CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Feature Selection for Document Classification

by

Syeda Zarwa Faiz

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Computing
Department of Computer Science

2021

My work is devoted to My Parents, My Teachers, My Family, and My Friends. I have a special feeling of gratitude for My Parents and brothers. Special thanks to my supervisor whose support make me able to reach this milestone.

CERTIFICATE OF APPROVAL

Feature Selection for Document Classification

by

Syeda Zarwa Faiz

(MCS191020)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Muhammad Muzammal	BU, Islamabad
(b)	Internal Examiner	Dr. M. Shahid Iqbal Malik	CUST, Islamabad
(c)	Supervisor	Dr. Abdul Basit Siddiqui	CUST, Islamabad

Dr. Abdul Basit Siddiqui
Thesis Supervisor
May, 2021

Dr. Nayyer Masood
Head
Dept. of Computer Science
May, 2021

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
May, 2021

Author's Declaration

I, **Syeda Zarwa Faiz** hereby state that my MS thesis titled "**Feature Selection for Document Classification**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Syeda Zarwa Faiz)

Registration No: MCS191020

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled “**Syeda Zarwa Faiz**” is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Syeda Zarwa Faiz)
Registration No: MCS191020

Acknowledgement

I praise and worship my Allah who is all in all. He is perfect source of strength in my life.

I want to show gratitude to my family including parents and brothers for their support and love. I would not be able to achieve anything without my family. I am thankful to my teachers. They give me a lot of knowledge.

Specially, I want to thank my supervisor Dr. Abdul Basit Siddiqui for this work. He has guided me in a very good way. I have learned a lot from him.

(Syeda Zarwa Faiz)

Abstract

A bulk of textual documents are available online on the Internet, digital libraries, and news sources, and their amount is increasing enormously day by day. Scientific literature is growing rapidly that it becomes a very crucial process to access relevant information. Most of the data generated today are unstructured which typically consists of text information. Understanding such text data is imperative given that it is rich in information and can be used widely across various applications. Document classification is the task of assigning a document to pre-defined classes, is very important for information organization, storage, and retrieval. In literature, researchers have proposed various techniques for performing single and multiclass classification. The features used for classification are based on metadata and content based approach. Metadata of the research articles are available free while the content of the article is not accessible because the major journals like IEEE, ACM, and Springer, etc have not given access to the overall content of the article. However, the content based approach produces better result as compared to metadata based approach due to the richness of features. In case when the content is not available, researchers have utilized metadata based approach. Researchers have proposed techniques that use statistical measures for textual representation and the semantics of the text is completely ignored. It is found in the literature that researchers try different combination of features and produces their results. This thesis proposed a model for performing multi-class classification of research articles. The model uses BERT for textual representation and it captures the semantics and context of terms. Moreover, the model does optimization of features. An optimal feature selection can play an important role in this area and can improve the overall accuracy of the document classification system. For experimentation, this study uses the JUCS dataset of the computer science domain. The proposed model outperformed the previously proposed techniques. The accuracy achieved is 98.15% for the JUCS dataset.

Contents

List of Figures

List of Tables

Abbreviations

ACM	Association of Computing Machinery
BERT	Bidirectional Encoder Representations from Transformers
BOW	Bag of Word
JUCS	Journal of Universal Computer Science
MLC	Multi-Label Classification
SLC	Single Label Classification
TF	Term Frequency
W2V	Word2Vec

Symbols

X_n	Number of features
C_i	Set of all Categories
D_k	Set of all Documents
C_m	Set of Category

Chapter 1

Introduction

1.1 Background

Researchers are generating huge amount of research articles and the amount of scientific documents are increasing day by day. The process of production of research articles never stopped instead it is continuously increasing day by day. Mostly these documents are available online. One can search these documents over the internet using search engines, digital libraries, and citation indexes. The huge amount of data on the web hinders the recommender systems to extract the research papers which is relevant to the posed query. Typically, users explore different repositories to extract the relevant research papers, such as Digital Library, Google Scholar, etc.

If a user posed a query on the web, it returns millions of generic hits in which some of them are related to a posed query. If user wants to find the related document, it is very difficult to read all those papers or articles. The reason behind this is, the papers over these repositories are not properly indexed according to their respective classes. This extensive disorganization of research articles has grabbed the attention of the research community to classify the research article into the appropriate category. That's why this is an important research area. Researchers have faced such a big challenge to classify the document into appropriate category or categories, due to the presence of such a huge amount of data on the web. Every research article associates with one and more categories. The issue of mapping the research articles with associated categories can assist in multifarious aspects by helping scholars such as 1) Helping researchers to find the relevant documents to their topic 2) Finding relevant literature to narrate the background concept of the proposed study and so on. 3) Search engines and digital libraries returns relevant document for user queries.

The research article classification has mainly divided into two broad categories, 1) Content based approaches and 2) Meta data based approaches. Normally the content based approaches produce good results as compared to metadata based approaches due to the richness of features [1] [2] [3]. However, the main issues with content based approaches is that it is not freely available because the major journal publisher like ACM, IEEE, etc. have not provided the access to the overall content of the research article. In such scenarios, some of the researchers have utilized Meta data as an alternative way for classification of research articles [4] [5] [6]. So, one of the possible substitutes of content based approach is metadata based approach. Meta data is actually data about data. Meta data of the research articles contain title, keywords, general term, authors, categories etc. and mostly these metadata are freely available online.

After reading the limitation of using content based approach that it is not freely available. This study has decided to use content based approach by using JUCS (Journal of Universal Computer Science) [7] whose content is freely available. Content based approach of the research article contains abstract, introduction, headings, and conclusions. This thesis mainly focuses on feature optimization for the classification of research articles.

1.2 Classification

In the classification process, a given dataset is categorized into different classes. The classification is performed on both structured and unstructured data and it predicts the class of a particular record in the dataset. The classes are often referred to as target, label, or categories. The dataset is trained first on the training dataset and then the model predicts on the testing dataset. Then, an unknown data can be predicted easily that in which class it will fall into [8]. The model used for classification requires the use of machine learning algorithms that learn how to assign a class label to record in the dataset. The example for the classification of emails as “spam” or “not spam” [9] is shown in Figure 1.1.

From literature, it is observed that research article classification is very helpful in retrieving relevant documents against a posed query. The research article classification method is mainly composed of selecting the Metadata features which could help in assigning some suitable category to the research articles. Researchers use all features to classify the document but with the increasing number of features, the complexity also increases. So this study focuses on optimizing the features in such a way that with less number of features it can predict higher accuracy. Mainly, there are two types of classification. 1) Single label Classification (in which an item belongs to only one class, while there are two or more class’s available), 2) Multi-label classification (in which an item belongs to more than one class). However, a single research article could belong to more than one category, this aspect diverts the attention of the research community towards the multi label classification of research articles. Most of the existing approaches produce low accuracy and authors have used a few numbers of categories. That’s why, this is an open research area to classify the document into multiple categories with high accuracy. This research work mainly focuses on multi-label classification by using metadata and content features of research articles.

1.3 ACM Classification System

ACM stands for Association for Computing Machinery. It is a society for computing and was founded in 1947. It is the world’s largest scientific society for computing. In 1964, ACM’s first classification system for the computing field was published. The entirely new system was then published in 1982. Then, its different versions were published in 1983, 1987, 1991, and now 1998 [10]. The version 1998 has served as the de facto standard classification system for the computing field

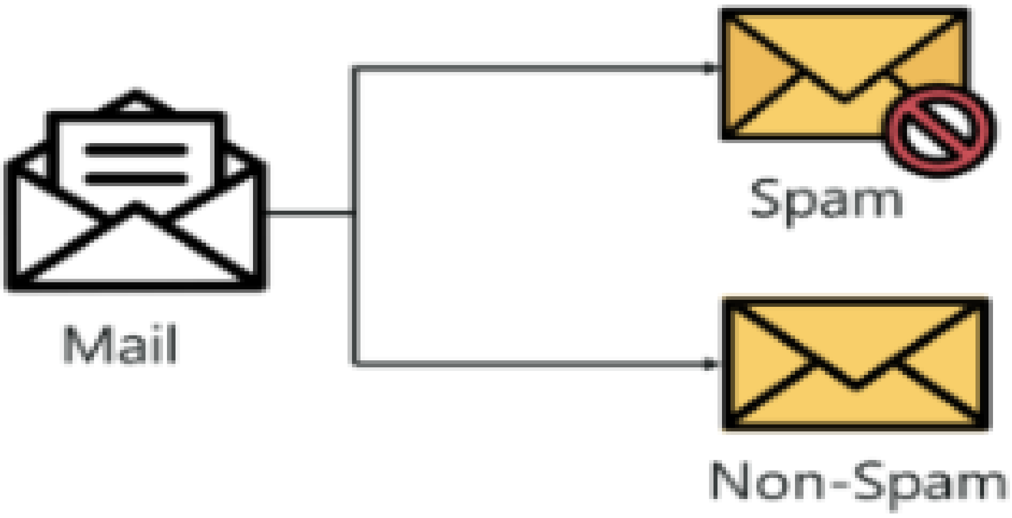


FIGURE 1.1: Classification of Email [9].

[11]. Then, in 2012 ACM Computing Classification System has been developed as a poly-hierarchical ontology and the 1998 version of ACM CSS was replaced by it. There are three levels in ACM CSS. The first level contains topics from A (General Literature) to K (Computing Milieux) and it total count is 11. In the second level, every topic of the first level have subtopics. for example, first level topic A (General Literature)", their subtopic is A.0 (General), A.1 (Introductory and Survey), ..., A.m (Miscellaneous) topics and a total count of the topic in the second level are 81. At the third level, every second level has further subtopics. For examples such as A.0.0 (Biographies / autobiographies), A.0.1 (Conference proceedings), ..., A.0.2 (General literary works),.., C.2.m (Miscellaneous), and a total count of a third level topic is 400.

1.4 JUCS Classification System

In the computer science domain, in literature, most of the conferences and journals have utilized the ACM classification hierarchy to categorize their research articles into different categories. This thesis points toward research document classification in the Computer Science domain by utilizing the metadata and content of research articles. But most of the journals have not given access to their content.

Table 1.1: JUCS Classification System.

Category	**Description**
A	General Literature
B	Hardware
C	Computer System Organization
D	Software
E	Data
F	Theory of Computation
G	Mathematics of Computing
H	Information Systems
I	Computing Methodologies
J	Computer Applications
K	Computing Milieux
L	Science and Technology of Learning
M	Knowledge Management

The content of the Journal of Universal Computer Science (JUCS) is freely available. It is the finest computerized publication and it covers the area of the computer science domain. It was founded in 1994 and it is appearing on monthly basis since its foundation. The content and metadata features can be extracted easily from J.UCS because it is an open access journal. The classification system of JUCS follows ACM classification system. The root-level categories of J.UCS are shown in Table 1.1.

1.5 Text Representation

Most of the data generated today are unstructured which typically consists of text information. Understanding such text data is imperative given that it is rich in information and can be used widely across various applications. However, the key to understanding such data is its representation. The research articles or documents are in textual form. Text representation is the basic problem in text mining and retrieval of information [12]. The text document is unstructured and it aims to represent the text into vector form which is a numeric representation of text and can be computed mathematically. To represent the text document in numeric form, numerous techniques have been proposed and used in literature like Bag of Word (BOW), Term Frequency (TF), Term Frequency, and Inverse Document frequency (TFIDF), etc. All of these rely on the frequency of terms and they have ignored the semantic and contextual meaning of terms. The latest approaches [1] [2] [3] [4] [5] [6] for research article classification have employed these conventional statistical measures like TF, BOW, and TFIDF, etc. due to which they have ignored the semantic and contextual information of terms and it might be assigned a wrong category to the research articles. This study focuses on the textual representation of the dataset before doing optimization. The technique used in this thesis considered the semantic and contextual meaning of terms. In literature, there are different semantic techniques for representation. One of the most well-known techniques which is used in different domains is word embedding [13] [14] [15]. The distributed representations of words learned by neural networks and their applications. Distributed word representations are called word embedding. For this, Word2Vec is one of the most popular techniques to learn word embeddings. For natural language processing, it is one of the technique.

A great technique was created by researchers at Google. The technique named as World2Vec was published in 2013 and is led by Tomas Mikolov. For training, the algorithm uses a neural network. The model is trained on a large corpus of text. After the training phase is over, the model can detect synonymous words or suggest additional words for a partial sentence. It represents every word with a vector. It considered the semantics of the text. The semantic similarity between the words can be represented by the cosine similarity between the vectors. For converting text into vectors it reads text from only one side i.e; from left to right and this is basically the disadvantage of word2vec that it is one way means it reads text from left to right and not the other way round. This thesis uses BERT for text representation. BERT stands for Bidirectional Encoder Representations from Transformers [16]. It is a machine learning technique based on transformers and is used for natural language processing (NLP). It is already pre-trained by

Google. BERT was created by Jacob Devlin and his colleagues from Google and it is published in 2018. The Bert model is already trained and there are two basic models: the BERT base model and the BERT Large model. Both models have different layers, hidden-units, heads etc. One can use any model depending on its requirements.

The description of both the models are given as :(1) the BERT BASE model, which has 12-layer, 768-hidden units, 12-heads, 110M parameter neural network architecture, and (2) the BERT LARGE model, a 24-layer, 1024-hidden units, 16-heads, 340M parameter neural network architecture. Both models are pretrained and their pre-training is done on two things i.e; both were trained on the BooksCorpus with 800M words and a version of the English Wikipedia with 2,500M words. One of the advantages of using Bert is it is two way means it reads text from both sides. The detailed working of this model is explained in chapter 3.

1.6 Feature Optimization

This study uses a machine learning algorithm for feature optimization because with more features its complexity increases. An optimal feature selection can play a vital role in this area and can improve the overall accuracy of the document classification system. One of the most modern algorithms for feature selection is the genetic algorithm [17].

It follows the natural phenomenon of biological evolution. It is a stochastic method for feature optimization. Organisms have genes that evolve over a period of time or over successive generations. By evolving they can adapt themselves better in the environment. It is a heuristic optimization method inspired by the procedures of natural evolution. The population of individuals is created by the algorithm. A new population is created after every generation by selecting the individuals who are most fitted in the problem domain. Then the individuals are recombined and different operations are performed using different operators like mutation, Crossover.

1.7 Problem Statement

Problem statement is as follow:

Feature selection plays an important role in classification/ prediction systems. For the task of document classification, it is significant to select those features which improve classification results. Currently, different techniques use metadata and content-based features and feature selection within each category is not clear. An optimal hybrid feature selection using metaheuristics needs to be defined to find a subset of related features in the area of document classification.

1.8 Research Questions

The following research questions are formulated relying on the problem statement describe above:

1. How feature selection can be helpful for better document classification?
2. How features can be represented using semantics and contexts?
3. How Metadata and Content based features can be combined for document classification?
4. How a metaheuristic can be used to perform feature selection?

