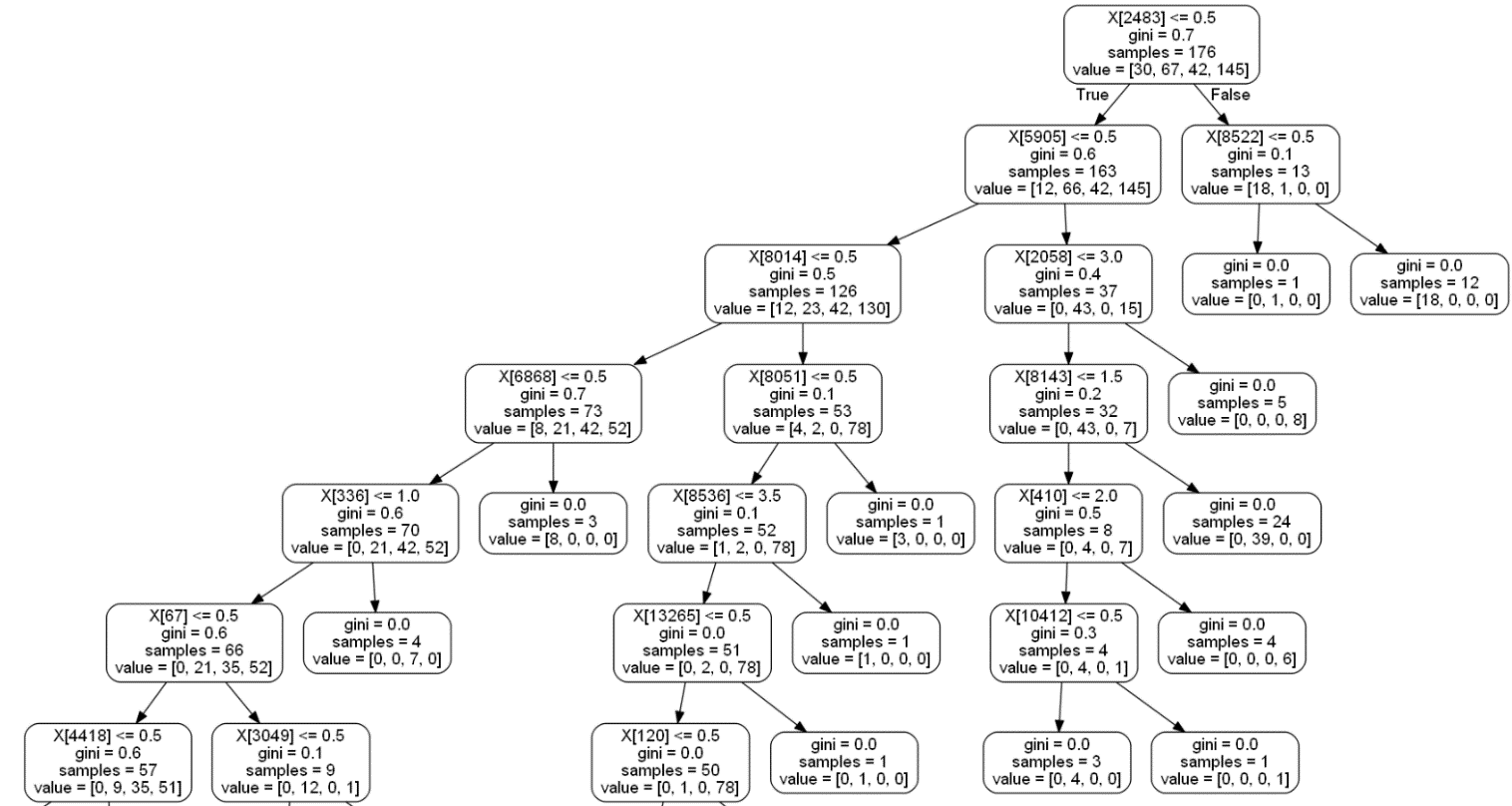


Figure 5. 1: Representation of a Sample Random tree

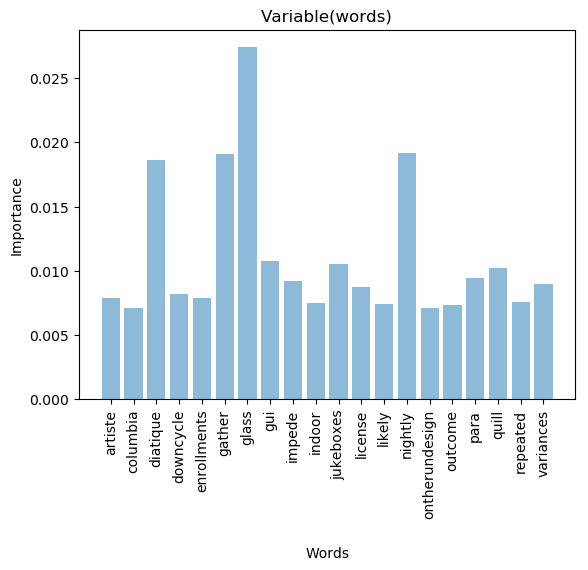


Figure 5. 2: Representation of 20 Important Words in Random Forest

Above figure shows different Importance levels observed for different words in the data set. Importance is given as percentage value and as shown the word ‘glass’ shows the highest importance level in classification.

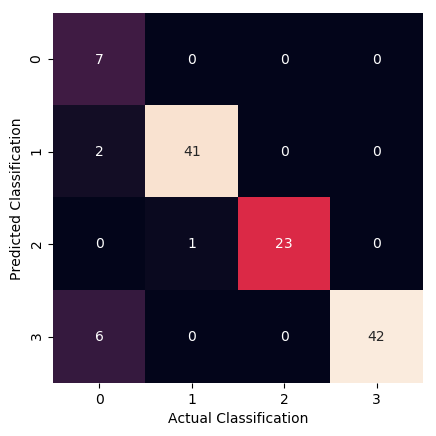


Figure 5. 3: Accuracy at 70% Train:Test Ratio

## 5.2 MLP Neural Network

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train Percentage | 1st Round | 2nd Round | 3rd Round | 4th Round | Average | Train Samples | Test Samples | False\_Count |
