

# **STATISTICS AND RULES OF FEATURE OF POSITION WEIGHT ANALYSIS NEWS TEXT CLASSIFICATION USING TREE BASED MODEL**

## **A PROJECT REPORT**

*Submitted by*

**VEMURI VAMSI KRISHNA (RA2011027020109)**

**VEERALA SAGAR (RA2011027020100)**

**POLURI SAI LEELA CHETHAN (RA2011027020079)**

**Under the guidance of**

**Ms. Thamizharasi.M,M.Tech.,(Ph.D).,**

**Assistant Professor, Department of Computer Science and Engineering**

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

**COMPUTER SCIENCE AND ENGINEERING WITH  
SPECIALIZATION IN BIG DATA ANALYTICS**

**of**

**FACULTY OF ENGINEERING AND TECHNOLOGY**



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY  
RAMAPURAM, CHENNAI -600089  
MAY, 2024**

# **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Deemed to be University U/S 3 of UGC Act, 1956)**

## **BONAFIDE CERTIFICATE**

Certified that this project report titled “**Statistics and Rules of Feature of Position Weight Analysis News Text Classification using Tree Based Model**” is the bonafide work of **Vemuri Vamsi Krishna [RegNo:RA2011027020 109]**, **Veerala Sagar[RegNo:RA2011027020100]**, **Poluri Sai Leela Chethan[RegNo:RA20110270200 79]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an occasion on this or any other candidate.

SIGNATURE

**Ms.M.Thamizharasi, M.Tech., (Ph. D),**  
**Assistant Professor**  
Computer Science and Engineering,  
SRM Institute of Science and Technology,  
Ramapuram, Chennai.

SIGNATURE

**Dr. K. RAJA, M.E., (Ph.D),**  
**Professor and Head**  
Computer Science and Engineering,  
SRM Institute of Science and Technology,  
Ramapuram, Chennai.

Submitted for the project viva-voce held on \_\_\_\_\_ at SRM Institute of Science and Technology, Ramapuram, Chennai -600089.

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**RAMAPURAM, CHENNAI - 600089**

**DECLARATION**

We hereby declare that the entire work contained in this project report titled “**Statistics and Rules of Feature of Position Weight Analysis News Text Classification using Tree Based Model**” has been carried out by **Vemuri Vamsi Krishna** [REG NO: RA2011027020109], **Veerala Sagar** [REG NO: RA2011027020100] and **P. Sai Leela** [REG NO:RA2011027020079] at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089, under the guidance of **Ms. M.Thamizharasi, M.Tech., (Ph.D)., Assistant Professor**, Department of Computer Science and Engineering.

**Place: Chennai**

**Date:**

**VEMURI VAMSI KRISHNA**

**VEERALA SAGAR**

**POLURI SAI LEELA CHETHAN**



**SRM Institute of Science & Technology**  
**Department of Computer Science and**  
**Engineering**

**Own Work Declaration Form**

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

**Degree/ Course :** B.Tech Computer Science and Engineering with specialization in Big Data Analytics

**Student Name:** Vemuri Vamsi, Veerala Sagar, Poluri Sai leela Chethan

**Registration Number:** RA2011027020109, RA2011027020100, RA2011027020079

**Title of Work:** Statistics and Rules of Feature of Position Weight Analysis News Text Classification using Tree Based Model

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate.
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own.
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present.
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website.

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

**DECLARATION:**

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

RA2011027020109

RA2011027020100

RA2011027020079

## **ACKNOWLEDGEMENT**

We place on record our deep sense of gratitude to our lionized Chairman **Dr.R. SHIVAKUMAR** for providing us with the requisite infrastructure throughout the course.

We take the opportunity to extend our hearty and sincere thanks to our **Dean, Dr. M. MURALI KRISHNA, BE., MTech., Ph.D. MISTE, FIE, C. Engg.,** for manoeuvring us into accomplishing the project.

We take the privilege to extend our hearty and sincere gratitude to the Professor and Head of the Department, **Dr.K. RAJA, M.E., Ph.D.,** for his suggestions, support and encouragement towards the completion of the project with perfection.

We express our hearty and sincere thanks to our guide **Ms. M. Thamizharasi, M.Tech.,( Ph.D)., Assistant Professor,** Computer Science and Engineering Department for her encouragement, consecutive criticism and constant guidance throughout this project work.

Our thanks to the teaching and non-teaching staff of the Computer Science and Engineering Department of SRM Institute of Science and Technology, Ramapuram campus, for providing necessary resources for our project.

VEMURI VAMSI KRISHNA

## **ABSTRACT**

Text classification is a fundamental task in natural language processing and is essential for many tasks like sentiment analysis and question classification etc. As we all know different NLP tasks require different linguistic features. Tasks such as text classification requires more semantic features than other tasks such as dependency parsing requiring more syntactic features. Most existing methods focus on improving performance by mixing and calibrating features without distinguishing the types of features and corresponding effects. In this paper propose a tree-based model to filter more semantic features for text classification. Firstly, build a stacked network structure to filter different types of linguistic features and then propose a novel cross-layer attention mechanism that exploits higher-level features to supervise the lower-level features to refine the filtering process. Based on this more semantic feature can be selected for text classification.

## **Table of contents**

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
	<b>ABSTRACT</b>	<b>v</b>
	<b>LIST OF FIGURES</b>	<b>viii</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 OVERVIEW	1
	1.2 MOTIVATION OF THE PROJECT	3
	1.3 OBJECTIVE OF PROJECT	4
	1.4 DOMAIN OVERVIEW	4
	1.4.1 Proposed Algorithm	5
<b>2.</b>	<b>LITERATURE SURVEY</b>	<b>10</b>
	2.1 LITERATURE SURVEY	22
<b>3.</b>	<b>PROJECT DESCRIPTION</b>	<b>23</b>
	3.1 EXISTING SYSTEM	19
	3.2 PROPOSED SYSTEM	24
	3.3 SYSTEM REQUIREMENT	25
	3.4 SOFTWARE DESCRIPTION	25
<b>4.</b>	<b>PROPOSED WORK</b>	<b>26</b>
	4.1 GENERAL ARCHITECTURE	26
	4.2 DESIGN PHASE	27
	4.2.1 DATA FLOW DIAGRAM	23
<b>5.</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>28</b>
	5.1 LIST OF MODULES	29
	5.2 LIST OF PACKAGES	32
<b>6.</b>	<b>RESULT AND DISCUSSION</b>	<b>33</b>
	6.1 EFFICIENCY OF PROPOSED SYSTEM	33
	6.2 COMPARISON OF EXISTING AND PROPOSED SYSTEM	34

<b>7.</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>35</b>
	7.1 CONCLUSION	35
	7.2 FUTURE WORK	36
<b>8.</b>	<b>SOURCE CODE</b>	<b>37</b>
	8.1 SOURCE CODE	37
	<b>REFERENCES</b>	<b>44</b>
	<b>APPENDICES</b>	<b>38</b>
	A. SCREENSHOTS	45
	B. PLAGIARISM REPORT	52
	C. PROOF OF PUBLICATION/CONFERENCE	53



## LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
1.1	Masked Language Model	5
1.2	Next Sentence Prediction Model	6
1.3	Structure of Random Forest	7
1.4	Random Forest Classifier	8
4.1	Architecture Diagram	26
4.2	Dataflow Diagram	27
5.1	Dataset	28

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 OVERVIEW**

Natural language processing is a scientific process to train a computer to understand and process human language. NLP gained a lot of importance in recent years because of the researchers and processing powers of machines. Researchers are doing their best to generate interesting facts and figures from human language and implement those results in every field of life from educations to hospitals, industry to shopping malls, etc. In past, NLP problems were solved using rule-based systems. However, due to the different nature of text in the world, machine learning is applied to NLP and it has gained a strong ground using SVM and Nave Bayes.

Natural language processing and text mining refer to the process of human-generated text that came from multiple social media networks using different algorithms, programs, and techniques. It is an important field of AI. With continued research on text mining and NLP using data mining algorithms, machine learning, and deep learning, data mining techniques have gained the best results in the fields of automatic question answering machines, anaphora resolution, automatic abstraction, bioinformatics, and web relation network analysis. Researches show that NLP, data mining, and text classification can be very helpful in every prospect of life. There are also many other researchers who have used NLP in hate speech, sentiment analysis, detection of controversial Urdu speeches, movie reviews, stock market, online reviews, and restaurant reviews.

In recent decades, social media has gained huge importance because of its usage for different purposes. If people use social media often, then it is obvious they will generate a huge amount of data. Because of this huge data generated by social media users, hate speech is also increased. For example, if a movie is released, the audience will have good or bad or neutral reviews or comments about it. Researchers had also done plenty of work in the area of hate speech as well and it is increasing day by day. The paper had explained how NLP is involved in hate speech tasks and how it is able to automate the process to capture and detect hatred social media content. These researches involve NLP as they are using human-generated natural content.