1.9 Purpose

Feature selection plays an important role in classification systems. For the task of document classification, it is significant to select those features which improve classification results. The main objective of this study is to provide an optimal feature selection using metaheuristic for the classification of research articles and to use metadata and content of a research article for classification and also to identify whether a Semantic model (use for text representation) is helpful in the classification of research papers into predefined classes or not.

1.10 Scope

This thesis focuses on mapping research articles belonging to the Computer Science domain into J.UCS category or categories at the root level. The root level categories are from A to M. The description of categories of J.UCS are shown in the Table 1.1. Moreover, for Computer Science research articles we have used J.UCS [18] dataset. The reason for the selection of this datasets is that it contains research articles whose content is available freely and it covers the area of the Computer Science domain. Because this study needs both metadata and content based features. BERT is used for the conversion of text to numeric values and performs feature optimization by using a machine learning algorithm to improve the classification accuracy.

1.11 Significance of the Solution

This research will contribute to classify the scientific documents (research papers) to the pre-defined J.UCS classification system and perform optimization of features to reduce the complexity. It is advantageous for a number of systems such as: retrieving the information, analyzing trends, finding experts, recommendation systems, search engines, and citation index. If the classification system is accurate, it is helpful for the authors of research papers in their paper submission process. Authors can find the category of their research contributions. Conference/Journal paper submission systems can assign categories to the research papers and assign reviewers to those papers. Researchers can easily find required papers if there is an accurate classification system maintained for all research papers.

1.12 Definitions, Acronyms, and Abbreviations

- Journal of Universal Computer Science (JUCS)
- Association of Computing Machinery (ACM)

- Term Frequency (TF)
- Word2Vec (W2V)
- Bidirectional Encoder Representations from Transformers (BERT)
- Single Label Classification (SLC)
- Multi-Label Classification (MLC)
- Bag of Word (BOW)

Chapter 2

Literature Review

The problem statement and research questions are explained clearly in chapter 1. This chapter focuses on critical analysis of all state-of-art-approaches, as every research study is dependent on the previous study that has already been performed in this field. The research community of document classification has proposed a number of new ideas for document classification and as we know the number of research documents is increasing on daily basis. After that, the attention of the research community moved towards research paper classification due to rapid invention in literature. The state-of-the-art proposed approaches in literature can be divided into two broad categories.

1. Content based approaches
2. Metadata based approaches

The Content based approaches have mostly focused on the overall content of the research article and it contain title, keywords, author name etc, and are available freely while the metadata based approaches have focused on metadata of the research article and it contain introduction, headings, methodology and conclusion and are not available freely. This is the reason why researchers mostly move towards metadata features instead of content based features due to the subscription requirement. Both are explained in detail in the below sections.

2.1 Metadata Based Approaches

The existing metadata based approaches use the metadata of research articles for classification of research document task. For example, the information about the metadata of the article is maintained by the ACM digital library [19]: author name, the title of the article, name of a journal, the article issue date, its volume number, the publication month, its year, and the page number, URL, DOI of the paper, etc. These elements can be manually extracted and can be extracted automatically from the research article. This type of metadata is almost freely available, while the whole content of the data is not freely available online. To get access to the content of the article subscription is needed because the major journals like ACM, IEEE have not given access to overall content of the article. So that is the big motivation for the research community to move from content to freely available metadata of the research documents. Now in this section, some metadata-based approaches are briefly highlighted.

Yohan et al. [20] proposed that Named Entity recognition (NER) helps the machine to identify named entities in text and classified them in their respective categories. The approach generates rules for identifying name entities and their classification, specifically for the Teluge language. For recognizing name entities and their classification, the approach exploits word, work lookup, and contextual-based features. The approach has comprehensively been evaluated using different Newspaper and Teluguwiki datasets. Moreover, the evaluation has been performed using full sentences.

Swapnil et al. [21] proposed a document classification algorithm and is based on Latent Dirichlet Allocation (LDA) [22]. The dataset is not labeled because it does not require a labeled dataset. In this dataset, there are different class labels and they build a model in which they assign one topic to one of the class labels. Dirichlet has an aggregation property, they combine all the same class label topics into a single topic. When the new unlabeled document comes for prediction, it finds the similarity or “closeness” to one of the aggregated topics. The algorithm is extended by combining the Expectation-Maximization (EM) algorithm and a naive Bayes classifier.

Khor and Ting [23] proposed a Bayesian-based approach to classify research papers. In this study, the dataset consists of 400 conference research papers and mapped to four different classes including e-learning, cognition issues, teacher education, and intelligent tutoring system. They split the dataset into two parts as 80-20 ratio. They have extracted keywords from 80 percent of the papers. The other 20 percent is used to predict the performance of accuracy. They extract the keywords using the features selection algorithm.

Godbole [24] proposed a method based on multi-label classification. They present an approach in which they combine the textual features and features that shows the relationship between classes. The dataset used for this study is based on real-world text. They used support vector machine for performing classification task. They do enhancement in svm model for building better models. This new method performs best as compared to the previous method.

Sajid et al. [25] proposed a fuzzy logic-based classifier for the classification of research paper. The research papers belong to the Computer Science domain. As the paper does not belong to only one class, therefore, they used fuzzy logic, and proposed fuzzy-based rule merger algorithm to merge the generated rules and fuzzy classifier to classify the paper into respective single and more categories. For experimental purpose, they have selected the JUCS datasets because its covers all areas of Computer Science domain. From this dataset, they have extracted title and keywords, which were used as a feature for papers classification. After performing a detailed evaluation of the approach, the results revealed that the approach achieved 0.93 precision and 0.96 F measure and they have used single label classification measures.

Ali and Asghar [26], proposed multi-label scientific Document Classification based on metadata features. The approach utilized two metadata features (title and keywords). For performing multi-label classification the approach first converts the data into single label classification by using four different conversion techniques (Min, Max, Ran, and Single). They also used different similarity measures for finding the relevancy between documents and labels. They have used PSO based classifier for the classification of documents. The technique has been evaluated on two different dataset of research articles (JUCS and ACM). The outcome of the study revealed that their approach achieved accuracy up to 0.78.

Riaz et al. [27] proposed Pattern Analysis of Citation-Anchors in Citing Documents for Accurate Identification of In-Text Citations. The in-text citation identification is the vast research area for the researchers. In literature, there is an automatic identification of in-text citation and when they perform experiments on it, they achieved accuracy 0.58. But there are various problems in it, this research work study previous techniques very well and find out the main problems in them. For experimentation of their proposed approach, they used a comprehensive dataset. This paper proposed a taxonomy that is based on the heuristics of previous studies. The dataset used for this model is JUCS and CiteSeer. This model achieved F-Score 0.97 which is too good as compared to previous techniques.

2.2 Content Based Approaches

Content based approaches mostly depend on the content of research articles. This is due to the fact that content includes abundance features. This section elaborates the state-of-the-art content-based approaches:

In 2015, Le et al. [28] performed a survey on all existing feature selection approaches for text classification. There are two main problems of feature selection filtration which are: 1) the computation of the feature score and 2) the categories are not balanced in it. This paper focuses on these problems. This paper analyzes two filter feature selection approaches. One of them is based on the frequency and the other one is based on cluster. They find out the weakness and strength of these approaches and proposes a feature selection method so that the performance of the document classification can be improved.

In 2016, Zhou et al. [29] proposes a model in which a classifier can automatically categorize Computer Science (CS) papers based on text content. The dataset used for the experimentation is CiteSeerX and arXiv. These datasets are labeled. Naive Bayes and Logistic Regression approaches are used for the experimentation of this approach. The scheme for selecting features is also different. They use different models of language and use different feature weighting schemes. They do experimentation by using Bi-gram modeling and the weights of the features are also normalized. It performs very well. The results obtained after experimentation shows that arXiv dataset achieved F-score 0.95, while CiteSeerX has a lower F − score 0.764. This shows that the labeling of arXiv is more accurate than CiteSeerX.

In 2015, Zhong et al. [30] proposes a method for selecting features. The proposed model extract features from the research article which has discriminative power. Then the similarity between the features and the research articles is computed based on their semantic similarity. The dataset used for experimentation is two datasets, viz. Reuters-21578 and 20-Newsgroups which are published. The classifier used for the classification is support vector machine (SVM) classifier. The results obtained after performing experimentations show that the proposed feature selection model performs very well than previous techniques.

Sajid et al. [31], have proposed an approach for research paper classification based on the references section of a research paper. The approach exploits the references segment of a research article to locate the topics of the paper. The study follows an assumption that most of the time, authors cite the articles belonging to the same domain or similar category. To validate the claim, the authors have employed a data set from the Journal of Universal Computer Science (JUCS). They selected this dataset because it covers all the Computer Science areas. In this approach, the stored references in the database have been matched with the extracted references of the paper. After performing experiment, the authors reported their accuracy up to 0.70.

The scientific community has mainly focused on content-based approaches. The main reasons of their attention towards content-based approaches is that it contains a lot of features and produced such remarkable results. However for the application of these approaches, one must have the availability of the content of the research articles which is not possible all the time because famous digital libraries like ACM, IEEE, and Springer provide subscription based services. So researchers have used alternative ways of using Meta data based features. Because Metadata of the research articles are available freely and there is no subscription requirement.

2.3 Evaluation Criteria

The conclusion derived after studying literature related to document classification is, there are different types of data, different classification algorithms, different datasets, different types of classification used.

- **Type of data:** Different data sources are used for classification, data sources are of two type's metadata and content of documents.
- **Type of Classification:** There are two types of classification used; binary class and multi class classification.
- **Datasets:** Different datasets are used by the approach and also presented the quantity of dataset.
- **Classification algorithm:** Different type of algorithm or methodology is used by the approach for the classification of documents.
- **Results:** After performing experiments, the results are presented.
- **Evaluation Measures:** There are different measures for evaluating the performance of a classifier like precision, recall, f-score and accuracy.

2.3.1 Analysis

After studying literature in detail, this study concluded that all the above mentioned approaches use content and metadata of the documents. So, data sources are divided into two broad categories.

1. **Content based approaches:** The content based approaches uses the overall content or text of the document and classifies the documents into respective single or multiple classes from which they belong.
2. **Metadata based approaches:** The Metadata based approach uses the metadata of the research documents and classifies the documents into respective single or multiple classes from which they belong.

2.3.2 Critical Analysis of Literature Review

In the content based approach the overall content of the document is utilized while in the metadata based approach, the metadata of the article is utilized. The metadata of the article is available freely while the content of the research article need subscription for accessing the data.

There are different datasets used by the researchers in literature like:

- Newspaper
- Teluguwiki
- ACM
- JUCS
- CiteSeerX
- arXiv
- Reuters-21578
- 20-Newsgroups

There are different algorithms used for the classification of research articles like:

- Expectation Maximization (EM)
- A Naive Bayes Classifier
- SVM
- Fuzzy Based Rule Merger
- PSO based classifier
- Logistic regression

The brief overview of most related techniques to our work proposed in literature is give in Table 2.1.

Table 2.1: Critical Analysis of Literature Review

Paper	Dataset, Classification types, Types of Data	Representation Technique, Algorithm	Result	Limitation
Multi-label Scientific Document Classification (2018)	JUCS (1460), ACM (86116) Multi-Class Meta-data	TFIDF, Static Threshold, Similarity Measures PSO based Classifier	Accuracy 78.79% 77.86%	Use statiscal features, Static Threshold, Cannot capture the meaning and contextual information of the terms
Pattern Analysis of Citation Anchors in Citing Documents for Accurate Identification of In-Text Citations (2017)	JUCS(1240), CiteSeerX (20,000) Multi-Class Meta-data	Rule based Repository, Heuristic based System	Average F-Score 0.97	Limited to Meta-data i.e. Cited-documents and Cited-by documents

Continued on next page

Table 2.1 – continued from previous page

Ref	Dataset, Classification types, Types of Data	Representation Technique, Algorithm	Result	Limitation
Multi-label Classification of Computer Science Documents using Fuzzy Logic (2016)	JUCS(1460), Multi-Class Meta-data	TF, Fuzzy based rules merger (FBRM) algorithm	Accuracy 91%	Limited to metadata i.e. title and keyword
Exploiting reference section to classify paper's topics (2011)	JUCS (1460) Multi-Class Classification Content-based	TFIDF, Static Threshold, Citation based category identification	Accuracy 70%	Limited to reference section, Use Static Threshold, Cannot capture the meaning and contextual information of terms

After the critical analysis of the research articles related to the document classification domain, both metadata and content based features were utilized. It was observed that some authors utilized the metadata of the documents but most of the authors have employed the contents of the documents. In the Content-based technique, the main focus is on the overall content of research papers like abstract, introduction, headings, methodology, literature, conclusion, etc. The Content of the research article plays a significant role in the implementation of different techniques. A lot of features can be extracted from the content of the paper which produces results with good accuracy. But there is a limitation with content based approach that most of the time content of the paper is not freely available. A large-scale analysis conducted in a study [32], reported that the analysis of the latest year (2015) has 45% percent of Open Access (OA) articles. Journals of all major publishers like IEEE, ACM, Elsevier, Springer, and IOS do not provide open access to their articles. There are financial, legal, and technical barriers to access the content of the paper.

The other is metadata based approach which is utilized by a few numbers of researchers because Metadata does not contain a rich number of feature. It mostly produces low accuracy as compare to content-based approaches. However, some of the Metadata play a pivotal role in classifying research articles such as, title, author name, keywords, general terms, etc each of these metadata serves a specific purpose that can be exploited to classify research papers into a different number of categories. One of the advantages of this approach is, Metadata of the research articles is freely available in many digital libraries like IEEE, ACM etc.

The research articles are in textual form. For performing the classification of research articles, text representation is an important task because the input to classification algorithm is in the form of vectors. So, there is a need to do vectorization of textual data. The current state-of-the-art depict that most of the existing studies have employed conventional statistical measures like TF, BOF, TFIDF, etc. These measures count the frequency of terms. But this study argues that the semantics of a text must also be considered which is ignored by existing statistical measures. For understanding the semantics of text various techniques have been proposed. World2Vec is one of the technique which considers the context and semantics of terms but there is a limitation that it is one way it reads text only from one side, not the other way round. Further, in the area of research papers classification, most of the approaches perform single label classification and there exist a very few approaches that have performed Multi-label classification. With the increasing number of features its complexity increases, so feature optimization is an important task for document classification. In literature, almost all the researchers have performed a manual combinations of features instead of doing optimization.

This study aims to classify research articles using metadata and content as individual features as well as their possible combination. For text representation, BERT model is used to store information semantically and contextually. For comparing our technique we have used Ali and Asghar [26]. They have performed multi-label classification using ACM hierarchy. They used ACM and JUCS datasets. They performed multi-label classification. Their approach scores up to 78% on the JUCS dataset and 77% on the ACM dataset. This study focuses to use Metadata and Content individually as well as its combination for the classification of the research document and use a semantic model for text representation and also perform feature optimization to achieve a good result as compare to Ali and Asghar [26].