| 90% | 95 | 92 | 94 | 92 | 93.25 | 365.4 | 40.6 | 2.7405 |
| 80% | 75 | 62 | 60 | 75 | 68 | 324.8 | 81.2 | 25.984 |
| 70% | 96 | 97 | 95 | 96 | 96 | 284.2 | 121.8 | 4.872 |
| 60% | 98 | 98 | 97 | 98 | 97.75 | 243.6 | 162.4 | 3.654 |
| 50% | 98 | 98 | 96 | 95 | 96.75 | 203 | 203 | 6.5975 |
| 40% | 34 | 34 | 35 | 38 | 35.25 | 162.4 | 243.6 | 157.731 |
| 30% | 78 | 80 | 80 | 84 | 80.5 | 121.8 | 284.2 | 55.419 |
| 20% | 88 | 90 | 87 | 86 | 87.75 | 81.2 | 324.8 | 39.788 |

Table 5. 2: Accuracy Figures at Multiple Iterations of MLP Implementation

Highest accuracy achieved was 98% which is for 60:40 Train:Test ratio.

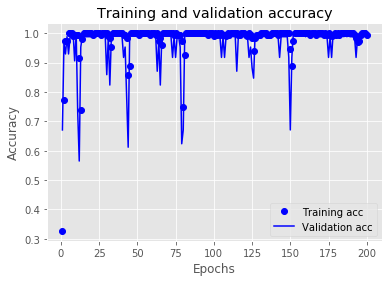
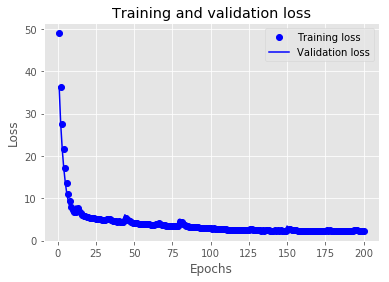