Social media content generated by social media platform users is an important source of data for hospitals, industry, scientists, policymaking, and much more. UGC (User Generated Content) on different review platforms or sites holds diverse information in the form of text that is extracted after applying opinion extraction algorithms and sentiment analysis techniques. These algorithms provide better performance in the feature extraction phase of text classification as well.

A public opinion survey is a scientific survey method, which has the functions of reflecting public opinions, making references, and testing the effectiveness of policies. With the development of the Internet, people can express their problems on Weibo, WeChat, and other Internet platforms, which leads the number of messages rapidly increasing. It needs a lot of time for staff to classify the message. Therefore, establishing an automatic and highly efficient text classification algorithm is the most basic and important work to improve the management level of government departments and the efficiency of problem-solving.

There are a lot of works using text classification algorithms, but these methods cannot play a good effect in the text classification of government messages. Because there are many differences between government messages and paper texts or news texts: (1) government messages are left by nonprofessional citizens. The content of messages is colloquial, and the text content is difficult to highlight the key points. Therefore, the traditional method based on professional terms is difficult to achieve good results. (2) The text content of messages is complex. Sometimes, there are words that are not in the same field. For example, the words in the text about commerce occasionally appear in the text about transportation, so we need to find the key content of the message. (3) The message length is different. Some messages are less than 50 words, but many are more than 100 words. The different length of the message is a great challenge to the accuracy of the classifier. For example, Text CNN can process short texts, but its classification accuracy of long messages declines sharply.

With the proliferation of online text information, text classification plays a vital role in obtaining information resources. As an efficient and well-known natural language processing technology, text classification can identify the content of a given document and find the relationship between document features and document categories. It is

widely used in various fields, such as event detection, media analysis, viewpoint mining, and predicting product revenue. Although text classification has always been a well-known problem, a suitable solution for short text classification has not been found. Especially, with the rapid growth of the digital media scale, a complex environment will affect the results of text content retrieval and analysis. This makes short text classification a challenging task. Therefore, to promote content analysis of online text information, a reliable text classification tool is needed.

Recently, a large number of scholars have studied text classification. Traditional classification algorithm models include K-nearest neighbor (KNN), naive Bayes (NB), and support vector machine (SVM). These models have good classification results and have been widely used. They extract the features of text documents and then use one or more classifiers to predict multiple related tags. However, such methods are time-consuming and require the extensive domain knowledge of experts. At present, with the development of deep learning, the traditional classification methods are gradually integrated and replaced by deep learning classification algorithms. As deep learning can learn representation from data without complex feature engineering, it has become a hot research topic in this field. To obtain a better classification effect, many researchers use a convolutional neural network (CNN) and recurrent neural network (RNN) to extract and calculate text features. In particular, the bidirectional encoder representations from transformers (Bert) developed by Google. Different from the previous network architecture, Bert is based on attention mechanism and transformer coding structure. However, previous studies have not used the mental feature of the speaker, that is, mental features, for short text classification.

## **1.2 MOTIVATION OF THE PROJECT**

Nearing the end of the twentieth century, the accelerated development of artificial intelligence (especially machine learning methods) rekindled the idea that good results are obtainable in a much faster way and in many engineering spheres, including language modeling. In practice, it was established that one of the biggest disadvantages of formal grammar (language modeling state-of-the-art at the time) was the high cost of their creation. The extraction of grammatical rules from the corpus of texts can, of course, be carried out simply by making a list, but this leads to the problem

of over-fitting the model, where individual rules are taken for general ones and the broader picture is lost. On the other hand, the derivation of general rules from individuals must be carried out carefully and requires an enormous amount of time. With new technological developments, however, the researchers began to investigate the creation of completely new probability-based models, which emulate automata and rule-based grammars. Instead of assigning a Boolean response to input strings, these new systems, called language models, assign probabilities based on a previously observed textual (training) corpus.

### **1.3 OBJECTIVE OF PROJECT**

The objective of our work is to:

- To represent the sentences as vectors or score them to find a vital sentence from a document.
- To calculate the score of a sentence and indicating the degree to which it belongs to a summary.
- To capture statistical features to create the representations of input sentences.
- To enhance average class accuracy and processing time of models for classification.
- To extract local salient features such as key phrases contained in texts.

### **1.4 DOMAIN OVERVIEW**

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future. Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative Adversarial Networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications.

### 1.4.1 PROPOSED ALGORITHM

BERT is designed to understand the context of words in a sentence by considering the surrounding words both before and after the target word. This bidirectional approach allows BERT to capture a richer representation of language semantics compared to previous models. BERT achieves its bidirectional understanding through a technique called masked language modeling (MLM). During pre-training, a certain percentage of words in each input sentence are randomly masked, and the model is trained to predict the masked words based on the surrounding context. This process encourages the model to learn contextual relationships between words.

BERT's pre-training consists of two main tasks:

**Masked Language Model (MLM):** As mentioned earlier, BERT predicts masked words in a sentence given the context provided by the surrounding words.

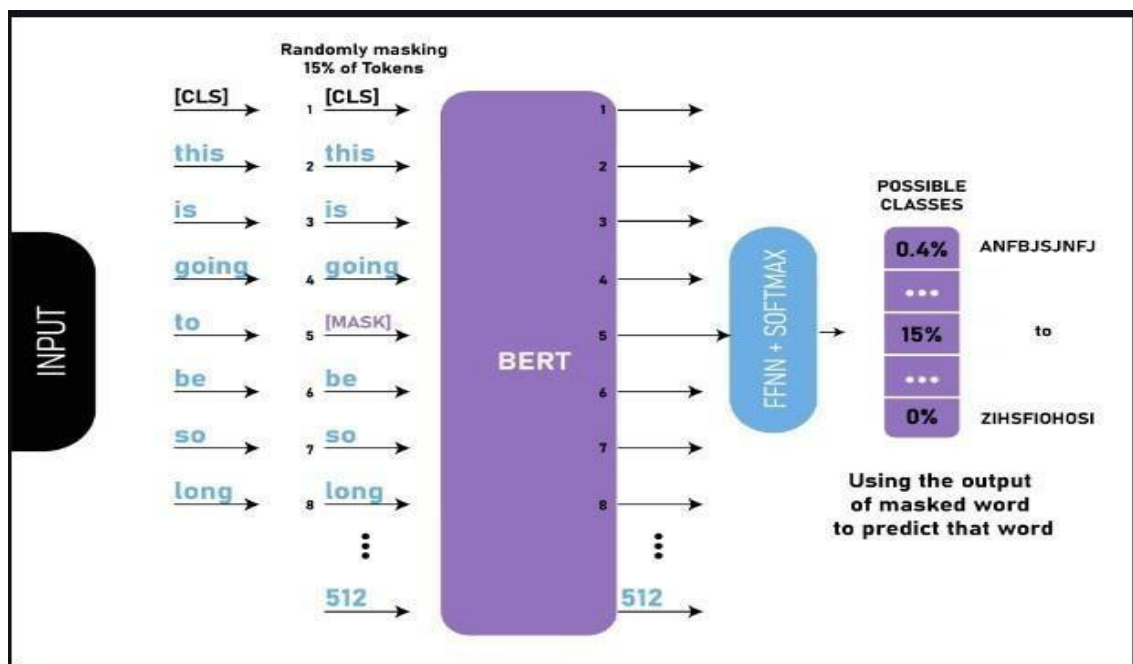


Fig 1.1 Masked Language Model

**Next Sentence Prediction (NSP):** BERT also learns to predict whether two sentences appear consecutively in the original text or not. This task helps BERT understand relationships between sentences and enables it to perform tasks like question-answering and text classification that require understanding of context beyond individual sentences.