Chapter 3

Research Methodology

3.1 Introduction

The critical analysis of already proposed approaches in literature review dictates that the research article classification community has proposed different techniques to classify the research articles into single and multiple categories. The primary observations from the literature review that motivated and signify the proposed framework are as follows: 1) As per our knowledge of literature, there does not exist any study that has comprehensively evaluated the freely available content and metadata individually and its possible combination, 2) there does not exist any study that has used semantic model for text representation that considers context and semantic of a term, 3) there does not exist any study that has done feature optimization using metaheuristic to reduce complexity. These observations have led us to propose a technique to address the issues discussed above. The proposed framework performed classification of research articles into a predefined JUCS classification system by comprehensively evaluating the metadata and content based features. Moreover for understanding contextual meaning of terms, BERT is used which is a bidirectional representation. Then feature optimization is done using metaheuristics. In this chapter, the proposed methodology is described for the classification of research articles. The Figure 3.1 shows the graphical representation of the proposed technique.

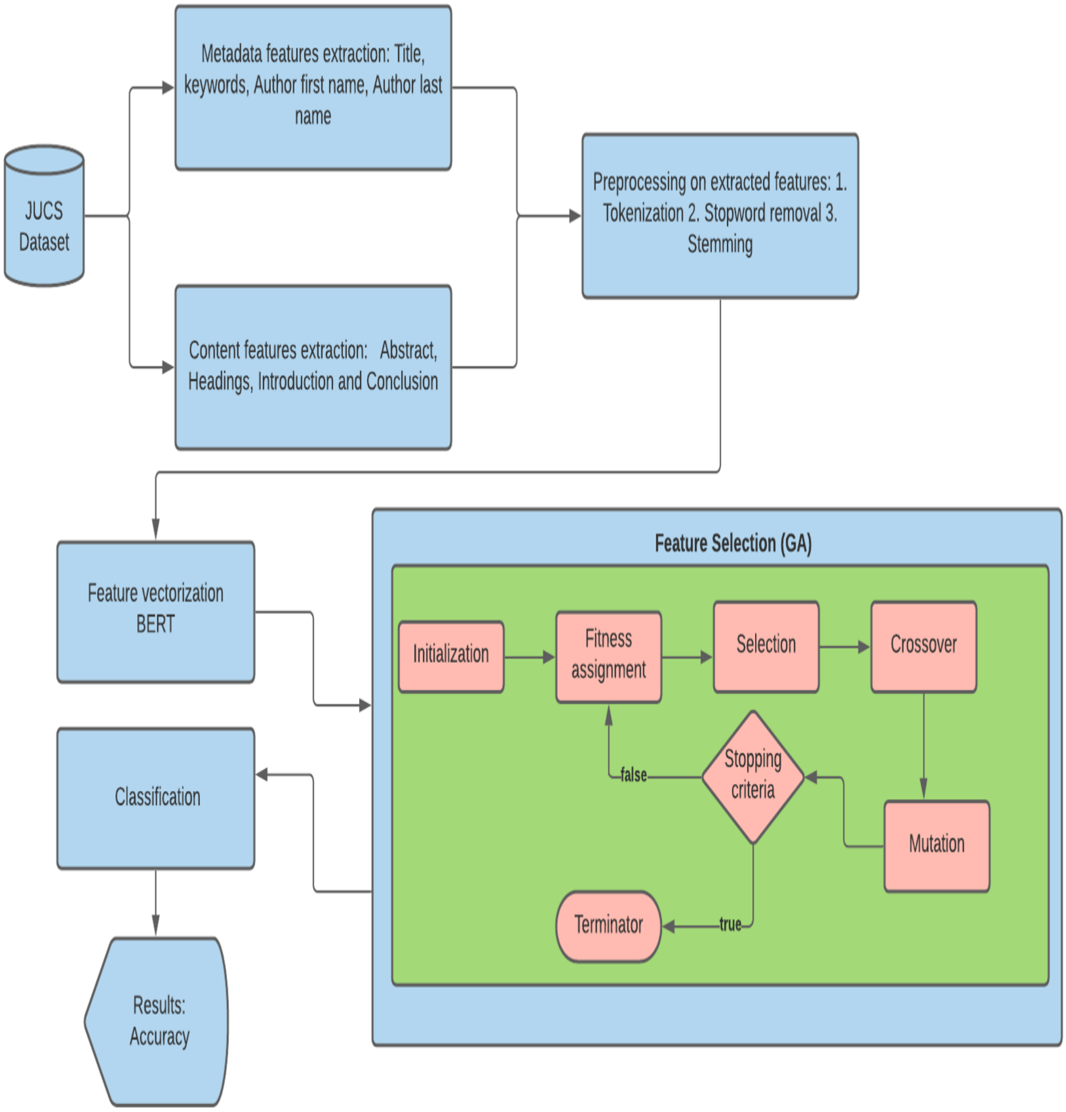

Metadata features extraction: Title, keywords, Author first name, Author last name
JUCS Dataset
Preprocessing on extracted features: 1. Tokenization 2. Stopword removal 3. Stemming
Content features extraction: Abstract, Headings, Introduction and Conclusion
Feature vectorization BERT
Feature Selection (GA)
Initialization
Fitness assignment
Selection
Crossover
Stopping criteria
false
Mutation
Classification
Terminator
true
Results: Accuracy

FIGURE 3.1: Methodology diagram of Proposed Solution

3.2 Dataset

For a comprehensive evaluation of the proposed system, the selection of datasets is a very crucial step. To evaluate the proposed technique, diversified dataset is carefully selected that contain research articles of the Computer Science domain. It contains research articles from the Journal of Universal Computer Science (JUCS) [18]. The reason behind the selection of the JUCS dataset is that it contains papers from multiple areas of the Computer Science domain which plays a significant role in comprehensive evaluation. A detailed description of the dataset is given in the next section.

Table 3.1: JUCS Dataset.

Features	**Records**
Total number of Research Papers	1460
Single Label Research paper percentage	51 %
Multi Label Research paper percentage	49 %
Total number of Classes at root level	13
Metadata and Content of a Research paper	Title, Keyword, Abstract, Author First name, Author Last name, Introduction, Headings, Conclusion

3.2.1 JUCS Dataset

The dataset of JUCS contains almost 1460 papers of 13 different categories of Computer Science domain. JUCS has extended the categories of ACM from 11 to 13 by adding L (Science and Technology Learning) and M (Knowledge Management) categories. Therefore, its root level contains 13 categories. For the evaluation of the proposed technique, the data of all categories are selected from the dataset for classification of research articles. The 51 % of data in dataset has been classified into a single label and 49 % of the data has been classified into multi-label data. The detailed statistics of a dataset are presented in Table 3.1. This study has discussed that a document can have multiple labels which in Figure 3.2.

Formally it is formulated as:
Each category has a set of documents where C is the set of all categories while D is the number of documents.

$$\mathcal{C}_i = \Sigma_{k=1}^{n} \mathcal{D}_k \tag{3.1}$$

Each document contains different features

$$\mathcal{D}_k = \Sigma_{n=1}^{i} \mathcal{X}_n \wedge \Sigma_{m=1}^{j} \mathcal{C}_m \tag{3.2}$$

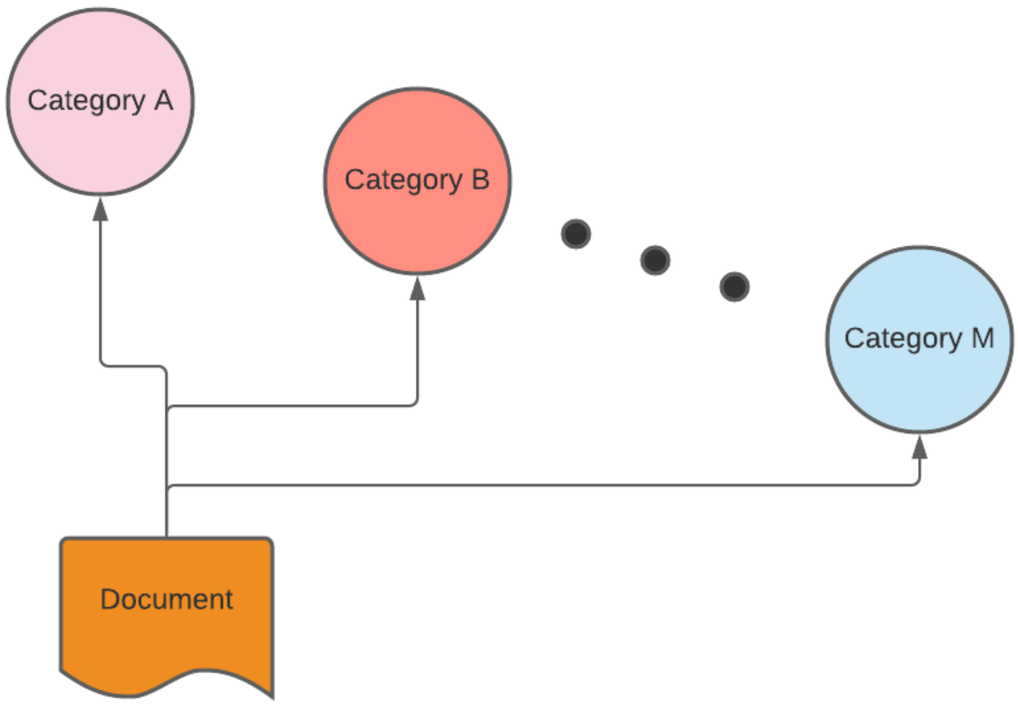


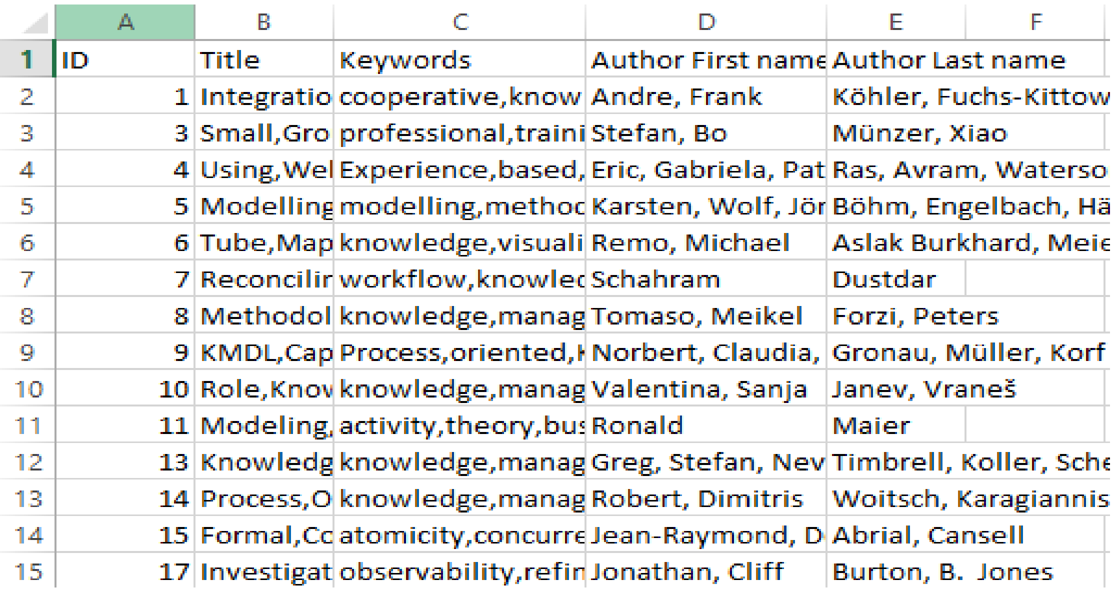
FIGURE 3.2: Document Example.

3.3 Feature Extraction

The JUCS dataset contains metadata and content of research articles. From this dataset, metadata of the article like title, keywords, author first name, and author last name have been extracted. The content of the article like abstract, Introduction, Headings, and Conclusion have also been extracted. The selection of these specific metadata and content based features is due to the reason that they all hold potential terms that can assist in determining the category of the research article. The extracted metadata and content based features are shown in Figure 3.3 and Figure 3.4 .

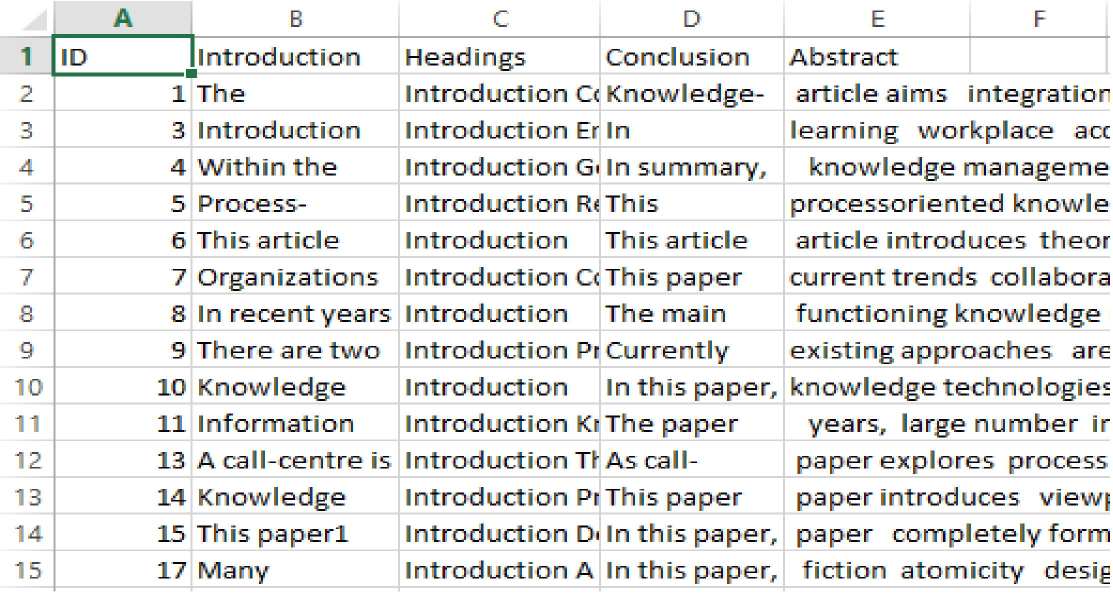
3.4 Pre-processing

Preprocessing is a data mining technique that involves transforming the dataset into its own understandable format. Generally, datasets are incomplete: lacking attribute values (Missing Value), containing noisy data (meaningless data), etc.