Figure 5. 4: Loss and Accuracy for Training and Validation

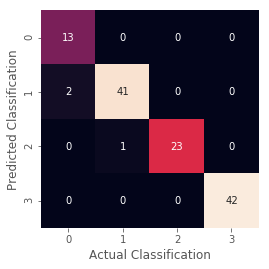
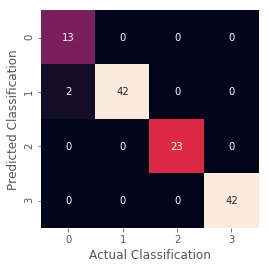


Figure 5. 5: Accuracy at 70% (left) & 60% (right) Train:Test

## 5.3 CNN Model

Table 5. 3: Accuracy Figures at Multiple Iterations of 2-Dimensional CNN Implementation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train Percentage | 1st Round | 2nd Round | 3rd Round | 4th Round | Average | Train Samples | Test Samples | False\_Count |
| 90% | 97 | 96 | 96 | 96 | 96.25 | 365.4 | 40.6 | 1.5225 |
| 80% | 96 | 96 | 96 | 94 | 95.5 | 324.8 | 81.2 | 3.654 |
| 70% | 94 | 95 | 95 | 95 | 94.75 | 284.2 | 121.8 | 6.3945 |
| 60% | 90 | 91 | 93 | 92 | 91.5 | 243.6 | 162.4 | 13.804 |
| 50% | 89 | 88 | 86 | 87 | 87.5 | 203 | 203 | 25.375 |
| 40% | 80 | 80 | 78 | 79 | 79.25 | 162.4 | 243.6 | 50.547 |
| 30% | 92 | 90 | 92 | 90 | 91 | 121.8 | 284.2 | 25.578 |
| 20% | 66 | 70 | 68 | 68 | 68 | 81.2 | 324.8 | 103.936 |

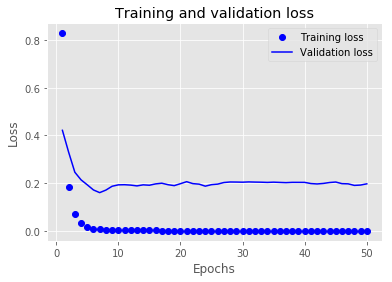


Figure 5. 7: Training and Validation Loss at 70% Train:Test Ratio

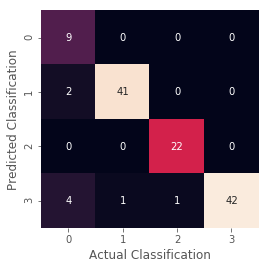


Figure 5. 8: Accuracy at 70% Train:Test Ratio

## 5.4 RF + MLP Model

Table 5. 4: Accuracy Figures at Multiple Iterations of Random Forest+MLP Implementation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train Percentage | 1st Round | 2nd Round | 3rd Round | 4th Round | Average | Train Samples | Test Samples | False\_Count |
| 90% | 55 | 61 | 40 | 38 | 48.5 | 365.4 | 40.6 | 20.909 |
| 80% | 35 | 44 | 44 | 46 | 42.25 | 324.8 | 81.2 | 46.893 |
| 70% | 45 | 43 | 45 | 39 | 43 | 284.2 | 121.8 | 69.426 |
| 60% | 56 | 58 | 64 | 62 | 60 | 243.6 | 162.4 | 64.96 |
| 50% | 34 | 39 | 42 | 62 | 44.25 | 203 | 203 | 113.1725 |
| 40% | 35 | 38 | 44 | 44 | 40.25 | 162.4 | 243.6 | 145.551 |
| 30% | 43 | 40 | 38 | 46 | 41.75 | 121.8 | 284.2 | 165.5465 |
| 20% | 34 | 28 | 36 | 38 | 34 | 81.2 | 324.8 | 214.368 |