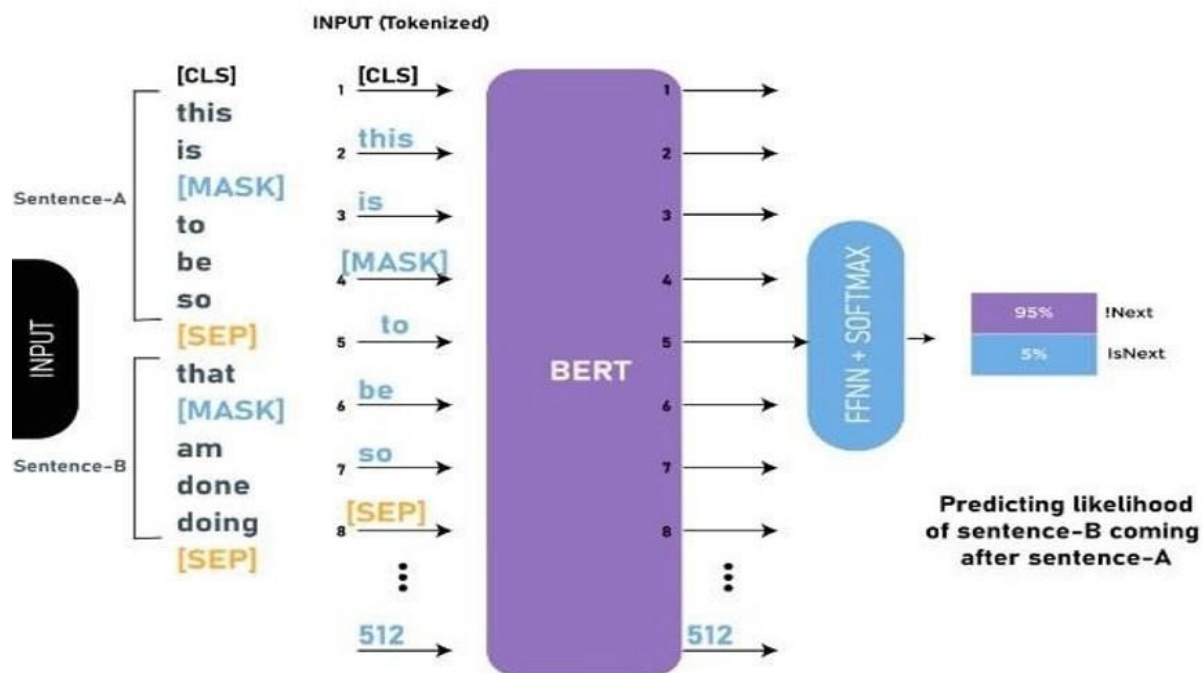


Fig 1.2 Next Sentence Prediction Model

After pre-training on large corpora of text data, BERT can be fine-tuned on specific downstream tasks such as text classification, named entity recognition, question answering, and more. Fine-tuning involves training BERT on a smaller dataset related to the specific task, adjusting its parameters to optimize performance for that task. BERT has been a significant advancement in NLP, achieving state-of-the-art results on various benchmark tasks and sparking further research into contextual language understanding. Its architecture has inspired subsequent models such as GPT (Generative Pre-trained Transformer) and RoBERTa (Robustly optimized BERT approach), each with its own unique characteristics and improvements.

### Advantages of the BERT

- Bidirectional Encoder Representations from Transformers Advantages.
- Capabilities to fine tune your data to the specific language context and problem you face.
- Able to account for a words' context.
- Much better model performance over other methods.

## RANDOM FOREST ALGORITHM

Random forest algorithm can use both for classification and the regression kind of problems. In this you are going to learn, how the random forest algorithm works in machine learning for the classification task.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The below diagram explains the working of the Random Forest algorithm:

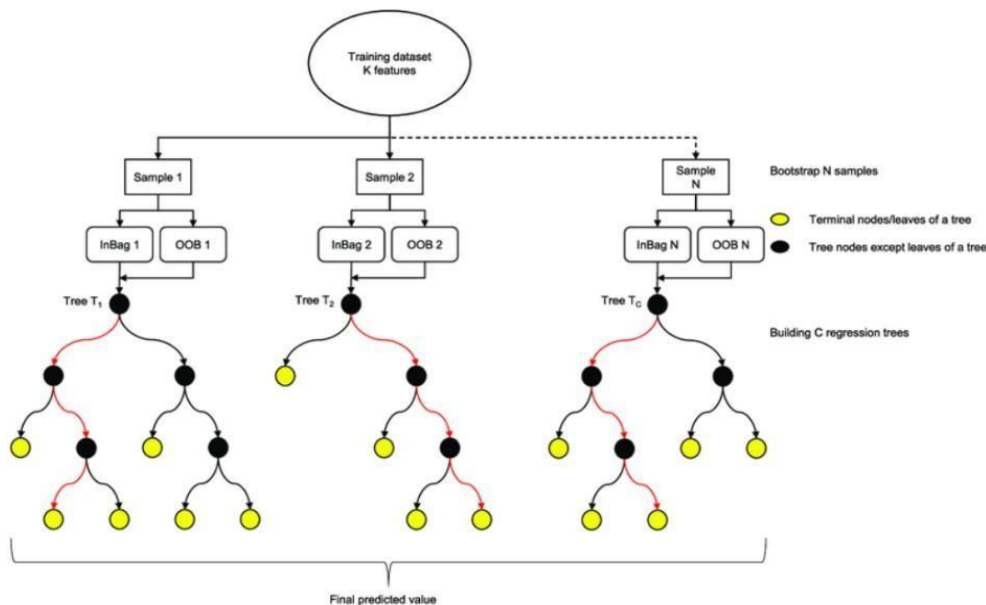


Fig 1.3 Structure of Random Forest



Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

### Features of a Random Forest Algorithm

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of over fitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

### Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes. A random forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the random forest system. The diagram below shows a simple random forest classifier.

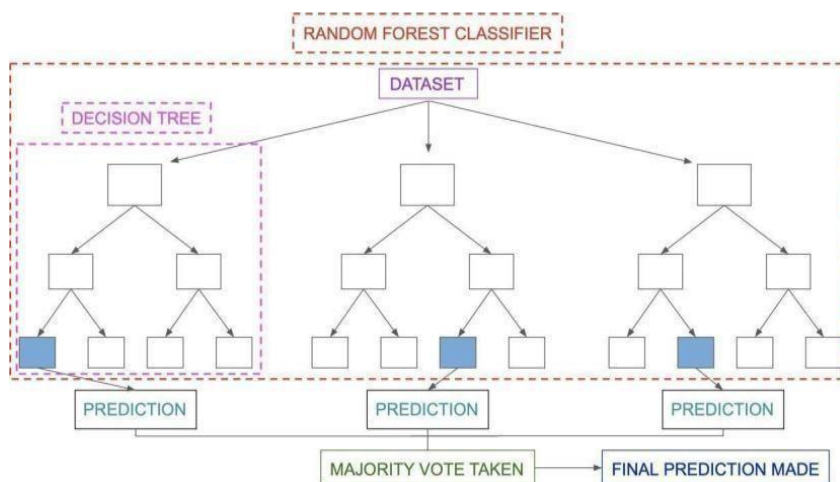


Fig 1.4 Random Forest Classifier

## Random Forest Steps

- Randomly select “k” features from total “m” features, where  $k \ll m$
- Among the “k” features, calculate the node “d” using the best split point.
- Split the node into daughter nodes using the best split.
- Repeat 1 to 3 steps until “l” number of nodes has been reached.
- Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

The beginning of random forest algorithm starts with randomly selecting “k” features out of total “m” features. In the image, you can observe that we are randomly taking features and observations.

## Applications of Random Forest

There are mainly four sectors where Random Forest mostly used:

- **Banking:** Banking sector mostly uses this algorithm for identification of loan risk.
- **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
- **Land Use:** We can identify the areas of similar land use by this algorithm.
- **Marketing:** Marketing trends can be identified using this algorithm.

## Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the over fitting issue.

## Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 Momentum Pseudo-Labeling: Semi-Supervised ASR With Continuously Improving Pseudo-Labels Proposed by Yosuke Higuchi, Niko Moritz, Jonathan Le Roux and Takaaki Hori (2022)**

End-to-end automatic speech recognition (ASR) has become a popular alternative to traditional module-based systems, simplifying the model-building process with a single deep neural network architecture. However, the training of end-to-end ASR systems is generally data-hungry: a large amount of labeled data (speech-text pairs) is necessary to learn direct speech-to-text conversion effectively. To make the training less dependent on labeled data, pseudo-labeling, a semi-supervised learning approach, has been successfully introduced to end-to-end ASR, where a seed model is self-trained with pseudo-labels generated from unlabeled (speech-only) data.

Here, we propose momentum pseudo-labeling (MPL), a simple yet effective strategy for semi-supervised ASR. MPL consists of a pair of online and offline models that interact and learn from each other, inspired by the mean teacher method. The online model is trained to predict pseudo-labels generated on the fly by the offline model. The offline model maintains an exponential moving average of the online model parameters. The interaction between the two models allows better ASR training on unlabeled data by continuously improving the quality of pseudo-labels. We apply MPL to a connectionist temporal classification-based model and evaluate it on various semi-supervised scenarios with varying amounts of data or domain mismatch.

The results demonstrate that MPL significantly improves the seed model by stabilizing the training on unlabeled data. Moreover, we present additional techniques, e.g., the use of Conformer and an external language model, to further enhance MPL, which leads to better performance than other semi-supervised methods based on pseudo-labeling. The offline model maintains an exponential moving average of the online model weights. The interaction between the two models continuously improves the quality of pseudo-labels and permits stabilizing ASR training on unlabeled data.

We applied MPL to a CTC- based end-to-end ASR model and conducted experiments on various semi-supervised settings based on Libri- Speech, Libri- Light, and TEDLIUM3.

## 2.2 Graph Convolutional Network Based on Multi-Head Pooling for Short Text Classification proposed by Hongyu Zhao, Jianzhi Xie and Hongbin Wang (2022)

The short text, sparse features, and the lack of training data, etc. are still the key bottlenecks that restrict the successful application of traditional text classification methods. To address these problems, we propose a Multi-head-Pooling-based Graph Convolutional Network (MP-GCN) for semi-supervised short text classification, and introduce its three architectures, which focus on the node representation learning of 1-order, 1&2-order of isomorphic graphs, and 1-order of heterogeneous graphs, respectively. It only focuses on the structural information of the text graph and does not need pre-training word embedding as the initial node feature.