	A	B	C	D	E	F
1	ID	Title	Keywords	Author First name	Author Last name	
2	1	Integratio	cooperative,know	Andre, Frank	Köhler, Fuchs-Kittow	
3	3	Small,Gro	professional,traini	Stefan, Bo	Münzer, Xiao	
4	4	Using,Wel	Experience,based,	Eric, Gabriela, Pat	Ras, Avram, Waterso	
5	5	Modelling	modelling,metho	Karsten, Wolf, Jör	Böhm, Engelbach, Hä	
6	6	Tube,Map	knowledge,visuali	Remo, Michael	Aslak Burkhard, Meie	
7	7	Reconcilir	workflow,knowle	Schahram	Dustdar	
8	8	Methodol	knowledge,manag	Tomaso, Meikel	Forzi, Peters	
9	9	KMDL,Cap	Process,oriented,k	Norbert, Claudia,	Gronau, Müller, Korf	
10	10	Role,Kno	knowledge,manag	Valentina, Sanja	Janev, Vraneš	
11	11	Modeling,	activity,theory,bu	Ronald	Maier	
12	13	Knowledg	knowledge,manag	Greg, Stefan, Nev	Timbrell, Koller, Sche	
13	14	Process,O	knowledge,manag	Robert, Dimitris	Woitsch, Karagiannis	
14	15	Formal,C	atomicity,concurre	Jean-Raymond, D	Abrial, Cansell	
15	17	Investigat	observability,refi	Jonathan, Cliff	Burton, B. Jones	

Figure 3.3: Metadata based Features.



	A	B	C	D	E	F
1	ID	Introduction	Headings	Conclusion	Abstract	
2	1	The	Introduction C	Knowledge-	article aims integration	
3	3	Introduction	Introduction Er	In	learning workplace acc	
4	4	Within the	Introduction G	In summary,	knowledge manageme	
5	5	Process-	Introduction R	This	processoriented knowle	
6	6	This article	Introduction	This article	article introduces theor	
7	7	Organizations	Introduction C	This paper	current trends collabora	
8	8	In recent years	Introduction	The main	functioning knowledge	
9	9	There are two	Introduction P	Currently	existing approaches are	
10	10	Knowledge	Introduction	In this paper,	knowledge technologies	
11	11	Information	Introduction K	The paper	years, large number ir	
12	13	A call-centre is	Introduction T	As call-	paper explores process	
13	14	Knowledge	Introduction P	This paper	paper introduces viewp	
14	15	This paper1	Introduction D	In this paper,	paper completely form	
15	17	Many	Introduction A	In this paper,	fiction atomicity desig	

Figure 3.4: Content based Features.

For preprocessing, this study has performed different steps such as 1) Tokenization 2) Noise Removal 3) Stop word's Removal 3) Stemming. Let us discuss these steps one by one:

3.4.1 Tokenization

Tokenization is the first step of preprocessing. In this process, text can be divided into a set of meaningful pieces. These pieces are called tokens. For example, a chunk of text is divided into words, or sentences. Depending on the task at hand, one can define own conditions to divide the input text into meaningful tokens. In this scenario, the sentences are divided into words. For this, the study have used the Natural Language ToolKit (NLTK), which is the best known and most used Natural language processing (NLP) library [33].

3.4.2 Noise Removal

Removing noise from data is important because it can adversely affect the accuracy. Generally, the datasets contain noise such as: 1) Null values 2) Unnecessary punctuation. There exist various methods to remove noise such as 1) Ignoring the missing record, 2) Filling the missing values manually, 3) Filling using computed values. There is a very limited number of instances that contain missing values however, these instances are ignored as it is the simplest and efficient method for handling the missing data. After tokenization, some of the punctuations can be considered as tokens which can be unnecessary (not meaningful) and can misguide us.

Therefore, this study has removed all of these unnecessary punctuations by using NLTK library. After doing all this, this study has deleted records that contain editor columns and special issue papers. Some papers don't have introduction and conclusion so we also deleted those records. The details are shown in Table 3.2.

3.4.3 Stop Word's Removal

Stop words are the most common words in a language such as top 25 stop words are (a, an, and, are, as, at, be, by, for, from, has, he, in, is, its, of, on, that, the, to, was, were, will, with). These words do not carry important meaning so they must be excluded from the document to achieve accurate measurement. Therefore, it is important to remove these stopwords. To remove stop words from all data parameters of a dataset, we have used the NLTK library because it contains a list of stop words. NLTK matches its list of stop words with the tokenized list and then performed stop word removal from the corpus.

Table 3.2: JUCS Papers Detail.

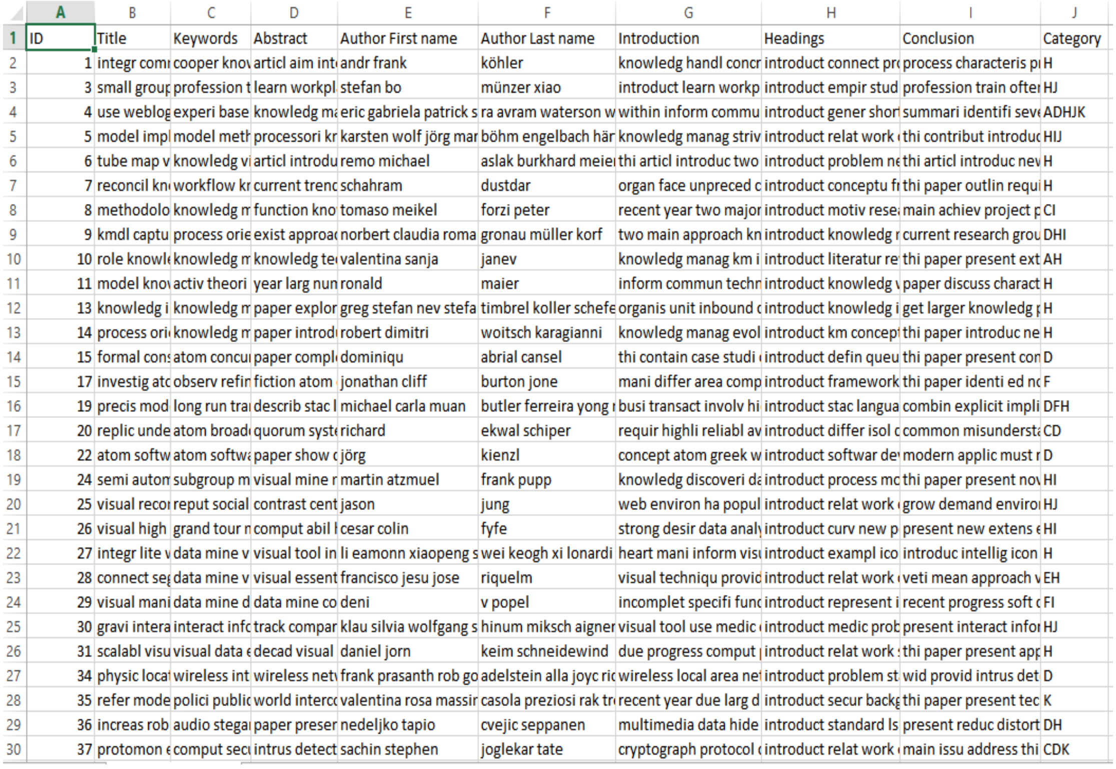
Types of Papers	**Records**
Special issue papers	107
Managing editor columns	90
Papers that don't include introduction	10
Papers that don't contain conclusion	187
Papers whose content are not available	33
Total number of papers excluded	417
Papers whose categories are not specified	94
Remaining paper for experimentation	939

3.4.4 Stemming

Stemming is the process of reducing the words to their base or root words. The advantage of stemming is that it reduces the size of the vocabulary. For example all these words, "consult", "consultant", "consulting", "consultative", "consultants" and "consulting", are stemmed into their root word "consult". This study has been performed stemming by using the porter stemmer algorithm (Porter, 1980), which converts all the terms of a text into their root terms. The stemming algorithm is applied to all the data of the dataset. The preprocessed dataset is shown in Figure 3.5

3.5 Text Representation

Most of the similarity measures and machine learning algorithms often take numeric vector as an input. However, before performing any operation on a text, we need a way to convert each document into numeric vectors. This is one of the fundamental problems in data mining, which aims to numerically represent the unstructured text documents to make them mathematically computable. For this numerous techniques have been presented in the literature. These techniques



	A	B	C	D	E	F	G	H	I	J
1	ID	Title	Keywords	Abstract	Author First name	Author Last name	Introduction	Headings	Conclusion	Category
2	1	integr com	cooper kno	articl aim int	andr frank	köhler	knowledg handl concr	introduct connect pr	process characteris p	H
3	3	small grou	profession t	learn workpl	stefan bo	münzer xiao	introduct learn workp	introduct empir stud	profession train ofte	HJ
4	4	use weblo	experi base	knowledg m	eric gabriela patrick s	ra avram waterson w	within inform commu	introduct gener shor	summari identifi sev	ADHJK
5	5	model impl	model meth	processori kr	karsten wolf jörg mar	böhm engelbach här	knowledg manag striv	introduct relat work	thi contribut introduc	HIJ
6	6	tube map v	knowledg vi	articl introdu	remo michael	aslak burkhard meie	thi articl introduc two	introduct problem ne	thi articl introduc nev	H
7	7	reconcil kn	workflow kr	current trenc	schahram	dustdar	organ face unpreced c	introduct conceptu fr	thi paper outlin requi	H
8	8	methodolo	knowledg m	function kno	tomaso meikel	forzi peter	recent year two major	introduct motiv resea	main achiev project p	CI
9	9	kmdl captu	process orie	exist approa	norbert claudia roma	gronau müller korf	two main approach kn	introduct knowledg r	current research grou	DHI
10	10	role knowle	knowledg m	knowledg te	valentina sanja	janev	knowledg manag km i	introduct literatur re	thi paper present ext	AH
11	11	model kno	activ theori	year larg nun	ronald	maier	inform commun techn	introduct knowledg v	paper discuss charact	H
12	13	knowledg i	knowledg m	paper explor	greg stefan nev stefa	timbrel koller schefe	organis unit inbound c	introduct knowledg i	get larger knowledg p	H
13	14	process ori	knowledg m	paper introd	robert dimitri	woitsch karagianni	knowledg manag evol	introduct km concep	thi paper introduc ne	H
14	15	formal con	atom concu	paper compl	dominiqu	abrial cansel	thi contain case studi	introduct defin queu	thi paper present con	D
15	17	investig atc	observ refin	fiction atom	jonathan cliff	burton jone	mani differ area comp	introduct framework	thi paper identi ed nc	F
16	19	precis mod	long run tra	describ stac l	michael carla muan	butler ferreira yong r	busi transact involv hi	introduct stac langua	combin explicit impli	DFH
17	20	replic unde	atom broad	quorum syste	richard	ekwal schiper	requir highli reliabl av	introduct differ isol c	common misundersta	CD
18	22	atom softw	atom softw	paper show c	jörg	kienzl	concept atom greek w	introduct softwar de	modern applic must r	D
19	24	semi auton	subgroup m	visual mine r	martin atzmuel	frank pupp	knowledg discoveri da	introduct process mc	thi paper present nov	HI
20	25	visual reco	reput social	contrast cent	jason	jung	web environ ha popul	introduct relat work	grow demand enviro	HJ
21	26	visual high	grand tour n	comput abil l	cesar colin	fyfe	strong desir data analy	introduct curv new p	present new extens e	HI
22	27	integr lite v	data mine v	visual tool in	li eamonn xiaopeng s	wei keogh xi lonardi	heart mani inform vis	introduct exampl ico	introduc intellig icon	H
23	28	connect se	data mine v	visual essent	francisco jesu jose	riquelm	visual techniqu provid	introduct relat work	veti mean approach v	EH
24	29	visual mani	data mine d	data mine co	deni	v popel	incomplet specifi func	introduct represent i	recent progress soft c	FI
25	30	gravi intera	interact info	track compar	klau silvia wolfgang s	hinum miksch aigner	visual tool use medic	introduct medic prob	present interact info	HJ
26	31	scalabl visu	visual data e	decad visual	daniel jorn	keim schneidewind	due progress comput	introduct relat work	thi paper present app	H
27	34	physic loca	wireless int	wireless net	frank prasanth rob go	adelstein alla joyc ric	wireless local area ne	introduct problem st	wid provid intrus det	D
28	35	refer mode	polici public	world interc	valentina rosa massir	casola preziosi rak tr	recent year due larg d	introduct secur back	thi paper present tec	K
29	36	increas rob	audio stega	paper preser	nedeljko tapio	cvejic seppanen	multimedia data hide	introduct standard ls	present reduc distort	DH
30	37	protomon e	comput sec	intrus detect	sachin stephen	joglekar tate	cryptograph protocol c	introduct relat work	main issu address thi	CDK

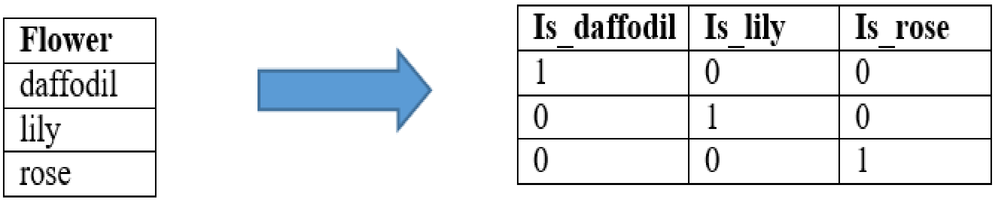
FIGURE 3.5: Preprocessed Dataset.

are mainly divided into two broad Category such as Count based approaches and Semantic based approaches. Some of the widely used count based techniques in research articles classification approaches are: 1) One Hot Encoding 2) Bag of Word (BOW) or Term Frequency (TF), 3) Term Frequency and Inverse Document frequency (TFIDF) etc, and semantic based approaches are: 1) Glove 2) FastText and 3)Word2Vec 4) BERT. Let us discuss their merits and demerits to be able to select the best approach for the implementation of the proposed model:

3.5.1 Count Based Approaches

3.5.1.1 One Hot Encoding

One-hot encoding is the most common and the most basic way to turn a text into a vector. It is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. In this strategy, each word is converted into a binary value 1 or 0, which indicates the word appears in a document or not. Suppose you have a 'flower' feature which



Flower
daffodil
lily
rose

Is_daffodil	Is_lily	Is_rose
1	0	0
0	1	0
0	0	1

FIGURE 3.6: Example of One-hot Encoding.

can take values 'daffodil', 'lily', and 'rose'. One hot encoding converts the 'flower' feature to three features, 'is_daffodil', 'is_lily', and 'is_rose' which all are binary. The graphical representation is shown in Figure 3.6.

Although this is a very simple strategy to implement but it has some disadvantages such as:

1. This method does not consider the position of a terms therefore it become difficult to examine the context of a word.
2. It does not consider the frequency information of a terms.
3. Vector representation size grows as the vocabulary size grows.

3.5.1.2 Bag of Word (BOW) or Term Frequency (TF)

As previously discussed, there is a frequency issue in one hot encoding. This strategy is straightforward and simple and they solve the frequency issue of one hot encoding. The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier. In BOW first, a fixed length vector is defined where each entry corresponds to a word in the pre-defined dictionary of words. The size of the vector equivalent to the size of the dictionary. Thereafter, to represent a document using this vector, one can count how many times each word of the dictionary appears in the document and then put this number in the corresponding vector entry. The following models a text document using bag-of-words. Here are two simple text documents:

TABLE 3.3: Example of Term Frequency.