Table 6. 1: Summary of Observations for Various Implementations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train Percentage | MLP | RandomForest | CNN | RF+MLP |
| 90% | 93.25 | 96.5 | 96.25 | 48.5 |
| 80% | 68 | 94 | 95.5 | 42.25 |
| 70% | 96 | 91 | 94.75 | 43 |
| 60% | 97.75 | 91 | 91.5 | 60 |
| 50% | 96.75 | 90 | 87.5 | 44.25 |
| 40% | 35.25 | 89.5 | 79.25 | 40.25 |
| 30% | 80.5 | 91 | 91 | 41.75 |
| 20% | 87.75 | 92.5 | 68 | 34 |

Figure 6. 1: Accuracy Levels of Four Different Techniques against various Train:Test Ratios

Figure 6. 2: Accuracy Variation against various Train:Test Ratios

# **Chapter 6: Conclusion and Future Work**

Throughout the results of previous researches done on document classification, it’s clear that the classification accuracy relies on the nature of the data set. Hence, this experiment solely focused on working on a domain specific data set for Systems Integration industry.

Once the data set is prepared, next focus was to test standard classification techniques and new classification techniques. For an example, Random Forest is one of the well accepted and standard implementation of classification while 2-Dimensional Convolution Neural Network has not been used in document classification.

Final results after multiple iterations of each technique yields few important facts and summarized in Table 6.1. Though Random Forest was known for higher accuracy levels regardless of the nature of the data set, MLP Neural Network with one hidden layer overrun the result by 7%. This clearly observed improvement was obtained for 60:40 and 50:50 Train:Test ratio.

Data set enhancement is one key area to focus on in future. Improvements can be done by adding more categories in to the data set with higher number of documents. More the document in each category, more the accuracy is.

The current study did not pay attention to the images contained within the data sets. Image classification techniques could be incorporated to the final implementation so that more granular and focused classification is performed.

Furthermore, the implemented CNN model should be tested for many other representations of input data. The only representation tested in this work is (200 x 100) bit map representation of input document. Optimum document representation format should be uncovered for each classification level by means more experiments.

Combined model (MLP+RF) can be further explored for optimum feature set extraction (according to the Importance of words from RF) for a better result. But this would not be a realistic model as this technique was extensively demanding more computation power compared to others.

# **References**

[1] Vontu, Inc., San Francisco, CA, “7 Requirements of Data Loss Prevention - A guide to evaluating solutions that protect your confidential information assets”, pp 1-4, 2007

## [2] Nageswaran Kumaresan, “Key Considerations in Protecting Sensitive Data Leakage Using Data Loss Prevention Tools”, <https://www.isaca.org/Journal/archives/2014/Volume-1/Pages/Key-Considerations-in-Protecting-Sensitive-Data-Leakage-Using-Data-Loss-Prevention-Tools.aspx>, Volume 1, 2014

[3] SANS Institute and Securosis, L.L.C., Sponsored by WEBSENSE, “Understanding and Selecting a Data Loss Prevention Solution”, pp 5 - 15

[4] ERNST & YOUNG, “Data loss prevention - Keeping your sensitive data out of the public domain”, pp 11 – 13, October 2011

[5] Rich Mogull, DLP Content Discovery: Best Practices for Stored Data Discovery and Protection - Securosis, L.L.C.

[6] Karel Fuka and Rudolf Hanka, “Feature Set Reduction for Document Classification Problems”, University of Cambridge, UK,

[7] S M Kamruzzaman and Chowdhury Mofizur Rahman, “Text Categorization using Association Rule and Naïve Bayes Classifier”, International Islamic University Chittagong, Bangladesh

[8] Thorsten Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, University of Dortmund, Baroper, Germany

[9] Russell Power, Jay Chen, Trishank Karthik and Lakshminarayanan Subramanian, “Document Classiﬁcation for Focused Topics”, NewYork University, New York

[10] A. Basu, C. Watters and M. Shepherd, “Support Vector Machines for Text Categorization” Dalhouse University, Canada B3H 1W5

[11] B S Harish, D.S. Guru and S Manjunath, “Representation and Classification of Text Documents: A Brief Review”, Manasagangotri, University of Mysore, India, IJCA Special Issue on ‘Recent Trends in Image Processing and Pattern Recognition’, pp116-119, 2010