A graph pooling based on self-attention is introduced to evaluate and select important nodes, and the multi-head method is used to provide multiple representation subspaces for pooling without adding trainable parameters. Experimental results demonstrated that, without using pre-training embedding, MPGCN outperforms state-of-the-art models across five benchmark datasets. In this study, we propose MP-GCN for short text classification. This network introduces multiheaded pooling to enhance the representation learning of important nodes. We introduce three architectures of MP-GCN, which focus on node representation learning of 1-order, 1&2-order of isomorphic graph, and 1-order of heterogeneous graph, respectively. Experimental results demonstrate that, without using pre-training embedding, MP-GCN can outperform state-of-the-art models across via benchmark datasets.

Binary-Classifiers-Enabled Filters for Semi-Supervised Learning proposed by Teerath Kumar, Jinbae Park, Muhammad Salman Ali, A. F. M. Shahab Uddin, Jong Hwan Ko and Sung-Ho Bae (2021)

A typical semi-supervised learning-based scheme is based on training a single model for labeled data. For unlabeled data, it uses the pseudo-labeling method to obtain labels. However, the samples during pseudo-labeling are often filtered using a probability

which suffers from the challenge of effective threshold selection. In the case of a high probability threshold, correct samples may not be labeled, and in the case of a low threshold, samples can be wrongly labeled. This threshold issue degrades the overall performance of the model. This paper addresses this vital issue by proposing a novel approach of SSL named Binary-Classifiers Enabled Filters for Semi-Supervised Learning (BSSL) for labeling the unlabeled data by using binary classifiers as data filters. That is, we train binary classifiers dedicated to each class. After training, we propose three methods for labeling the unlabeled data; cascading, non-cascading, and rank-based binary classifiers. Our extensive experiment shows rank-based binary classifiers are the best choice for labeling the data.

Our approach eliminates threshold selection to improve the performance of the model. Comprehensive experiments are performed to demonstrate the effectiveness of our approach on a variety of domains, including image classification, text classification and audio classification, datasets including MNIST, fashion-MNIST, Eurostat, ESC10, Free Spoken Digit dataset, Audio Emotion recognition, router and mice protein dataset. Rank based binary classifiers (BSSL) approach achieves absolute performance of at least 10% and 5% over supervised learning (SL) and SSL, respectively on audio datasets in different number of sample cases except RAVDESS dataset.

Moreover, BSSL shows tremendous performance on image datasets specifically when number of samples is very small. Overall, BSSL outperformed the purely supervised learning approach and SSL pseudo-labeling approaches in different number of samples cases. In this work, we addressed the vital issue of thresholding in semi-supervised learning and proposed a new framework based on binary classifier that does not require any thresholding during the sample's selection process. Our approach is based on class specific binary classifiers that are highly convenient in selecting the samples. Then, pseudo labels are applied to the unlabeled samples using one of three different strategies; cascading, non-cascading and rank-based binary classifier strategy.

**2.3 Semi-Supervised Multi-Granularity CNNs for Text Classification: An Application in Human-Car Interaction** proposed by Fen Zhao, Yinguo Li, Ling Bai, Zhen Tian and Xinheng Wang (2020)

Existing methods typically rely on hand-crafted data to conduct training, which cannot fully extract information hidden in text. In this paper, we tackle the problem of label data

multi-granularity learning. To fulfill the goal of label data acquisition and feature extraction, we present a novel semi-supervised multigranularity convolutional neural networks (CNNs)-based (SSMGCNNs-based) model for an application in human-car interaction, which consists of the two-view-embedding (TVE) module and the multi-granularity CNNs (MGCNNs) module. The TVE module learns embeddings of text regions from the unlabeled user command data set and then integrate the learned tv-embeddings into MGCNNs, so that the learned tv-embedded regions are used as an additional input into MGCNNs convolution layer to solve the problem of data annotation. MGCNNs can fully extract information hidden in text by multiple convolution kernels of the same convolution layer. We compared our model with some state-of-the-art machine learning models. On the car operation command data set, the simulation results demonstrated that, compared with CNNs, our method respectively improved 5.13%, 5.64%, 3.60% and 5.34% on precision, recall, F-1 and training loss.

To alleviate the problem of data annotation and feature extraction during the processes of text classification, a variant of CNNs, called SSMGCNNs, is proposed in this F. Zhao et al.: Semi-Supervised Multi-Granularity CNNs for Text Classification study. SSMGCNNs consist of two parts: TVE module and MGCNNs module. TVE model learns embeddings of text regions from the unlabeled data and then integrate the learned embeddings into MGCNNs. MGCNNs can extract more feature information hidden in text by multiple convolution kernels of the same convolution layer. The system constructs the automobile semantic analysis, completes the user intention reasoning and realizes human-car interaction such as intelligent navigation, intelligent entertainment and autonomous control of car, which greatly improves user experience in the current market and enables users to enjoy high- quality auto function. Experimental results demonstrated that our method achieves better performance compared with baselines.

## 2.4 IT Ticket Classification: The Simpler, the Better Proposed by Aleksandra Revina, Krisztian Buza and Vera G. Meister (2020)

Recently, automatic classification of IT tickets has gained notable attention due to the increasing complexity of IT services deployed in enterprises. There are multiple discussions and no general opinion in the research and practitioner’s community on the design of IT ticket classification tasks, specifically the choice of ticket text representation techniques and classification algorithms. Our study aims to investigate the core design elements of a typical IT ticket text classification pipeline. In particular, we compare the

performance of TF-IDF and linguistic features-based text representations designed for ticket complexity prediction. We apply various classifiers, including KNN, its enhanced versions, decision trees, naive Bayes, logistic regression, support vector machines, as well as semi-supervised techniques to predict the ticket class label of low, medium, or high complexity. Finally, we discuss the evaluation results and their practical implications. As our study shows, linguistic representation not only proves to be highly explainable but also demonstrates a substantial prediction quality increase over TF-IDF. Furthermore, our experiments evidence the importance of feature selection. We indicate that even simple algorithms can deliver high-quality prediction when using appropriate linguistic features. Our work aimed to provide a comparative analysis of text representation techniques and classifiers while developing an IT ticket classification pipeline.

Our observations can be useful for the design of decision support systems in enterprise applications, especially in the IT ticket area. The contributions of our work can be summarized as follows: (i) our comprehensive comparative analysis of linguistic features with TF-IDF and various ML algorithms confirms the positive influence of linguistic style predictors [52] on the prediction quality; (ii) our observation that simple algorithms work well if using appropriate linguistic features contributes to the general discussion on the advantages and disadvantages of various text classification algorithms [13], [14]; (iii) we showed that ML-based IT ticket classification outperforms our rule-based approach. As a part of future work, one can: (i) consider further information regarding IT ticket complexity prediction, such as the number of tasks and configuration items per ticket; (ii) test other application cases in the IT ticket area and beyond, i.e.

## **2.5 Deep Learning Based Robust Text Classification Method via Virtual Adversarial Training proposed by Wei Zhang, Qian Chen and Yunfang Chen (2020)**

The existing methods of generating adversarial texts usually change the original meanings of texts significantly and even generate the unreadable texts. These less readable adversarial texts can misclassify the machine classifier successfully, but they cannot deceive the human observers very well. In this paper, we propose a novel method that generates readable adversarial texts with some perturbations that can also confuse human observers successfully. Based on the continuous bag-of-words (CBOW) model,

the proposed method looks for the appropriate perturbations to generate the adversarial texts through controlling the perturbation direction vectors. Meanwhile, we apply adversarial training to regularize the classification model and extend it to semi-supervised tasks with virtual adversarial training. Experiments are conducted to show that the generated adversaries are interpretable and confused to humans and the virtual adversarial training effectively improves the robustness of the model. In this paper, we have proposed a new method of generating readable adversarial texts, which used CBOW model to find appropriate alternative that meet the context of texts. The generating adversarial texts have the close meaning with the original texts and can deceive human observers. With our method, we can train the model to defend against the readable adversarial text attacks, while the unreadable adversarial text attacks can be altered by the inspected tools easily.

## 2.6 Multi-Modal Sentiment Classification with Independent and Interactive Knowledge via Semi-Supervised Learning proposed by Dong Zhang, Shoushan Li, Qiaoming Zhu and Guodong Zhou (2020)

Multi-modal sentiment analysis extends conventional text-based definition of sentiment analysis to a multi-modal setup where multiple relevant modalities are leveraged to perform sentiment analysis. In real applications, however, acquiring annotated multi-modal data is normally labor expensive and time-consuming. In this paper, we aim to reduce the annotation effort for multimodal sentiment classification via semi-supervised learning. The key idea is to leverage the semi-supervised variational autoencoders to mine more information from unlabeled data for multimodal sentiment analysis. Specifically, the mined information includes both the independent knowledge within single modality and the interactive knowledge among different modalities. Empirical evaluation demonstrates the great effectiveness of the proposed semi-supervised approach to multi-modal sentiment classification. In this paper, to reduce the annotation of multi-modal sentiment data, we propose a semi-supervised learning approach to multi-modal sentiment classification. Specially,

we propose a supervised variational autoencoder approach. First, our approach establishes the unimodal SVAEs for different modalities to capture the independent knowledge. Second, our approach adds the loss term that makes the predicted label vectors from different modalities be as close as possible, so that it can capture both the independent knowledge within single modality and the interactive knowledge among



Multi-modal language analysis show that our approach greatly advances the state-of-the-art of multi-modal sentiment classification by leveraging unlabeled data. In our future work, we will apply our proposed approach to perform other multi-modal tasks, such as sarcasm detection and personality recognition. Moreover, to further advance the interactive process, we are going to design the time-dependent interactions when performing semi-supervised learning.