Text	Ali	Likes	to	Watch	Cartoon	Mary	Too	also	Football
B1	1	2	1	1	2	1	1	0	0
B2	0	1	1	1	0	1	0	1	1

1. Ali likes to watch Cartoon. Mary likes cartoon too.
2. Mary also likes to watch football.

The vectors of both of these documents are shown in Figure 3.3 . This strategy solves the frequency issue of one hot encoding. Moreover, this strategy does not consider the order of words and the semantic and context of words.

3.5.1.3 Term Frequency and Inverse Document Frequency (TFIDF)

TFIDF [34] tells us about the significance of terms in a document. It contains two concept Term Frequency (TF) and Inverse Document Frequency (IDF). **Term Frequency**, which measures the frequency of occurrence of a term in a document. The text document varies in their length. If a document is long, then a word may appear more times in it than a document whose length is short. The term frequency of a document is divided by the length of the document. It will be easily understandable by an example. If a user wants to query “the lazy fox”, let the user have a set of English text documents. Then rank the documents by the order of relevancy of the query. First delete all those documents which do not contain all the three words “the”, “lazy”, and “fox”. Then some documents are left. Then count how many times a word occurs in a document. The number of times a term occurs in a document is called its term frequency.

$$TF(t) = \frac{NumberOfTimesTerm(t)AppearsInDocument}{TotalNumberOfTermsInDocument} \tag{3.3}$$

The **Inverse document frequency (IDF)** measures the importance of a term in a document. In term frequency, all the terms are equally important. But there are some terms, such as “is”, “of”, and “the”, which may appear a lot of times but their importance is very little. Consider the above example of the query “the lazy fox”. The term “the” is so common but it has no importance. On the other side, the term “lazy” and “fox” are more important terms. The term “the” cannot show the relevancy or non-relevancy of the document. Hence, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

$$IDF(t) = log_{10} \frac{TotalNumberOfDocuments}{NumberOfDocumentsWithTerm(t)init} \tag{3.4}$$

An example is given for understanding the TFIDF: a text document contains 100 words where the word “data” appears 10 times. The term frequency (i.e., tf) for data is then (10 / 100) = 0.1. Now, assume one has 1 million documents and the word data appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(1,000,000 / 1,000) = 3. Thus, the Tf-idf weight is the product of these quantities: 0.1 * 3 = 0.3. However, the TFIDF performs well in different scenarios and it has solved many issues of previous techniques but they lack Semantic and the context of the terms.

After a detailed analysis of some of the famous text representation techniques which have been used in the literature of research article classification, it has been observed that while capturing information these techniques mostly rely on the frequency of terms and ignored the semantic and context of terms. The current state-of-the-art approaches [1] [2] [3] [4] [5] [6] for research article classification have employed these conventional statistical measures like one hot Encoding, BOW, and TFIDF etc. Due to which they have not considered the semantic and context of the terms so that’s why they might assign a wrong category to the research articles. However, this study has focused on text representation in this thesis. Before performing any mathematical operation like finding similarity between text document, the semantic and context of a text is considered as representation which is ignored by existing statistical measures. So now in next section this thesis have discussed an alternative of the above mentioned strategies which can capture the semantic and contextual information of terms and it is widely used in different domains and producing good accuracy [13] [15] [35].

3.5.2 Semantic Based Techniques

3.5.2.1 Word Embedding

To represent a word we must know the semantic and context of a term in which the term is used because the meaning of a term varies in a different context. For example, let us consider the word ‘bank’. There are different meanings of word ‘bank’. One meaning of a term are a financial institution and another one is land alongside a body of water. If in a sentence, a bank occurs with neighboring words such as: money, government, treasury, interest rates, etc. we can understand it is the former meaning. Contrarily, if neighboring words are water, shore, river, and land, etc. the case is latter. After performing an in-depth study we have identified one of the techniques known as word Embedding which is used in different fields such as 1) Image processing 2) NLP Tasks 3) Biosciences etc, to represent a text by using different models.
A word embedding [36] is a learned representation for text where words that have the same meaning have a similar representation. It is the approach of representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems. Each word is represented by a real-valued vector, often tens or hundreds of dimensions.

3.5.2.2 Word Embedding Algorithm

Word embedding[37] methods learn a real-valued vector representation for a predefined fixed sized vocabulary from a corpus of text. The learning process is either joint with the neural network model on some task, such as document classification, or is an unsupervised process, using document statistics. Three techniques are used to learn word embedding from text data.

3.5.2.3 Embedding layer

The text document should be preprocessed so that each word can be converted into one hot encoding. There are different dimensions of the model such as 50, 100 or 300. First, the size of the vector space should be specified. The initialization of vectors is done by random numbers. The embedding layer is used on the front end of a neural network and is fit in a supervised way using the Backpropagation algorithm. A lot of training data is required to learn this approach of embedding layer and it is a slow process.

3.5.2.4 Word2Vec

It is a method of learning word embeddings from a text corpus. It is a statistical method. Tomas Mikolov, et al. at Google developed it in 2013. It is based on neural network training and it makes the embedding more efficient. It converts the words into vectors by considering their semantics. For example, that subtracting the "man-ness" from "King" and adding "women-ness" results in the word "Queen", capturing the analogy "king is to queen as man is to woman". Word2vec approach has two different learning models which are considered as a part of this approach to learn the word embedding; they are:

- Continuous Bag-of-Words, or CBOW model.
- Continuous Skip-Gram Model.

The CBOW model learns the embedding by predicting the current word based on its context. The continuous skip-gram model learns by predicting the surrounding words given a current word. The advantage of the approach is that finest word embedding can be learned efficiently (low space and time complexity), allowing larger embedding to be learned having more dimensions from much larger corpora of text. But it also has a limitation that it is one way means it reads text from left to right and not the other way round.

3.5.2.5 Glove

Glove is another semantic based technique proposed by Pennington et al. [38] in 2014 with Stanford University. This technique come after word2vec one year letter.

This technique studied that World2Vec has some weak points such as Word2Vec model learns the word embeddings by relating target words to their context. But it does not consider the frequency of terms. World2Vec considers that if a word occurs more often, it has no information except it adds more training examples. But Glove focuses on the frequency of words and shows that its co-occurrences contain important information and it is not useless information. By considering all this, Glove builds word embeddings. The main issue of Glove technique is that it focuses more on word co-occurrences over the whole corpus. It is behaving like count-based approach.

3.5.2.6 FastText

FastText was released in 2016 by Facebook. At that time, there was one problem which remains unsolved for some time. The problem is the generalization to unknown words. FastText claims that they can remove this hindrance. The previous techniques use words for the embedding of words. But FastText moves on a deeper level than words and they started to consider the parts of words and the characters. It makes a word its context and its building blocks are characters now. So, for example, take the word, "Computer" with n=3, the representation of this word given by FastText is: $\langle co, com, omp, mpu, put, ute, ter, er\rangle$, where the angular brackets show the starting and ending of the word. FastText works well with rare words. If the words during the phase of training is not seen, its means that it is sliced into n-grams.
This approach has two advantages.

- It solves the generalization problems.
- There is no need for more training examples.

There is some drawback of this model such as:

- High memory requirement and
- More focus on Syntactic of the word rather than semantic.

3.5.3 BERT

BERT stands for Bidirectional Encoder Representations from Transformers [36]. BERT is a recent paper published by researchers at Google AI Language. It has brought a revolution in the field of machine learning by presenting state-of-the-art results in a wide variety of NLP tasks, including Question Answering, Natural Language Inference, and others. Researchers are pre-training a neural network model on a known task, for instance, ImageNet, and then performing fine-tuning using the trained neural network as the basis of a new purpose-specific model.

3.5.3.1 Working

Transformer is a mechanism that learns contextual relations between words in a text. BERT makes use of a Transformer. There are two mechanisms in transformers 1) an encoder that reads the text input and 2) a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary.

One of the main advantages of using BERT is, as opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it's non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word). It is designed to pre-train deep bidirectional representations [39] from an unlabeled text by jointly conditioning on both left and right contexts. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks.

- BERT stands for Bidirectional Encoder Representations from Transformers. It is based on the architecture of the Transformer.
- BERT is a pre-trained model. Its pre-training is done on two types of unlabeled textual data. Firstly, it is trained on a large collection of text which

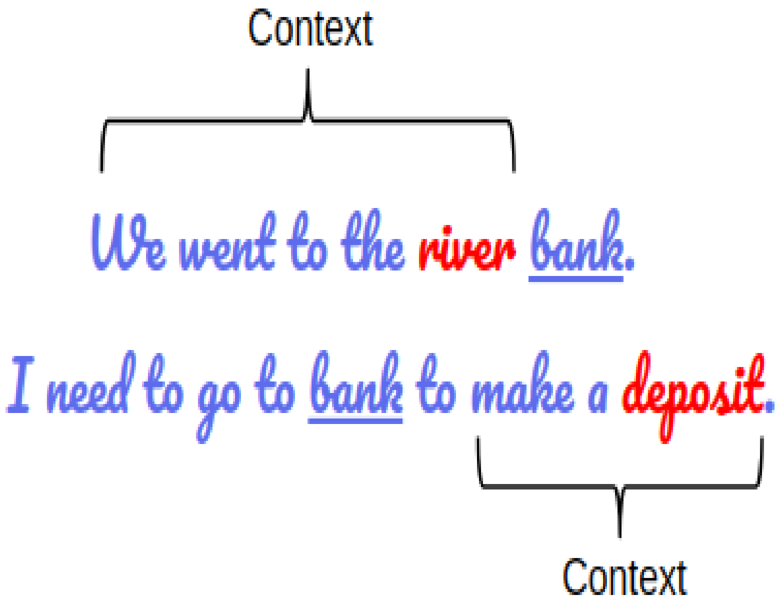

Context
We went to the river bank.
I need to go to bank to make a deposit.
Context

Figure 3.7: Bert Example [36].

includes the entire Wikipedia that includes 2,500 million words, and secondly, on the Book Corpus which contains 800 million words. This pre-training of the model is the reason behind the success of Bert.

- When the training phase starts, it goes into a deeper level and develops better understandings of the language of the model.
- BERT is a "deeply bidirectional" model. It means that during the training phase, it learns information from both the left and the right sides of the context of tokens.

This bi-directionality of a model plays a significant role in understanding the language. Figure 3.7 shows the example of Bert. There are two sentences in Figure3.7 and the word "bank" is involved in both of them.

If sentence 1 is read, then the word bank is used in the context of riverbank. And sentence 2 is used in the context of a cash deposit or withdrawal from bank. But if these sentences are read from only one side then there is a mistake in predicting the meaning of "bank" in any of these two sentences. The solution of this problem is that one has to read the sentences from both sides. This is the way how Bert works. The output layers can be added for fine-tuning and can perform various NLP tasks. The training data is small for performing NLP tasks. Moreover, the context of the word is not taken by these NLP models. See the above example of the word "bank", then in both sentences the context of the word is different. However, an embedding like Word2Vec will give the same vector for "bank" in both contexts.

After a detailed analysis of count and semantic based technique this study has concluded that BERT model is best for capturing the semantic and contextual information of terms. So this thesis has used the BERT model for vectorization of a text.

3.5.3.2 BERT Input

The input to the Bert consists of one or more than one sentence, and it uses a special token.

1. The [SEP] token is used to differentiate between two sentences.
2. The token that appears at the start of the text is [CLS], and this token is especially to perform the tasks of classification.

We have to use both tokens if the number of a sentence is one.
2 Sentence Input: [CLS] Ali likes to play football. [SEP] His friend likes to plays with him.
1 Sentence Input: [CLS] Ali likes to play football. [SEP]

3.5.3.3 Tokenization and Word Embedding

Next, let's see how the words are converted into vector form. The example is shown in Figure 3.8.

For example, the sentence is given as:

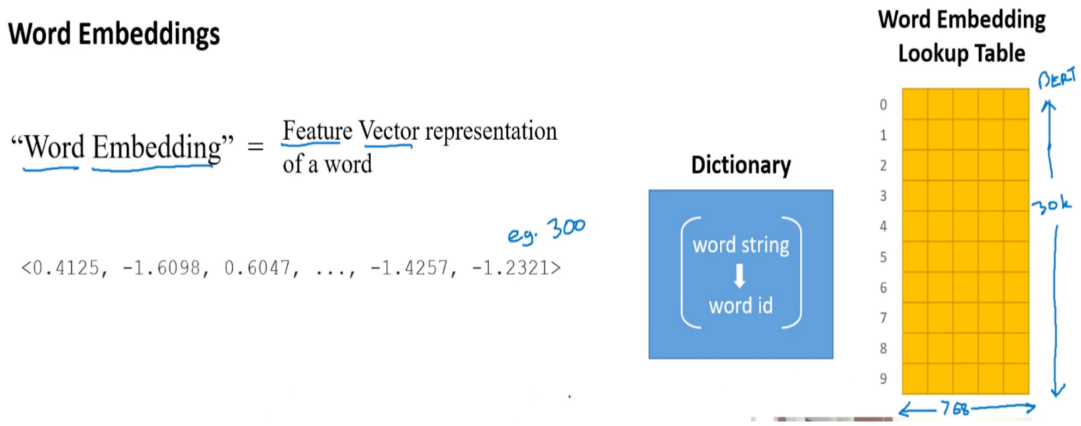


Figure 3.8: Word Embedding [40].

sen = "Ali likes to play football."

marked = "[CLS]" + sen + "[SEP]"

Then the sentence is tokenized using BERT tokenizer. This is done by importing bert tokenizer from transformers.

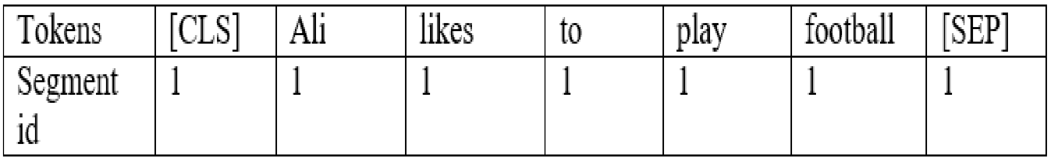
tokenized = tokenizer.tokenize(marked)

After that, the tokens are given as: ['[CLS]', 'Ali', 'likes', 'to', 'play', 'football', '.', '[SEP]']

3.5.3.4 Segment ID

If the number of sentences is more than one, then BERT is trained on and expects sentence pairs, using 1s and 0s to distinguish between the two sentences. That is, for each token in "tokenized_text," it must specify which sentence it belongs to sentence 0 (a series of 0s) or sentence 1 (a series of 1s). If there is a single sentence it requires series of 1s, so a vector of 1s for each token in our input sentence will be created. But if the number of sentences is more than one, assign each word in the first sentence plus the '[SEP]' token a 0, and all tokens of the second sentence a 1.