[12] M. Thangaraj and M. Sivakami, “Text Classification Techniques: A Literature Review”, Interdisciplinary Journal of Information, Knowledge and Management, Volume 13, pp127-129, 2018

[13] Stacy Stanford and Roberto Iriondo, “The best public Datasets for Machine Learning and Data Science”, 2018 <https://towardsdatascience.com/the-50-best-public-datasets-for-machine-learning-d80e9f030279>

[14] “Machine Learning Repository”, <https://archive.ics.uci.edu/ml/index.php>

[15] Martin Thoma, “The Reuter’s Data Set”, 2017, <https://martin-thoma.com/nlp-reuters/>

[16] Leo Breiman, “RANDOM FORESTS”, University of California, Berkeley, CA, pp7-11, January 2001

[17] Magali R. G. Meireles, Paulo E. M. Almeida and Marcelo Godoy Simões, “A Comprehensive Review for Industrial Applicability of Artificial Neural Networks”, IEEE Transactions on Industrial Electronics, Vol. 50, No. 3, pp587-588, June 2003

[18] “Feed Forward Neural Network”, <https://en.wikipedia.org/wiki/Feedforward_neural_network>

[19] Hassan Ramchoun, Mohammed Amine Janati Idrissi, Youssef Ghanou and Mohamed Ettaouil, “Multilayer Perceptron: Architecture Optimization and Training”, International Journal of Interactive Multimedia and Artificial Intelligence, University Sidi Mohammed Ben Abdellah, Fez, Morocco, Vol. 4, pp 26-27.

[20] “Artificial Neural Network”, <https://en.wikipedia.org/wiki/Artificial_neural_network>

[21] Yuji Roh, Geon Heo and Steven Euijong Whang, “A Survey on Data Collection for Machine Learning: a Big Data - AI Integration Perspective”, Engineering Research Center Program, National Research Foundation of Korea (NRF), Korea, Nov-2018, pp 1-5.

[22] Sally Jo Cunningham, “Dataset cataloging metadata for machine learning applications and research”, Department of Computer Science University of Waikato Hamilton, New Zealand, pp 1-3

# **Annexure I**

## Bag of Words

Bag of Words (BoW) model is a popular document analysis algorithm which is mostly used in text and image classiﬁcation. Bag of words model is based on following two assumptions.

• Documents are created by repetitively drawing one word from a bag of words which forms the vocabulary,

• Words in the bag may occur multiple times in a document.

Figure (below) shows documents generation process using a bag of words vocabulary.

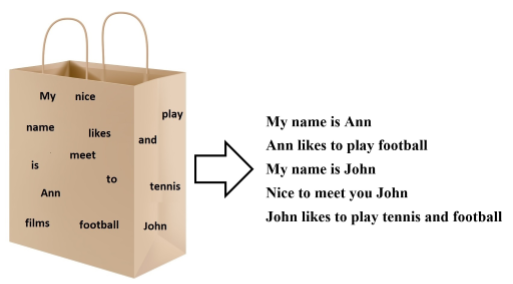


Figure 7. 1: Bag Of Words Used in Document Interpretation

The most general and widely used bag of words model can be described as follows. Let t1,t2,...,tn denote distinct terms used for indexing documents and D1,D2,...,Dm documents. Document Di is represented by a term vector which is defined as:

*Di = (ai1, ai2,…..,ain)T*

where *aij* is a weight of a term *tj* in the document *Di*. The values *aij* can be just simple frequencies of the term *tj* in the document *Di* or they are further normalized.

As an example, considering few simple documents to understand how Bag-Of-Words concept works in real scenario of document comparison;

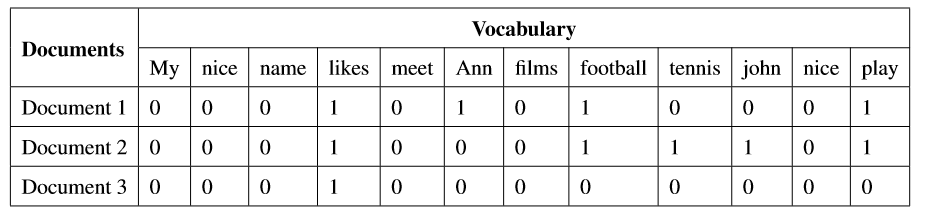
Document 1. “Ann likes to play football.”