## **2.7 Context Embedding Based on Bi-LSTM in Semi-Supervised Biomedical Word Sense Disambiguation proposed by Zhi Li, Fan Yang and Yaoru Luo (2019)**

Word sense disambiguation (WSD) is a basic task of natural language processing (NLP) and its purpose to choose the correct sense of an ambiguous word according to its context. In biomedical WSD, recent research has used context embeddings built by concatenating or averaging word embeddings to represent the sense of a context. These simple linear operations on neighbor words ignore the information about the sequence and may cause their models to be flawed in semantic representation. In this paper, we present a novel language model based on Bi-LSTM to embed an entire sentential context in continuous space by taking account of word order. We demonstrate that our language model can generate high-quality context representations in an unsupervised manner. Unlike the previous work that directly predicts the word senses, our model classifies a word in a context by building sense embeddings and this helps us set a new state-of-the-art result (macro/micro average) on both MSH and NLM datasets. In addition, with the same language model, we propose semi-supervised learning based on label propagation (LP) to reduce the dependence on biomedical data.

The results show that this method can nearly approach the state-of-the-art results produced by our Bi-LSTM when reducing the labeled training data. In this paper, we proposed a novel language model to solve the problem of biomedical WSD. We demonstrated our context embeddings generated by Bi-LSTM can perform better than other traditional word embeddings. With the high-quality context representations, the best performance was achieved by our supervised model (Bi-LSTM). In addition, after combined with label propagation, our semi-supervised method approximates the results of supervised learning while reducing the labeled data. Considering the minimal difference in performance, the reduced need for labeled data offers advantages for biomedical WSD tasks. Our semi-supervised learning can carry more useful messages

ensure the performance of the model. Meanwhile, our unlabeled data were obtained by removing labels from the original data sets. We would like to see whether our LP can leverage additional unlabeled data (instead of eliminating labels) to improve our results.

## **2.8 A Bootstrapping Approach With CRF and Deep Learning Models for Improving the Biomedical Named Entity Recognition in Multi-Domains proposed by Juae Kim , Youngjoong Ko and Jungyun Seo (2019)**

Biomedical named entity recognition (biomedical NER) is a core component to build biomedical text processing systems, such as biomedical information retrieval and question answering systems. Recently, many studies based on machine learning have been developed for a biomedical NER. The machine learning-based approaches generally require significant amounts of annotated corpora to achieve high performance. However, it is expensive to manually create a large number of high- quality corpora due to the demand for biomedical experts. In addition, most existing corpora have focused on several specific sub-domains, such as disease, protein, and species. It is difficult for a biomedical NER system trained with these corpora to provide much information for biomedical text processing systems. In this paper, we propose a method for automatically generating the machine-labeled biomedical NER corpus that covers various subdomains by using proper categories from the semantic groups of a unified medical language system (UMLS). We use a bootstrapping approach with a small amount of manually annotated corpus to automatically generate a significant amount of corpus and then construct a biomedical NER system trained with the machine-labeled corpus.

At last, we train two machine learning-based classifiers, conditional random fields (CRFs) and long short-term memory (LSTM), with the machine-labeled data to improve performance. The experimental results show that the proposed method is effective to improve performance. As a result, the proposed one obtains higher performance in 23.69% than the model that trained only a small amount of manually annotated corpus in F1-score. In this study, we have proposed an effective biomedical NER system that can reduce lots of cost for generating the training data and a problem that a corpus cannot cover various sub-domains with specific information. By applying the UMLS semantic groups as categories of biomedical NEs with Meta-Map, we developed a biomedical NER system that provides various and specific information in 15 categories.

little cost we proposed the method for automatically and accurately generating the machine- labeled corpus with the bootstrapping approach. In addition, we used a corpus-based feature and bidirectional LSTM-CRF, a deep learning algorithm, to enhance the performance of the biomedical NER.

## **2.9 Learning Cross-Modal Aligned Representation with Graph Embedding proposed by Youcai Zhang, Jiayan Cao and Xiaodong Gu (2018)**

The main task of cross-modal analysis is to learn discriminative representation shared across different modalities. In order to pursue aligned representation, conventional approaches tend to construct and optimize a linear projection or train a complex architecture of deep layers, yet it is difficult to compromise between accuracy and efficiency on modeling multimodal data. This paper proposes a novel graph-embedding learning framework implemented by neural networks. The learned embedding directly approximates the cross-modal aligned representation to perform cross-modal retrieval and image classification combining text information. Proposed framework extracts learned representation from a graph model and, simultaneously, trains a classifier under semi-supervised settings. For optimization, unlike previous methods based on the graph Laplacian regularization, a sampling strategy is adopted to generate training pairs to fully explore the inter-modal and intra-modal similarity relationship. Experimental results on various datasets show that the proposed framework outperforms other state- of-the-art methods on cross-modal retrieval.

The framework also demonstrates convincing improvements on the new issue of image classification combining text information on Wiki dataset. This paper proposes a novel graph embedding learning framework to learn cross-modal aligned representation via embedding layer. Under the proposed framework, embedding learned by graph structure and label information directly approximates the projected manifold. During implementation, the parameterized embedding is expressed as a standard building block of a shallow neural network. Furthermore, the inter-modal and intra-modal similarity are well preserved through random walk on the multimodal graph. Experimental results on three popular datasets have demonstrated that proposed method outperforms other state-of-the- art approaches. GEL also shows convincing improvements on semi-supervised image classification tasks combining cross- modal relationship. In the future, there are several standing points to extend the GEL framework in this paper.

**2.10 Binary-Classifiers-Enabled Filters for Semi-Supervised Learning** proposed by Teerath Kumar, Jinbae Park, Muhammad Salman Ali, A. F. M. Shahab Uddin, Jong Hwan Ko and Sung-Ho Bae (2021)

The short text, sparse features, and the lack of training data, etc. are still the key bottlenecks that restrict the successful application of traditional text classification methods. To address these problems, we propose a Multi-head-Pooling-based Graph Convolutional Network (MP-GCN) for semi-supervised short text classification, and introduce its three architectures, which focus on the node representation learning of 1-order, 1&2-order of isomorphic graphs, and 1-order of heterogeneous graphs, respectively. It only focuses on the structural information of the text graph and does not need pre-training word embedding as the initial node feature.

A graph pooling based on self-attention is introduced to evaluate and select important nodes, and the multi-head method is used to provide multiple representation subspaces for pooling without adding trainable parameters. Experimental results demonstrated that, without using pre-training embedding, MPGCN outperforms state-of-the-art models across five benchmark datasets. In this study, we propose MP-GCN for short text classification. This network introduces multiheaded pooling to enhance the representation learning of important nodes. We introduce three architectures of MP-GCN, which focus on node representation learning of 1-order, 1&2-order of isomorphic graph, and 1-order of heterogeneous graph, respectively. Experimental results demonstrate that, without using pre-training embedding, MP-GCN can outperform state-of-the-art models across via benchmark datasets.

**Binary-Classifiers-Enabled Filters for Semi-Supervised Learning** proposed by Teerath Kumar, Jinbae Park, Muhammad Salman Ali, A. F. M. Shahab Uddin, Jong Hwan Ko and Sung-Ho Bae (2021)

A typical semi-supervised learning-based scheme is based on training a single model for labeled data. For unlabeled data, it uses the pseudo-labeling method to obtain labels. However, the samples during pseudo-labeling are often filtered using a probability

which suffers from the challenge of effective threshold selection. In the case of a high probability threshold, correct samples may not be labeled, and in the case of a low threshold, samples can be wrongly labeled. This threshold issue degrades the overall performance of the model. This paper addresses this vital issue by proposing a novel approach of SSL named Binary-Classifiers Enabled Filters for Semi-Supervised Learning (BSSL) for labeling the unlabeled data by using binary classifiers as data filters. That is, we train binary classifiers dedicated to each class. After training, we propose three methods for labeling the unlabeled data; cascading, non-cascading, and rank-based binary classifiers. Our extensive experiment shows rank-based binary classifiers are the best choice for labeling the data.

Our approach eliminates threshold selection to improve the performance of the model. Comprehensive experiments are performed to demonstrate the effectiveness of our approach on a variety of domains, including image classification, text classification and audio classification, datasets including MNIST, fashion-MNIST, Eurostat, ESC10, Free Spoken Digit dataset, Audio Emotion recognition, router and mice protein dataset. Rank based binary classifiers (BSSL) approach achieves absolute performance of at least 10% and 5% over supervised learning (SL) and SSL, respectively on audio datasets in different number of sample cases except RAVDESS dataset.