Let's try to convert the above sentence into tokens. There are 8 tokens in the above sentence so mark each token as belonging to the sentence "1". This is shown in Figure 3.9.



Tokens	[CLS]	Ali	likes	to	play	football	[SEP]
Segment id	1	1	1	1	1	1	1

FIGURE 3.9: Converting into Segment ID.

The Figure 3.10 shows the procedure for tokenizing the dataset. In step 1, the data is read from a CSV file. In step 2, iterate all the records of the dataset. In step 3, the records are split into words. In step 4, preprocessing is performed on the dataset. In step 5, every sentence of the dataset is marked with special tokens. In step 6, the sentences are tokenized using Tokenizer. In step 7, the tokenized sentences are converted into segment id. Step 8 shows the process of converting into segment ID and then in step 9, the file is saved in a CSV file.

3.5.3.5 BERT Pre-trained model

Now the data is converted to tensors that are the input format for the model. Next, the pre-trained BERT model is called. There are many pre-trained models available but the models are divided into two broad categories:

Algorithm 1 :Procedure for tokenizing all sentences using Tokenizer and converting into segment ID

```
Input: JUCS Dataset
Output: Tokenized Sentences
 1: $Data \leftarrow Read\ Dataset\ Records$
 2: for all $i = 1$ to $len(Data)$ do
 3:     $Records \leftarrow i.split("")$        $(SplitRecordsintowords....)$
 4:     $Update \leftarrow Preprocessing(Records)$
 5:     $marked \leftarrow [CLS] + Update + [SEP]$
 6:     $tokenized \leftarrow tokenizer.tokenize(marked)$
 7:     $indexed\_token \leftarrow tokenizer.convert\_tokens\_to\_ID(tokenized)$
 8:     $segment\_ID \leftarrow [1] * len(Data)$
 9:     $File \leftarrow File.save('Preprocessed\_data.csv')$
10: end for
```

FIGURE 3.10: Procedure for tokenizing all sentences using Tokenizer and converting into segment ID

(a) Bert base

(b) Bert large

This study uses the model Bert base uncased that is shown in step 1 of figure 3.11. It contains 12 layers, there are 768 hidden units, 12 heads, 110M Parameters and it is trained on lowered cased English text.

3.5.3.6 Vector Representation

Step 2 in the figure 3.11 is the output of the model, step 3 shows that the output is saved into hidden states. Now try to understand the output. The hidden states have four dimensions, which are as follow:

1. There are 13 layers in it, 12 layers are the Bert outputs and one additional layer is the input embedding.
2. The number of sentences is the batch number. In the example above, there is one sentence.

Algorithm 2 :Procedure for Vector Representation

```
Input: Dataset
Output: Vector
 1: $Pretrained\_Model \leftarrow BertModel.fromPretrained(output\_hidden\_states = True)$
 2: $Output \leftarrow Pretrained\_model()$
 3: $hiddenStates \leftarrow Output$
 4: $Data \leftarrow Read\ Dataset\ Records$
 5: $Model \leftarrow Output$
 6: for all $col$ in $Data$ do
 7:     for all $row$ in $Data$ do
 8:         for all $word$ in $row$ do
 9:             $Sum \leftarrow Sum + Model[word]$
10:         end for
11:         $Average \leftarrow Average(Sum)/len(row)$
12:         $File \leftarrow WriteVectorinFilewithRowLabel$
13:     end for
14: end for
```

FIGURE 3.11: Procedure for Vector Representation

3. There are 8 words in our example sentence.

4. Every token has 768 hidden units which are also called its feature number.

The feature vector of one sentence in the example gives 79,872 unique values. This is the representation of a single sentence. As the sentence length will increase the number of unique values also increases, this will increase the complexity. There should be only one vector to represent a single token or a single sentence. But as discussed above, each token has 13 different layers and every layer has a length of 768 units. If a single vector representation of a word is needed, some of the layers have to be aggregated. But there is a question that how the aggregation of layers is performed so that it will give the best results.
Let's try to concatenate the last four layers, it gives a single word vector per token. The length of each vector will be 4 x 768 = 3,072. If an alternative method is tried which is, that the vector of words will be created by doing the sum of the last four layers then the vector will have a length of 22 x 768 = 16,896. This study has tried two different methods but all the above values are still big enough. Let's try to get a single vector that represent the whole sentence. There are different approaches for doing it, but the most commonly used approach is by taking the average of the second to last hidden layer of each token. It will produce a single vector for a whole sentence and it has a length of 768 values. So in this model, the last strategy of taking an average of layers and producing a single 768 length vector is used.

The Figure 3.11 shows the procedure for vector representation of the dataset. In step 1, the bert pre-trained model is used. In step 2, the pre-trained model is saved in output, and step 3, the output is saved in the hidden states. In step 4, all the records are read from the dataset. In step 5, the output is saved in Model. In step 6, iterate all the columns of the dataset. In step 7, iterate all the rows of the dataset. In step 8, iterate all the words of the dataset. In step 9, the vectors are generated against each word. Then sum all the words in a sentence. In step 11, the average of a row is taken and it generates a single vector for every feature value. In step 12, the file is saved. In this way, we have vector representation of text.

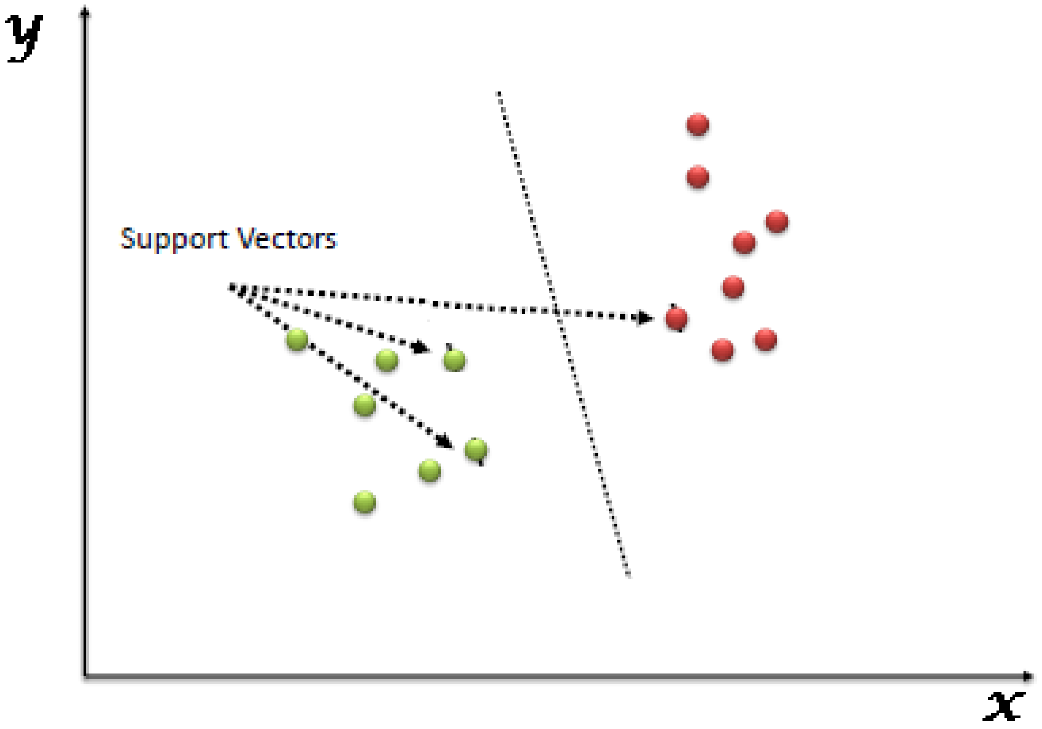


FIGURE 3.12: Example of Support Vector Machine [41].

3.6 Classification

The dataset is in vector form now, next step is to perform classification. This study uses the Support Vector Machine (SVM) Classifier [41]. It is a supervised machine learning algorithm which can be used for both classification and regression problems. But most commonly it is used in classification problems. Each data item

Algorithm 3 :Procedure for performing Classification

```
Input: Dataset in Vector form
Output: Accuracy
$Data \leftarrow Read\ Dataset\ Records$
$X \leftarrow Features$
$Y \leftarrow Target$
$Train\_Test \leftarrow Splitdatasetintrainingandtesting$
$Model \leftarrow Training\_dataset$
$Prediction \leftarrow Predict\_Testing\_dataset$
$Actual \leftarrow Test\_labels$
$Accuracy \leftarrow Accuracy\_score(Prediction, Actual)$
```

FIGURE 3.13: Procedure for performing Classification

as a point is plotted in n-dimensional space (where n is the number of features) with the value of each feature being the value of a particular coordinate. Then, the classification is performed by finding the hyper-plane that differentiates the two classes very well. The example is shown in Figure 3.12.

The Figure 3.13 shows the procedure for the classification of the dataset. In step 1, all the records are read from the dataset. In step 2, features are label as X. In step 3, Target is label as Y. In step 4, the dataset is split into training and testing data. In step 5, the model is trained on an svm classifier. In step 6, the dataset is used for the prediction of test labels. In step 7, actual test labels are retrieved. Then in step 8, the accuracy of the classifier is achieved.

3.7 Optimization Algorithm

Over the last decades, there has been a growing interest in algorithms inspired by the observation of the natural phenomena. It has been shown by many researchers that these algorithms are good replacements as tools to solve complex computational problems. Various heuristic approaches have been adopted by researchers including genetic algorithm, tabu search, simulated annealing, and ant colony and particle swarm optimization. This thesis focuses on using an optimizer that can do feature selection for document classification because as the number of features increases its complexity increases.

3.7.1 Genetic Algorithm

This study uses a genetic algorithm [17] for selecting features to classify documents. Moreover, the overall accuracy of the classification system can be improved. It is a revolutionary algorithm for feature selection. It is a stochastic method and heuristic approach for function optimization. It is influenced by the procedure of natural selection. The population of individuals is created for achieving good results. The Figure 3.14 shows the state diagram of how the genetic algorithm works.

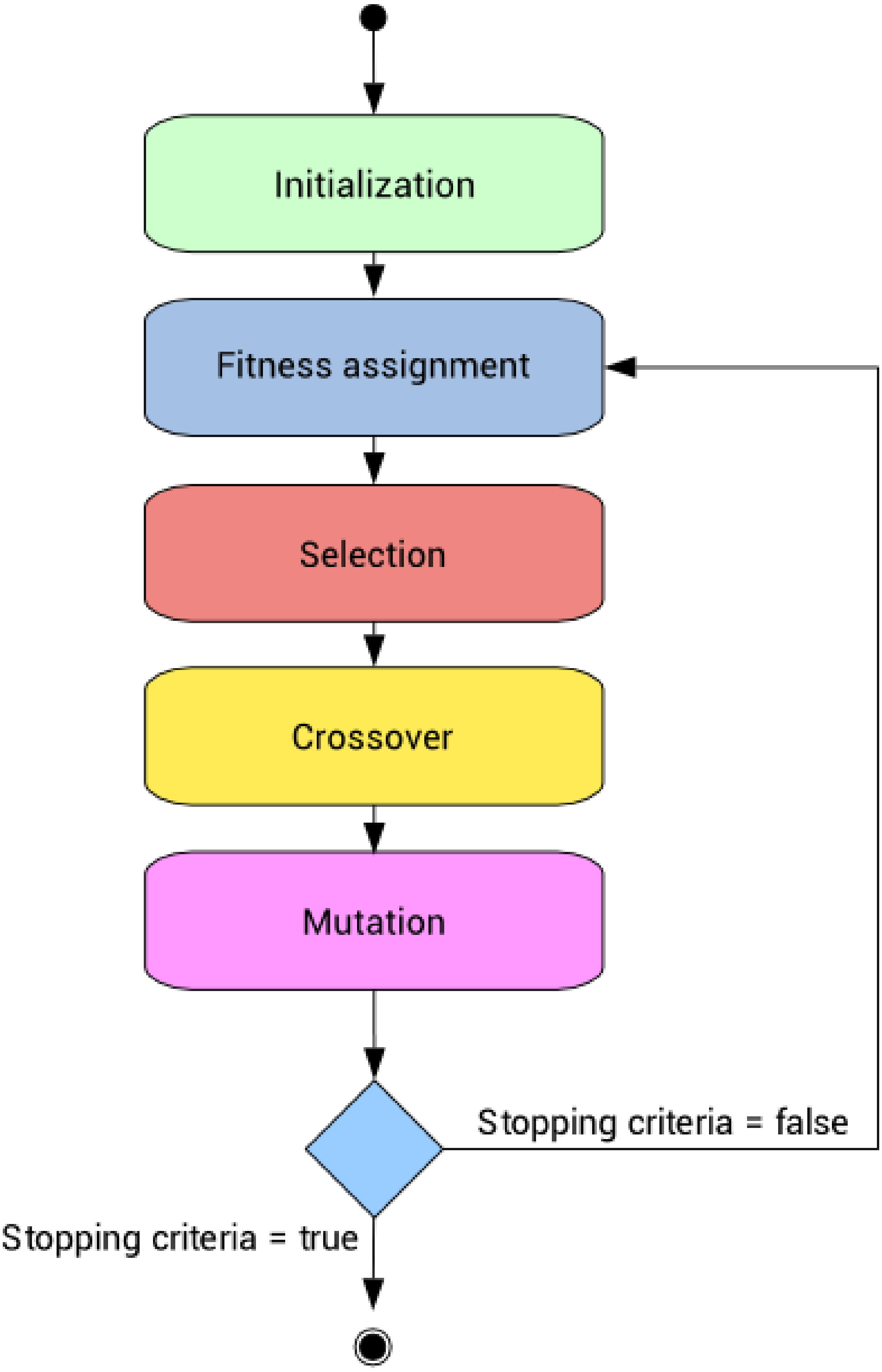


Figure 3.14: Working of Genetic Algorithm [17].

3.7.1.1 Initialization

Firstly, the individuals are created and are initialized in the population. Every individual is represented by a neural network in the population that was created. Genes are in binary representation which shows the presence or absence of a specific feature. In this case, eight features are represented by a neural network. For example, if a population is generated which typically consist of two individuals, it mean that there are two different neural networks and each network has some arbitrary features. In this case, there are eight features so the genes of every individual are 8 in number. If a gene has a positive value then it means that a particular gene is present in the network. The Table 3.5 and 3.5 shows the individuals.

3.7.1.2 Fitness assignment

As discussed, there are different individuals in the population. Every individual should have a fitness value. Then, a model is used for training every individual which is a neural network. The prediction is then performed on the testing instances. This study used accuracy as a fitness function.

Table 3.4: Individual 1

1	0	1	0	1	0	1	0

Table 3.5: Individual 2

1	1	0	0	1	0	0	0

3.7.1.3 Selection

The next step after the fitness assignment is selection. In the next generation, the individuals in the population are selected for recombination. Individuals having the best fitness value will be survived. The individuals are selected based on their fitness value. The size of individuals who are selected is half of the population size. In the example above, a population of two individuals is created. So, here the number of a selected individuals is one.