Document 2. “John likes to play tennis and football.”

Document 3. “Diana likes swimming.”

Vocabulary that used to compare these documents is shown in Figure (above). There are high frequently used terms such as “is”, “are” and “to” which are not included in the vocabulary.

In practice high frequently used terms such as “is”, “are” and “to” are not included in the vocabulary. BoW model generates a term frequency matrix for each document based on count of words in the vocabulary in each document as shown in below table. Similar documents could be identified by comparing term frequency matrix.

Table 7. 1: Term Frequency Matrix for Example Documents

By analyzing the term frequency matrix, it can be observed that document 1 and 2 are similar to each other while document 3 is diﬀerent.

In above example a global vocabulary is considered. However, diﬀerent vocabularies provide an opportunity to group similar documents together more precisely. Figure 27 illustrates concept of diﬀerent BoW model classes. Words related to sports are grouped in a sports BoW class, words related to business are grouped in a business BoW class and words related to mathematics are grouped in a mathematics BoW class. When a new document is received, based on term frequency matrix of each BoW classes most suitable class for that document is identiﬁed.



Figure 7. 2: Different Vocabularies for Different Document Classes

Consider following example to illustrate this idea further;

* Sports vocabulary ={goal, foul, penalty, football, captain, player}
* Mathematics vocabulary ={calculus, multiplication, subtraction, division, polynomials, curve}

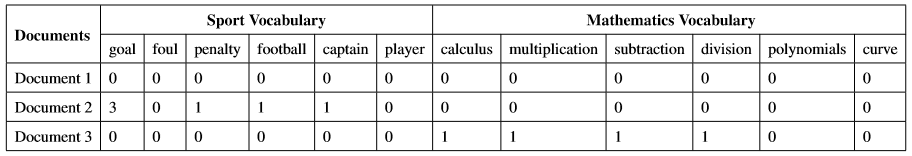
Document 1: “Ann likes to read books.”

Document 2: “Football match between Brazil and Argentina is drawn. Each team scored 2 goals. Brazil captain scored two goals for their team, Argentina misses one penalty goal chance otherwise, they may have won the game.”

Document 3: “Student should be familiar with basic mathematics operations such as multiplication, subtraction and division. Calculus is part of secondary level education.”

Term frequency matrix of document 1, 2 and 3 for mathematical and sports BoW vocabularies are shown in table below

Table 7. 2: Term frequency matrix of documents for mathematical and sports BoW vocabularies



By observing term frequency matrices, it can be seen that document 1 does not belong to either sports or mathematics classes. Document 2 belongs to sports class and document 3 is belong to mathematics class.

# **Annexure II**

## ANN Implementation - Pseudocode

Open the raw dataset

Create a word Dictionary out of all the words in all the documents <word\_dict>

Assign values of each word into the raw data set using KEY:VALUE pair of word dictionary

Assign a value for Train : Test ratio

Segregate Train Data and Test Data in to two different Arrays

Segregate Train Labels and Test Labels in to two different Arrays

Vectorize both Train and Test data separately to represent in binary array

Perform One Hot Encoding for the Classification Labels

Segregate the Train Values in to two (30% as Validation Data and 70% as Training Data)

Segregate the Train Labels in to two (30% as Validation Data and 70% as Training Data))

Define the ANN Model with parameters (Nodes per Input Layer, Nodes per output Layer, Dropout Value, Number of Hidden Layers, Activation Function and Input Shape)

Define Batch Size and Number of Epochs

Train the previously defined ANN Model using Train Values

Evaluate the Model against Validation Data (30% of initial Training Data)

Use Evaluation Results to create Graphs and Check the Accuracy of the Model

Perform the prediction for the initial Test Data set

Evaluate the Performance and Result Accuracy

Create Visualizations of the results