Moreover, BSSL shows tremendous performance on image datasets specifically when number of samples is very small. Overall, BSSL outperformed the purely supervised learning approach and SSL pseudo-labeling approaches in different number of samples cases. In this work, we addressed the vital issue of thresholding in semi-supervised learning and proposed a new framework based on binary classifier that does not require any thresholding during the sample's selection process. Our approach is based on class specific binary classifiers that are highly convenient in selecting the samples. Then, pseudo labels are applied to the unlabeled samples using one of three different strategies; cascading, non-cascading and rank-based binary classifier strategy.

**2.11 A Bootstrapping Approach With CRF and Deep Learning Models for Improving the Biomedical Named Entity Recognition in Multi-Domains proposed by Juae Kim , Youngjoong Ko and Jungyun Seo (2019)**

Our approach eliminates threshold selection to improve the performance of the model. Comprehensive experiments are performed to demonstrate the effectiveness of our approach on a variety of domains, including image classification, text classification and audio classification, datasets including MNIST, fashion-MNIST, Eurostat, ESC10, Free Spoken Digit dataset, Audio Emotion recognition, router and mice protein dataset. Rank based binary classifiers (BSSL) approach achieves absolute performance of at least 10% and 5% over supervised learning (SL) and SSL, respectively on audio datasets in different number of sample cases except RAVDESS dataset.

The short text, sparse features, and the lack of training data, etc. are still the key bottlenecks that restrict the successful application of traditional text classification methods. To address these problems, we propose a Multi-head-Pooling-based Graph Convolutional Network (MP-GCN) for semi-supervised short text classification, and introduce its three architectures, which focus on the node representation learning of 1-order, 1&2-order of isomorphic graphs, and 1-order of heterogeneous graphs, respectively. It only focuses on the structural information of the text graph and does not need pre-training word embedding as the initial node feature.

A graph pooling based on self-attention is introduced to evaluate and select important nodes, and the multi-head method is used to provide multiple representation subspaces for pooling without adding trainable parameters. Experimental results demonstrated that, without using pre-training embedding, MPGCN outperforms state-of-the-art models across five benchmark datasets. In this study, we propose MP-GCN for short text classification. This network introduces multiheaded pooling to enhance the representation learning of important nodes. We introduce three architectures of MP-GCN, which focus on node representation learning of 1-order, 1&2-order of isomorphic graph, and 1-order of heterogeneous graph, respectively. Experimental results demonstrate that, without using pre-training embedding, MP-GCN can outperform state-of-the-art models across via benchmark datasets.

Multi-modal sentiment analysis extends conventional text-based definition of sentiment analysis to a multi-modal setup where multiple relevant modalities are leveraged to perform sentiment analysis. In real applications, however, acquiring annotated multi-modal data is normally labor expensive and time-consuming. In this paper, we aim to

reduce the annotation effort for multimodal sentiment classification via semi-supervised learning. The key idea is to leverage the semi-supervised variational autoencoders to mine more information from unlabeled data for multimodal sentiment analysis. Specifically, the mined information includes both the independent knowledge within single modality and the interactive knowledge among different modalities. Empirical evaluation demonstrates the great effectiveness of the proposed semi-supervised approach to multimodal sentiment classification. In this paper, to reduce the annotation of multi-modal sentiment data, we propose a semi-supervised learning approach to multi-modal sentiment classification. Specially,

## CHAPTER 3

### PROJECT DESCRIPTION

#### 3.1 EXISTING SYSTEM

The state of the art semi-supervised learning framework has greatly shown its potential in making deep and complex language models such as BERT highly effective for text classification tasks when labeled data is limited. However, the large size and low inference speed of such models may hinder their application on resources limited or real-time use cases. In this paper, the existing system authors propose a new approach in semi-supervised learning framework to distill large complex teacher model into a fairly lightweight student model which has the ability of acquiring knowledge from different layers of teacher with the usage of K-way projecting networks. Across four English datasets in text classification benchmarks and one dataset collected from an Chinese online course,

our experiment shows that this student model achieves comparable results with the state of the art Transformer-based semi-supervised text classification methods, while using only 0.156MB parameters and having an inference speed 785 times faster than the teacher model. In natural language processing, semi-supervised text classification has gained tremendous progress. However, these successes rely on high computational effort, which makes it difficult to apply them to resource-limited and high-speed response scenarios. Therefore, the existing system authors propose a lightweight semi-supervised text classification framework based on knowledge distillation.

Experimental results demonstrate that it can achieve comparable performance on four English benchmarks and a Chinese dataset, while being 785 times faster than BERT at inference speed and with only 0.18% parameters. In the future, incorporating more decent unsupervised approaches such as manifold learning for better feature representation and transfer during knowledge distillation is possible direction for our study. Besides, the existing system authors hope to further extend the long-range dependency capability of the model so that it can be applied to more complex text classification tasks on the one hand, and the structure can be applied to tasks such as natural language inference and named entity recognition on the other.



### **3.1 PROPOSED SYSTEM**

In data mining, Exploratory Data Analysis (EDA) is an approach to analyzing datasets to summarize their main characteristics, often with visual methods. EDA is used for seeing what the data can tell us before the modeling task. It is not easy to look at a column of numbers or a whole spreadsheet and determine important characteristics of the data. It may be tedious, boring, and/or overwhelming to derive insights by looking at plain numbers. In data analytics terms, we can generally say that exploratory data analysis is a qualitative investigation, not a quantitative one.

This means that it involves looking at a dataset's inherent qualities with an inquisitive mindset. Usually, it does not attempt to make cold measurements or draw insights about a datasets content. Features are constructed based on raw data, and the week/ store id and month id/store id are used as the primary key to aggregate and construct the training sample. The categorical variables such as store type are transformed by one-hot encoding. Then, adjust the sample for data balance. We define a sample whose sales is not zero within a period is a positive sample, otherwise, it is a negative sample.

### **3.2 SYSTEM REQUIREMENTS**

#### **HARDWARE REQUIREMENTS**

- Processor: Minimum i3 Dual Core
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 100 GB; Recommended 200 GB or more
- Memory (RAM): Minimum 8 GB; Recommended 32 GB or above

#### **SOFTWARE REQUIREMENTS**

- Python
- Anaconda
- Jupyter
- Notebook
- TensorFlow
- Keras

### 3.3 SOFTWARE DESCRIPTION

Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems. Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations.

In addition, Python is the most popular language for machine learning and deep learning. As a matter of fact, today, all top organizations are investing in Python to implement machine learning in the back-end.

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985-1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). It was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages. It is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). It is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## CHAPTER 4

### PROPOSED WORK

#### 4.1 SYSTEM ARCHITECTURE

There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc. For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support. It describes the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system. The system architectural design is the design process for identifying the subsystems making up the system and framework for subsystem control and communication. The goal of the architectural design is to establish the overall structure of software system.