3.7.1.4 Crossover

After the selection of half of the population, the individuals which are selected are combined again using the crossover operator and it generates a new population. The operator selected two individuals randomly and their features are integrated to get the offspring. Consider the above example of two individuals, these individuals are selected by the crossover operator and it produces offspring. This procedure of offspring production continues until the newly created population size becomes equal to the size of the old population.

The example of the crossover method is shown in Table 3.6, Table 3.7, Table 3.8, Table 3.9, Table 3.10, Table 3.11.

Table 3.6: Individual 1

1	0	1	0	1	0	1	0

Table 3.7: Individual 2

1	1	0	0	1	0	0	0

Table 3.8: Offspring 1

1	1	1	0	1	0	0	0

Table 3.9: Offspring 2

1	1	0	0	1	0	1	0

There are different crossover operators [42], two basic are described as:

Table 3.10: Offspring 3

1	1	1	0	1	0	1	0

Table 3.11: Offspring 4

1	0	0	0	1	0	0	0

1. **Single Point Crossover**

   It is the most widely used operator. The crossover site is selected randomly and better offspring can be obtained by combining the good quality parents.

2. **N Point Crossover**

   It is the same as single-point crossover. But adding more crossover sites effects the disruptions of building blocks and the performance of the algorithm reduces.

3.7.1.5 Mutation

The offspring generated after performing crossover have high similarity with their parents. The new generation is constructed but the problem is, its diversity is low. This problem is solved by the mutation operator in such a way that it changes the value of some genes or features in the offspring randomly. A random number is generated that lies between 0 and 1, to decide if the feature is mutated or not. The variable is flipped when the number that generated randomly is lower than the mutation rate. The mutation rate generally has less value. Usually, it is selected as 1/m, where m is the number of features. Then the features of every individual is mutated with this value. The example is shown in Figure 3.12 and 3.13.

Table 3.12: Offspring1 Original

1	1	1	0	1	0	0	0

The parameters of GA used are shown in Table 3.14.

Table 3.13: Offspring1 Mutated

1	1	1	0	1	1	0	0

Table 3.14: Parameters of GA.

Parameters	**Values**
Encoding	Binary(1 shows the selection of a particular feature; 0 shows that the particular feature is not selected)
Size of population	50
Number of generations	100
Crossover operator	One Point Crossover
Mutation operator	Mutation Flip Bit
Mutation probability	0.02
Selection	Tournament

The Figure 3.15 shows the procedure for getting a subset of features. In step 1, the size of the population is initialized. In step 2, the individuals in the population are initialized. In step 3, a variable i is initiated with zero. In step 4, the individual's fitness is calculated. In step 5, there is a loop that iterate through all the individuals in a population. In step 6, the individuals having the best fitness rate is selected. In step 7, a crossover operation is performed on individuals to get the offsprings. In step 8, the mutation of individuals is performed. In step 9, the fitness of newly created individuals is created. Step 10 and 11 shows the increment of the loop. In step 12, the loop ends. In step 13, a feature subset is derived having the highest fitness value.

3.7.2 Evaluation

The proposed technique is evaluated based on accuracy, precision, recall, and F1 score. The approach is evaluated on the JUCS dataset. The evaluation parameter is given below:

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{3.5}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.6}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3.7}$$

$$F1 - Score = \frac{2(Precision)(Recall)}{Precision + Recall} \tag{3.8}$$

Algorithm 4 :Procedure for Feature Selection

Input: Dataset in Vector form

Output: Optimal Feature Subset

```
$N \leftarrow Size\ of\ Population$
$P \leftarrow Initialize\ individuals\ in\ Population$
$i \leftarrow 0$
$CalculateFitnessofP_i$
for all $P_i$ do
    $SelectIndividualsfromP_i$
    $RecombineIndividuals$
    $MutateIndividuals$
    $CalculateFitnessofNewlyCreatedIndividuals$
    $P_{(i+1)} \leftarrow NewlyCreatedIndividuals$
    $i \leftarrow i + 1$
end for
$S \leftarrow DerivedwithBestIndvidualsinP_{(i+1)}$
```

FIGURE 3.15: Procedure for Feature Selection

Chapter 4

Result and Evaluation

Chapter 3 explained the proposed methodology in detail. The results are obtained by applying the proposed methodology. In this chapter, the results have been given about the data extraction, preprocessing data, vector representation and then feature optimization.

4.1 Dataset

The evaluation of the proposed technique is based on the dataset. The details about the dataset are already explained in chapter 3. JUCS dataset is selected for performing experimentation. The JUCS dataset consists of 1460 research articles. While analyzing the dataset, it has been analyzed that 94 research articles do not contain their respective categories, 90 research articles are managing editor columns, 10 research articles does not contain an introduction, 187 research papers don't have a conclusion, 33 research papers content can't be copied therefore these records are removed from the dataset. The number of categories to which research papers belong is 13. These categories belong to the computer science domain. But this study has deleted those records which belong to categories L (Science and Technology of Learning) and M (Knowledge Management) because they have very few records in the dataset. The number of papers left for experimentation is 933. The state of the papers is illustrated in Table 4.1.

Table 4.1: JUCS Papers Detail.

Types of Papers	**Records**
Special issue papers	107
Managing editor columns	90
Papers that don't include introduction	10
Papers that don't contain conclusion	187
Papers whose content are not available	33
Papers belonging to category L and M	6
Papers whose categories are not specified	94
Remaining papers for experimentation	933

4.2 Data Extraction

The next step is to extract features from the JUCS dataset. For data extraction, there are two possible ways: 1) Manual extraction 2) Automatic extraction that is machine-oriented. This study has done manual extraction of data. There are two types of data: Metadata and content based. It is very easy to write an algorithm for extraction of metadata features but it is very difficult to write an algorithm for extracting the content based features, that's why we have done manual extraction of data. This study has extracted Metadata and Content based features which are eight in number. In this thesis, Title, Keywords, Author first name, and Author last name, Abstract, Introduction, Headings, and Conclusion are extracted from the JUCS dataset. All these features are present because this study has already removed records whose content is not available.

4.3 Pre-processing

The extraction of all metadata and content based features is done, next step is to pre-process the extracted data because the data needs to be cleaned. These steps are performed in preprocessing:

1. The first step is to read the dataset from a CSV file using the pandas library.
2. Then convert all datasets in lower case.
3. Tokenize the text of all features by using NLTK Library i.e. tokenize.
4. Removing noise from all features i.e. punctuations, digits, etc.
5. Removing stop words from all features using NLTK Stop words.
6. Stemming the text of all features using NLTK stemmer i.e. Porter Stemmer.
7. Then rejoin the words and write in a file.

4.4 Text Representation

The pre-processing is done successfully and now the data in a CSV file is ready for vector representation. This study uses BERT for the vectorization of text which considers semantic meaning of terms. This process includes the following steps:

4.4.1 Input

The input to the BERT is slightly different as compared to other approaches. The JUCS dataset is converted into an input form that BERT accepts. For this, there are two special tokens:

1. The [SEP] token is used to differentiate between two sentences.
2. The token that appears at the start of the text is [CLS], and this token is especially to perform the tasks of classification.

After converting text into the input form which BERT accepts, this study has tokenized the sentence and tokenization is done using BERT tokenizer and then convert it into segment id. There is an algorithm for tokenizing and segmenting all dataset and it is given in chapter 3.

4.4.2 BERT model

There is no need to train the Bert model because it is a pre-trained model. Its pre-training is done on two types of textual data which is unlabeled. Firstly, it is trained on a large collection of text which includes the entire Wikipedia that includes 2,500 million words, and secondly, on the Book Corpus which contains 800 million words. This pre-training of the model is the reason behind the success of Bert. When the training phase starts, it goes into a deeper level and develops better understandings of the language of the model. BERT is a "deeply bidirectional" model. It means that during the training phase, it learns information from both the left and the right sides of the context of tokens. The BERT pre-trained model is available on the web. This study uses the Bert base uncased model which has 12 layers and 768 hidden units. The algorithm for using the Bert model is given in chapter 3.

4.4.3 Vectorization

The next step is to convert the text dataset into vector form. For a single word, this algorithm generates 13 separate vectors each of length 768, because Bert base model has 768 hidden layers. In a single document, there are multiple words so vector size also increases. This generates a vector of variable length. To generate a vector of fixed length for the entire sentence, take an average of all the layers and it produces a single length of 768 vectors. The algorithm for text representation is given in chapter 3. By using this algorithm this thesis has successfully converted the text of all features into vectors form. The successful conversion percentage of all features is 100%.

4.4.4 Classification

The next step after vector representation is classification. For classification, SVM classifier is used. This study first split the dataset into training and testing and then train the training dataset on an svm classifier. After the training phase is over,

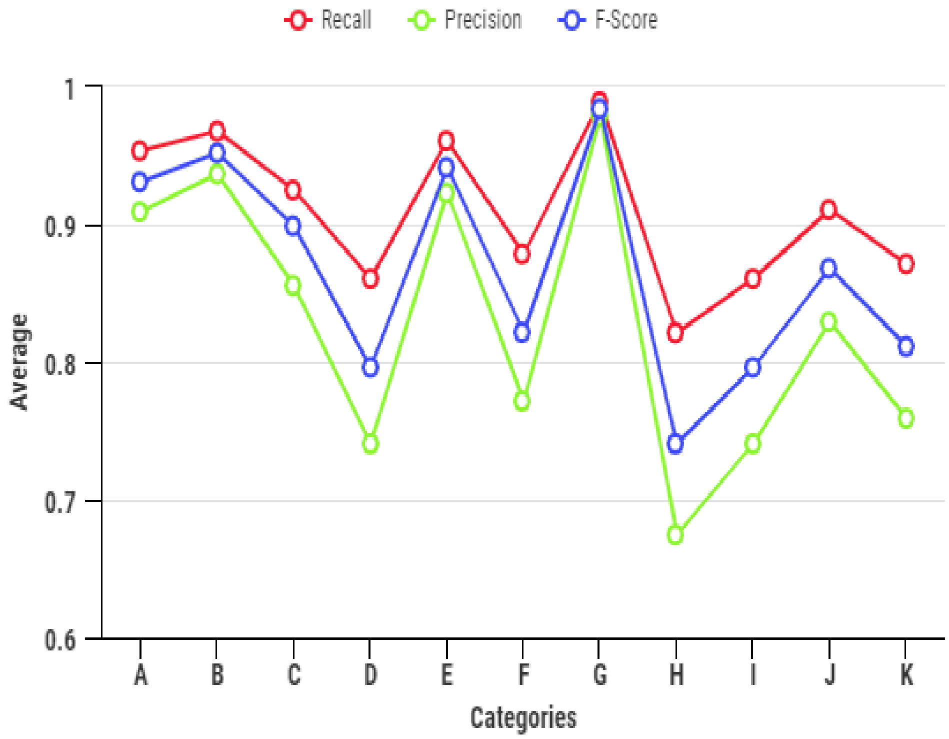


FIGURE 4.1: Category wise average Precision, Recall and F-Score.

the classifier predicts based on the test dataset. The algorithm for classification is given in chapter 3. For evaluation of the proposed technique, there are eight features in the dataset i.e. Title, Keywords, Abstract, Author first name, Author last name, Introduction, Headings, and, Conclusion and this study has used the records of all 11 categories. This study has performed multiclass classification.

The classification measures used for classification are accuracy, precision, recall, and f1 score. The Accuracy achieved by the classifier is 91%, precision is 0.8290, recall is 0.91 and f1 score is 0.8674. The average result of each measure is represented through the graph. Figure 4.1 shows the graph between categories and their precision, recall and F-Score values. The Table 4.2 shows the value of precision, recall and F-Score against every category. Precision is the percentage of the true positive as correct. Recall is the percentage of the true positive as predicted. F-Score is the harmonic mean of the precision and recall. Category G has the heighest value for precision, recall and F-Score. Category H has the lowest precision, recall and F-Score values.

Table 4.2: Category wise Precision, recall and F-Score.

Category	Precision	Recall	F-Score
A	0.9092	0.9535	0.9309
B	0.9367	0.9678	0.9520
C	0.8556	0.925	0.8989
D	0.7408	0.8607	0.7962
E	0.9229	0.9607	0.9414
F	0.7718	0.8785	0.8217
G	0.9786	0.9892	0.9839
H	0.6747	0.8214	0.7408
I	0.7408	0.8607	0.7962
J	0.8294	0.9107	0.8681
K	0.7593	0.8714	0.8115

4.5 Optimization

This study has achieved very good accuracy by performing feature selection. For feature selection genetic algorithm is used. The number of features used are eight in this dataset. In literature, researchers have only utilized metadata features and they do manual combinations of features. This study has done optimization to reduce complexity. A lot of feature subset are possible which define different combinations of features. With the increase number of features, the complexity is exponential. That's why this study has performed optimization by using genetic algorithm which is a metaheuristic. Genetic algorithm is one of the revolutionary algorithm for feature selection. It operates on a population of individuals to produce better results. After performing optimization the following combination of features is produced giving good accuracy. The different combinations of features is shown in Table 4.4. Figure 4.2 shows how accuracy of the algorithm fluctuate as the number of iterations is increasing.

TABLE 4.3: Subset of Features.

Name of Features	**Number of Features**	**Accuracy**
Title, Abstract, Author First name, Introduction, Conclusion	5	98.15%
Keywords, Abstract, Author First name, Introduction, Headings	5	96.93%
Abstract, Conclusion	2	95.70%
Title, Author First name, Conclusion	3	95.09%

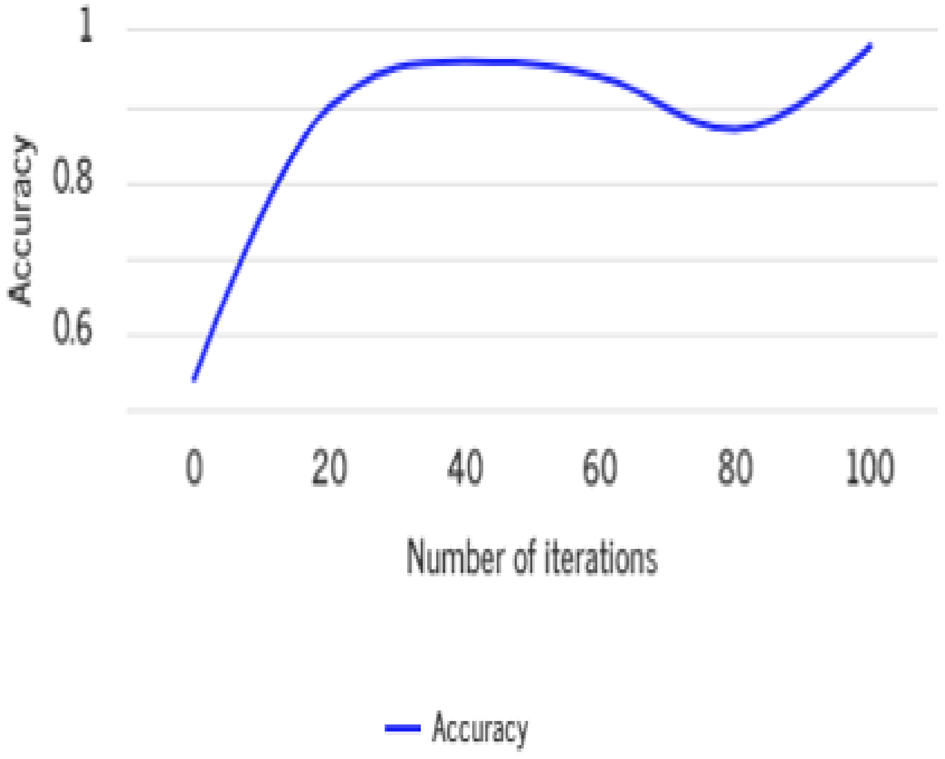


FIGURE 4.2: Number of iterations vs accuracy.