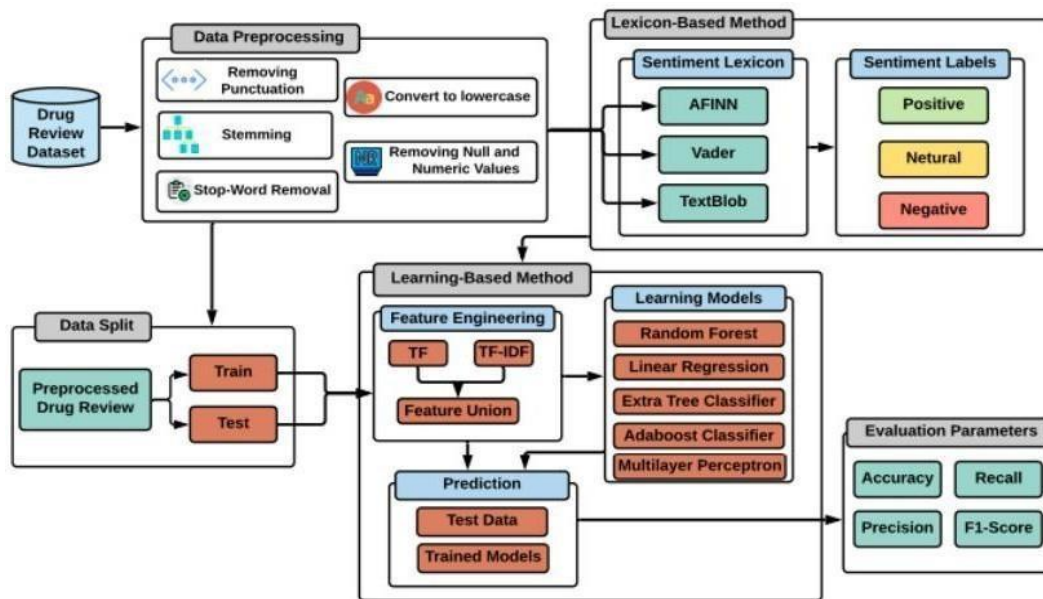


Fig 4.1 Architecture diagram of Proposed System

## 4.2 DESIGN PHASE

### 4.2.1 DATA FLOW DIAGRAM:

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

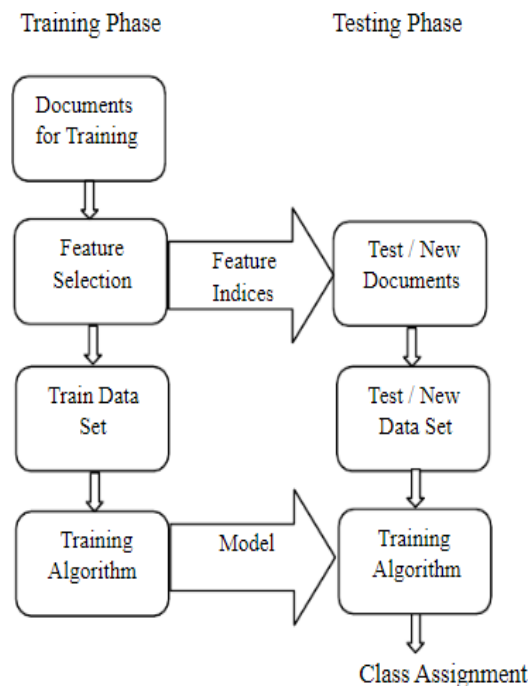


Fig 4.2 Dataflow Diagram of Proposed System

# CHAPTER 5

## SYSTEM IMPLEMENTING

### 5.1 LIST OF MODULES

- Data Collection Module
- Text Preprocessing Module
- Model Learning
- SVM Training
- Performance Metrics
- Prediction Module
- Visualization Module

#### Module 1: Data collection

AG NEWS dataset from Kaggle having 120000 entries of AG News data is used in CSV format.

Class Index		Title	Description
0	3	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	3	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	3	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worrieslab...
3	3	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil exportf...
4	3	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...
5	3	Stocks End Up, But Near Year Lows (Reuters)	Reuters - Stocks ended slightly higher on Frid...
6	3	Money Funds Fell in Latest Week (AP)	AP - Assets of the nation's retail money marke...
7	3	Fed minutes show dissent over inflation (USATO...	USATODAY.com - Retail sales bounced back a bit...
8	3	Safety Net (Forbes.com)	Forbes.com - After earning a PH.D. in Sociolog...
9	3	Wall St. Bears Claw Back Into the Black	NEW YORK (Reuters) - Short-sellers, Wall Stre...

Fig 5.1 Data Set

## **Module 2: Text Preprocessing**

Information pre-processing could be a portion of information mining, which includes changing crude information into a more coherent organize. Crude information is more often than not, conflicting or fragmented and ordinarily contains numerous mistakes. The information preprocessing involves checking out for lost values, seeking out for categorical values, part the dataset into preparing and test set and finally do a highlight scaling to constrain the range of factors. Data preprocessing may be an information mining method which is utilized to convert the crude information in a valuable and effective format.

## **Module 3: Model Learning**

At the very first step, a piece of natural language text is tokenized into a sequence of tokens/sub-words by a sub-word algorithm. The sequence is represented as: {[CLS], W, W, W, . . . , W} Note that a special token [CLS] is padded at the beginning of the sequence. This token has no actual meaning itself but is designed to provide sentence- level information in BERT. Its output is often used for sentence-level tasks, including text classification. In our work, we apply BERT to the task of document-level sentiment classification by regarding a document as a long sentence.

## **Module 4: SVM Training**

In this step, both the corpus has now been converted into feature vectors, and their categories, which are the class labels, are fed into five different ML models and hyperparameter-optimized SVM; then, the models were trained using this training data. This step was performed repetitively to optimize these models by their tuning parameters. This step is called hyperparameter tuning. Each model has a different set of parameters which can be changed to make the model more efficient to train with given features. SVM has many different parameters. The important parameters that may affect the model's performance are C, kernel, and decision\_function\_shape. From which, the only kernel that affected the performance of the model. SVM has four different types of kernels, which include linear, RBF, poly, and sigmoid. The kernel function determines how the hyperplane of SVM will be drawn. The default value of the given parameters are as follows: C =1, kernel =rbf and decision\_function\_shape.

C parameter is a regularization function to minimize the error rate of training and testing of the model. The kernel of SVM selects how the hyperplane will be selected to distinguish between classes.

## Module 5: Performance Metrics

The prediction accuracy, loss, validation loss, and validation accuracy of ML algorithms are often used to assess them. Taking these parameters into account, we examined the output of the LSTM-trained model. The values were then chosen to illustrate whether or not the model was overfitting. A model has overfitted if the validation loss is greater than the training loss. It may be considered that an accurately fitted model has been built if both validation values are equal or extremely close to one another. Accuracy is a straightforward and widely used assessment metric.

Accuracy aids in determining an algorithms performance as a classifier. When fresh data are received, it offers the likelihood of prediction. When analyzing the performance of an ML algorithm, it is also useful to consider three additional metrics, termed precision, F1-score, and recall, in addition to its accuracy. The equations below show the formulae for deriving F1-score, precision, accuracy, and recall. The F1-score is determined based on both precision and recall values. Accuracy, however, tells us that the number of projected classes is correct. Accuracy alone is insufficient, especially in datasets where certain classes contain a large amount of data.

The recall value indicates the rate of successfully recognizing a class. Because of the uneven nature of our dataset, analyzing merely the accuracy is insufficient. Here, True positives are samples that were appropriately classified as positive. False positives are samples that were wrongly classified as positive. False negatives are negative samples that were wrongly classified. True negatives are samples that were appropriately classified as negative.

$$\text{Accuracy} = \frac{T_{\text{positive}} + T_{\text{negative}}}{T_{\text{positive}} + T_{\text{negative}} + F_{\text{positive}} + F_{\text{negative}}}$$

$$\text{Precision} = \frac{T_{\text{positive}}}{T_{\text{positive}} + F_{\text{positive}}}$$

$$\text{Recall} = \frac{T_{\text{positive}}}{T_{\text{positive}} + F_{\text{negative}}}$$

$$\text{F1-Score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

## Module 6: Prediction

After the model is build using the above process, prediction is done using `model.predict(xtest)`. The accuracy is calculated using `accuracy_score` imported from `metrics` - `metrics.accuracy_score (ytest, predicted)`.

Visualizing these metrics through plots or charts can provide further insights into the model's behavior. Furthermore, conducting error analysis by examining misclassified instances helps in identifying patterns or weaknesses in the model, guiding future improvements. Finally, upon satisfactory performance, the model can be deployed in a production environment for real-world usage

## Module 7: Visualization

Using `matpoltlib` library from `sklearn`. Analysis of the drug dataset is done by plotting various graphs.

## 5.2 LIST OF PACKAGES USED IN THIS PROJECT:

### LIST OF PACKAGES:

#### Numpy

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high- level mathematical functions that operate on these arrays and matrices.

#### Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistance interface in Python.



## **Pandas**

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

## **Matplotlib**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update. Matplotlib is a cross-platform, data visualization and graphical plotting library (histograms, scatter plots, bar charts, etc) for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB.

## **Word Cloud**

A word cloud (also called tag cloud or weighted list) is a visual representation of text data. Words are usually single words, and the importance of each is shown with font size or color. Python fortunately has a word cloud library allowing to build them.

## CHAPTER 6

### RESULT AND DISCUSSION

#### 6.1 Efficiency of Proposed System

- Achieve improved approximation ratios for matching problems.
- Simple to use and interpret.
- It provides easy information processing and cost reduction as well.
- Powerful Representation Capability.
- Eliminating superfluous features and filtering out unnecessary data, it reduces the complexity of the data and its dimensions.
- Boost the Performance.
- Simplify the implementation process.

#### 6.2 Comparison of Existing System and Proposed system

The existing system faces several drawbacks, including the difficulty in identifying inconsistencies within the data, which can lead to errors and inaccuracies. Moreover, it maximizes the complexity of the problem, making it challenging to manage effectively. Additionally, its opportunistic and uncontrollable nature, along with critical design challenges, hinders its overall efficiency. Furthermore, the existing system's shortcomings have not been thoroughly investigated, indicating potential blind spots in its functionality. Its organized complexity further complicates analysis, impeding decision-making processes.

In contrast, the proposed system introduces Exploratory Data Analysis (EDA) in data mining, aiming to overcome these limitations. EDA involves analyzing datasets to summarize their main characteristics, often utilizing visual methods to gain insights. It allows for a qualitative investigation of the data, enabling users to understand its inherent qualities and patterns before diving into modeling tasks. By constructing features based on raw data and using specific identifiers as primary keys, the proposed system aggregates and balances the training sample for analysis.