Table 4.4: Comparison of Techniques.

Papers	Features	Accuracy
Sajid et al. (2011)	Authors, References	70%
Sajid et al. (2016)	Title, Keywords	91%
Ali and Asghar (2018)	Title, Keywords, Abstract, Author	78.79%
Proposed Approach	Abstract, Conclusion	95.70%

4.6 Comparison

There are multiple techniques to perform multi class classification and the document classification community has proposed different approaches. Researchers have utilized metadata features only because it is freely available. Mostly the content of the research articles is not available. But this study has utilized both metadata and content of features. The features used are 1) Title, 2) Keywords, 3) Abstract, 4) Author first name, 5) Author last name, 6) Introduction, 7) Headings, and 8) Conclusion. The proposed approach is compared with the approaches proposed in the literature. In the literature, they have utilized only metadata of the research articles of the JUCS and ACM dataset. The comparison is shown in the Figure 4.3.

Sajid et al. (2011) propose a technique by using the JUCS dataset and they uses content based approach. They achieved 70% accuracy on their proposed approach. Then again Sajid et al. (2016) proposed fuzzy based rule merger and they used JUCS dataset. They uses metadata based approach and achieved 91% accuracy. Ali and Asghar (2018) proposed an approach to convert multi label to single label using JUCS dataset. Their approach is based on metadata and they achieved 78.79% accuracy. The proposed technique uses both metadata and content based approach and JUCS dataset is used for the experimentation. The results outperformed the previous techniques by achieving accuracy 98.15%.

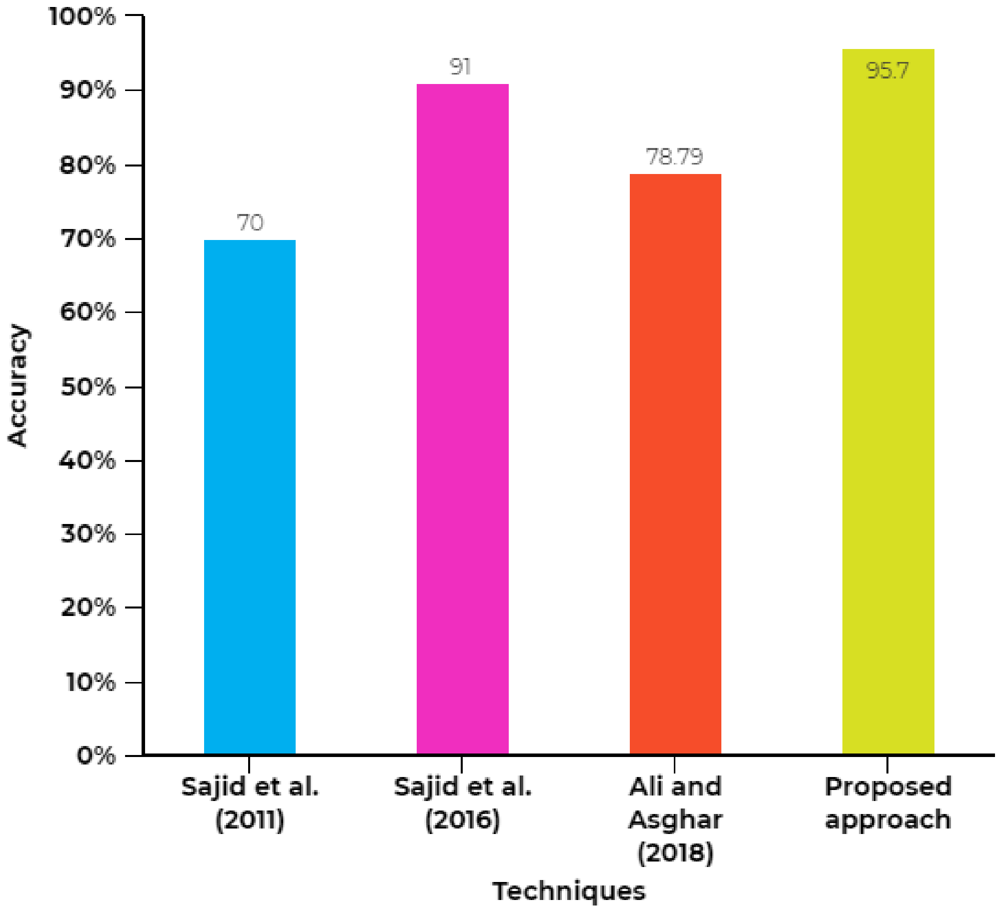


Figure 4.3: Comparision of Techniques.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Classification of research articles is a big challenge for researchers. It is very important in retrieving relevant papers, and in paper submission. In literature, there are many techniques proposed for classifying the articles into predefined categories. Mostly these techniques categorize data into two types 1) Metadata based approaches, 2) Content based approaches. Mostly, researchers have used content based approaches. The main advantage of using content based approach is richness of features. But there is also a limitation of this approach, that most of the time the content of the papers are not available because the major journals like IEEE and many more does not give access to overall content of the research articles. As an alternative way, researchers have utilized metadata based approach which is freely available but it has limitation of having less number of features. The metadata of the article contain title, keywords, author name etc; while the content of the article contain introduction, heading and conclusion.

The research articles are in the form of text documents. For classification of the articles, text representation is an important task. In literature, researchers have utilized the statistical measures like TFIDF. These measures gain information using the frequency of terms. All these techniques do not take into account the semantics of the text. In literature, researchers have made manual combination of

features. But with more number of features, its complexity increases. These all problems which are mentioned above led us to propose a solution.
This study has utilized both metadata and content based approach. The freely available JUCS dataset is used for experimentation. The dataset contains all papers of the JUCS journal. The features are extracted manually from the research articles. This study has extracted eight features which are Title, Keywords, Abstract, Author first name, Author last name, Introduction, Headings, and Conclusion. After feature extraction, preprocessing is performed on the dataset. The steps involved in preprocessing are 1) tokenize all the text into words, 2) then remove all the stop words, 3) and at last do stemming of the words.

Now, the dataset is preprocessed, the next step is to do text representation. For representation of text, frequency based techniques have been used in literature but this study takes into account the semantics and context of the term used in the text. This study used Bert model for text representation and it is two way technique means it reads text from both sides of the sentence. The Bert model is already pre-trained. This study has used this pre-trained model and it generates vectors of the dataset. Now the dataset is in the form of vectors and is ready for classification task. This study has performed classification using svm classifier. The evaluation measures used for experimentations are accuracy, precision, recall and F1 score. After classification of the research articles, the next step is to do optimization of features. The genetic algorithm is used for optimizing the features. The experimentation has done by evaluating all the features and the results achieved having accuracy 90.90%, precision 0.8290, recall 0.91 and f1 score 0.8674. After doing optimization by using genetic algorithm, the results achieved having accuracy 98.15% for the features subset which include Title, Abstract, Author first name, Introduction, and Conclusion. This study compared the proposed approach with the approach proposed by Ali and Asghar. This approach used metadata features and achieved an accuracy of 78% on the JUCS dataset. The proposed approach has utilized both metadata and content of the research articles and it achieved good accuracy. Moreover, for text representation both the semantics and context of the text are considered. The feature optimization is done for reducing its complexity.

5.2 Future Work

This study has identified some of the work which is to be done in future and it is described below:

1. This study will extend this work by evaluating this technique on large dataset.
2. This study will make a semantic model which is trained on computer science domain.
3. This study can merge GA with other popular meta-heuristics.

Bibliography

[1] K. Senthamarai and N. Ramaraj, “Similarity based technique for text document classification,” *International Journal of soft computing*, vol. 3, no. 1, pp. 58–62, 2008.

[2] A. P. Santos and F. Rodrigues, “Multi-label hierarchical text classification using the acm taxonomy,” in *14th Portuguese Conference on Artificial Intelligence (EPIA)*, vol. 5, no. 5. Springer Berlin, 2009, pp. 553–564.

[3] T. Wang and B. C. Desai, “Document classification with acm subject hierarchy,” in *2007 Canadian Conference on Electrical and Computer Engineering.* IEEE, 2007, pp. 792–795.

[4] S. Godbole and S. Sarawagi, “Discriminative methods for multi-labeled classification,” in *Pacific-Asia conference on knowledge discovery and data mining.* Springer, 2004, pp. 22–30.

[5] P. K. Flynn, “Document classification in support of automated metadata extraction form heterogeneous collections,” 2014.

[6] N. Sajid, M. Afzal, and M. Qadir, “Multi-label classification of computer science documents using fuzzy logic,” *Journal of the National Science Foundation of Sri Lanka*, vol. 44, no. 2, 2016.

[7] M. T. Afzal, W.-T. Balke, H. Maurer, and N. Kulathuramaiyer, “Improving citation mining,” in *2009 First International Conference on Networked Digital Technologies.* IEEE, 2009, pp. 116–121.

[8] "Classification in machine learning: Classification algorithms," Jul 2020. [Online]. Available: https://www.edureka.co/blog/classification-in-machine-learning/

[9] J. Brownlee, "4 types of classification tasks in machine learning," Aug 2020. [Online]. Available: https://machinelearningmastery.com/types-of-classification-in-machine-learning/

[10] "Computing classification system." [Online]. Available: https://www.acm.org/publications/computing-classification-system

[11] [Online]. Available: https://dl.acm.org/ccs

[12] J. Yan, "Text representation." 2009.

[13] A. U. Dey, S. K. Ghosh, E. Valveny, and G. Harit, "Beyond visual semantics: Exploring the role of scene text in image understanding," *arXiv preprint arXiv:1905.10622*, 2019.

[14] L. Xiao, G. Wang, and Y. Zuo, "Research on patent text classification based on word2vec and lstm," in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1. IEEE, 2018, pp. 71–74.

[15] Q. Pan, H. Dong, Y. Wang, Z. Cai, and L. Zhang, "Recommendation of crowdsourcing tasks based on word2vec semantic tags," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] "Genetic algorithms for feature selection." [Online]. Available: https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection

[18] M. T. Afzal, N. Kulathuramaiyer, H. A. Maurer, and W. Balke, "Creating links into the future." *J. UCS*, vol. 13, no. 9, pp. 1234–1245, 2007.

[19] A. M. Khan, A. Shahid, M. T. Afzal, F. Nazar, F. S. Alotaibi, and K. H. Alyoubi, "Swics: Section-wise in-text citation score," *IEEE Access*, vol. 7, pp. 137 090–137 102, 2019.

[20] P. Yohan, B. Sasidhar, S. A. H. Basha, and A. Govardhan, "Automatic named entity identification and classification using heuristic based approach for telugu," *International Journal of Computer Science Issues (IJCSI)*, vol. 11, no. 1, p. 173, 2014.

[21] S. Hingmire, S. Chougule, G. K. Palshikar, and S. Chakraborti, "Document classification by topic labeling," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013, pp. 877–880.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[23] K.-C. Khor and C.-Y. Ting, "A bayesian approach to classify conference papers," in *Mexican International Conference on Artificial Intelligence.* Springer, 2006, pp. 1027–1036.

[24] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Pacific-Asia conference on knowledge discovery and data mining.* Springer, 2004, pp. 22–30.

[25] N. Sajid, M. Afzal, and M. Qadir, "Multi-label classification of computer science documents using fuzzy logic," *Journal of the National Science Foundation of Sri Lanka*, vol. 44, no. 2, 2016.

[26] T. Ali and S. Asghar, "Multi-label scientific document classification," *Journal of Internet Technology*, vol. 19, no. 6, pp. 1707–1716, 2018.

[27] R. Ahmad, M. T. Afzal, and M. A. Qadir, "Pattern analysis of citation-anchors in citing documents for accurate identification of in-text citations," *IEEE Access*, vol. 5, pp. 5819–5828, 2017.

[28] N. H. N. Le and B. Q. Ho, "A comprehensive filter feature selection for improving document classification," in *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, 2015, pp. 169–177.

[29] T. Zhou, "Automated identification of computer science research papers," 2016.

[30] W. Zong, F. Wu, L.-K. Chu, and D. Sculli, "A discriminative and semantic feature selection method for text categorization," *International Journal of Production Economics*, vol. 165, pp. 215–222, 2015.

[31] N. A. Sajid, T. Ali, M. T. Afzal, M. Ahmad, and M. A. Qadir, "Exploiting reference section to classify paper's topics," in *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, 2011, pp. 220–225.

[32] H. Piwowar, J. Priem, V. Larivière, J. P. Alperin, L. Matthias, B. Norlander, A. Farley, J. West, and S. Haustein, "The state of oa: a large-scale analysis of the prevalence and impact of open access articles," *PeerJ*, vol. 6, p. e4375, 2018.

[33] E. Loper and S. Bird, "Nltk: The natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[34] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting tf-idf term weights as making relevance decisions," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, pp. 1–37, 2008.

[35] T. Chen, Q. Mao, M. Lv, H. Cheng, and Y. Li, "Droidvecdeep: Android malware detection based on word2vec and deep belief network." *TIIS*, vol. 13, no. 4, pp. 2180–2197, 2019.

[36] M. S. Z. R. computer science graduate, "What is bert: Bert for text classification," Jun 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/

[37] [Online]. Available: http://ling.snu.ac.kr/class/AI_Agent/deep_learning_for_nlp.pdf

[38] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[40] D. Dhami, “Understanding bert-word embeddings,” Jul 2020. [Online]. Available: https://medium.com/@dhartidhami/understanding-bert-word-embeddings-7dc4d2ea54ca

[41] S. R. am a Business Analytics and I. professional with deep experience in the Indian Insurance industry. I have worked for various multi-national Insurance companies in last 7 years., “Svm: Support vector machine algorithm in machine learning,” Dec 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[42] P. Kora and P. Yadlapalli, “Crossover operators in genetic algorithms: A review,” *International Journal of Computer Applications*, vol. 162, no. 10, 2017.