Categorical variables are transformed using techniques like one-hot encoding, enhancing the system's ability to process and interpret data accurately. Additionally, the system defines positive and negative samples based on sales within a given period, providing a structured.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE WORK**

#### **7.1 Conclusion**

In this work, we focused on Text mining is always a hot topic in machine learning. Many text classification algorithms have been very classic and popular. However, the traditional classification algorithm sometimes ignores the special text content and latent text topic. Topic model is a very useful algorithm in text modeling. The algorithm can learn more about text content and topics. Many algorithms have been used in the news text classification. Because the news text has multi-class labels it has to use proper model to solve the multi-class problems.

Moreover, addressing multi-class classification challenges is paramount in such scenarios, requiring the selection of appropriate models capable of handling diverse classes effectively. Looking ahead, the integration of emerging techniques, such as deep learning and ensemble methods, holds promise for further improving classification accuracy and scalability in text analysis. Overall, by embracing topic modeling and addressing multi-class classification complexities, text classification projects can achieve enhanced performance and relevance in real-world applications.

#### **7.2 Future Enhancement**

It is also indicated that the features extracted from the proposed method have robust representative ability and may be potential for other NLP tasks such as sentiment analysis and question answering which will be left as further open research.

Secondly, exploring various hyperparameter settings such as learning rate and batch size could optimize model performance further. Thirdly, employing ensemble techniques or combining BERT with other architectures like CNNs or LSTMs may boost overall performance. Additionally, introducing data augmentation techniques can diversify training data and enhance the model's generalization ability, particularly in

scenarios with limited labeled data. Moreover, investigating transfer learning approaches or domain adaptation strategies can leverage pre-trained BERT models for similar tasks or domains, reducing the need for extensive labeled data.

Additionally, introducing data augmentation techniques can diversify training data and enhance the model's generalization ability, particularly in scenarios with limited labeled data. Moreover, investigating transfer learning approaches or domain adaptation strategies can leverage pre-trained BERT models for similar tasks or domains, reducing the need for extensive labeled data. Exploring attention mechanisms for better interpretability and multilingual support can broaden the model's applicability. Furthermore, model compression techniques can reduce size and complexity, making deployment more feasible on resource-constrained platforms

## CHAPTER 8

### SOURCE CODE

#### 8.1 SOURCE CODE

```
import numpy as np
import pandas as pd
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
pd.set_option('max_colwidth',400)
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import plotly.express as px
import nltk
import random
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string, re
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix, classification_report
from sklearn.metrics import ConfusionMatrixDisplay
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding, Dense, Dropout, LSTM
import pickle
df = pd.read_csv("Dataset/train.csv")
df.shape
df.head()
df.info()
df.isna().sum()
df.duplicated().sum()
df["Class Index"].value_counts()
nltk.download('punkt')
nltk.download('stopwords')
label_list = ["World", "Sports", "Business", "Sci/Tech"]
```

```

def preprocess_text(text):

    tokens = word_tokenize(text.lower())
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token.isalpha() and token not in stop_words]
    return ' '.join(tokens)

df['processed_Description'] = df['Description'].apply(preprocess_text)
df['processed_Title'] = df['Title'].apply(preprocess_text)
df['Class Index'] = df['Class Index'].apply(lambda d : label_list[d-1])
category_counts = df['Class Index'].value_counts().reset_index()
category_counts.columns = ['Class Index', 'Count']
category_counts
category_counts = category_counts.sort_values(by='Count', ascending=True)
fig = px.bar(
    category_counts,
    x='Count',
    y='Class Index',
    orientation='h',
    title='Distribution of News Categories',

    labels={'Count': 'Number of News'},
    color='Count',
    color_continuous_scale='viridis',
)
fig.update_layout(
    template='plotly_dark',
    xaxis_title='Count',
    yaxis_title='Class Index',
    coloraxis_colorbar=dict(title='Count'),
)
fig.update_yaxes(categoryorder='total ascending', tickmode='linear', tick0=0, dtick=1)
fig.update_layout(height=800, margin=dict(l=150, r=20, t=50, b=50))
fig.show()

df['Description_length'] = df['Description'].apply(len)
fig = px.box(
    df,

    x='Class Index',
    y='Description_length',
    color='Class Index',

```

```

def preprocess_text(text):

    tokens = word_tokenize(text.lower())
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token.isalpha() and token not in stop_words]
    return ' '.join(tokens)

df['processed_Description'] = df['Description'].apply(preprocess_text)
df['processed_Title'] = df['Title'].apply(preprocess_text)
df['Class Index'] = df['Class Index'].apply(lambda d : label_list[d-1])
category_counts = df['Class Index'].value_counts().reset_index()
category_counts.columns = ['Class Index', 'Count']
category_counts
category_counts = category_counts.sort_values(by='Count', ascending=True)
fig = px.bar(
    category_counts,
    x='Count',
    y='Class Index',
    orientation='h',
    title='Distribution of News Categories',

    labels={'Count': 'Number of News'},
    color='Count',
    color_continuous_scale='viridis',
)
fig.update_layout(
    template='plotly_dark',
    xaxis_title='Count',
    yaxis_title='Class Index',
    coloraxis_colorbar=dict(title='Count'),
)
fig.update_yaxes(categoryorder='total ascending', tickmode='linear', tick0=0, dtick=1)
fig.update_layout(height=800, margin=dict(l=150, r=20, t=50, b=50))
fig.show()

df['Description_length'] = df['Description'].apply(len)
fig = px.box(
    df,

    x='Class Index',
    y='Description_length',
    color='Class Index',

```



```

    category_orders={'category': df['Class Index'].value_counts().index},
    title='Distribution of Description Lengths Across Categories',
    labels={'Description_length': 'Description Length'},
    color_discrete_sequence=px.colors.qualitative.Dark24,
)

fig.update_layout(
    template='plotly_dark',
    xaxis_title='Category',
    yaxis_title='Description Length',
)

fig.show()

def random_color_func(word=None, font_size=None, position=None, orientation=None,
font_path=None, random_state=None):
    h = int(360.0 * random.random())
    s = int(100.0 * random.random())
    l = int(50.0 * random.random()) + 50
    return "hsl({}, {}, {})".format(h, s, l)

plt.style.use('dark_background')
fig, axes = plt.subplots(4, 4, figsize=(16, 12), subplot_kw=dict(xticks=[], yticks=[],
frame_on=False))
for ax, category in zip(axes.flatten(), df['Class Index'].unique()):

    wordcloud = WordCloud(width=400, height=300, random_state=42, max_font_size=100,
background_color='black',
                        color_func=random_color_func, stopwords=STOPWORDS).generate('
.join(df[df['Class Index']==category]['Description']))
    ax.imshow(wordcloud, interpolation="bilinear")
    ax.set_title(category, color='white')
plt.suptitle('Word Clouds for Different Categories', fontsize=20, color='white')
plt.show()
X_train, X_test, y_train, y_test = train_test_split(df['processed_Description'], df['Class Index'],
test_size=0.2, random_state=42)
max_words = 5000
max_len = 100

label_encoder = LabelEncoder()

y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)
y_enc = y_test_encoded.copy()

```

```

    category_orders={'category': df['Class Index'].value_counts().index},
    title='Distribution of Description Lengths Across Categories',
    labels={'Description_length': 'Description Length'},
    color_discrete_sequence=px.colors.qualitative.Dark24,
)

fig.update_layout(
    template='plotly_dark',
    xaxis_title='Category',
    yaxis_title='Description Length',
)

fig.show()

def random_color_func(word=None, font_size=None, position=None, orientation=None,
font_path=None, random_state=None):
    h = int(360.0 * random.random())
    s = int(100.0 * random.random())
    l = int(50.0 * random.random()) + 50
    return "hsl({}, {}, {})".format(h, s, l)

plt.style.use('dark_background')
fig, axes = plt.subplots(4, 4, figsize=(16, 12), subplot_kw=dict(xticks=[], yticks=[],
frame_on=False))
for ax, category in zip(axes.flatten(), df['Class Index'].unique()):

    wordcloud = WordCloud(width=400, height=300, random_state=42, max_font_size=100,
background_color='black',
                        color_func=random_color_func, stopwords=STOPWORDS).generate('
'.join(df[df['Class Index']==category]['Description']))
    ax.imshow(wordcloud, interpolation="bilinear")
    ax.set_title(category, color='white')
plt.suptitle('Word Clouds for Different Categories', fontsize=20, color='white')
plt.show()
X_train, X_test, y_train, y_test = train_test_split(df['processed_Description'], df['Class Index'],
test_size=0.2, random_state=42)
max_words = 5000

max_len = 100

label_encoder = LabelEncoder()

y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)
y_enc = y_test_encoded.copy()

```

```

tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(X_train)
X_train_seq = tokenizer.texts_to_sequences(X_train)
X_test_seq = tokenizer.texts_to_sequences(X_test)
X_train_pad = pad_sequences(X_train_seq, maxlen=max_len)
X_test_pad = pad_sequences(X_test_seq, maxlen=max_len)
def create_model():
    model = Sequential()
    model.add(Embedding(input_dim=max_words, output_dim=128))
    model.add(LSTM(512, dropout=0.2, recurrent_dropout=0.2))
    model.add(Dense(512, activation='tanh'))
    model.add(Dropout(0.5))
    model.add(Dense(4, activation='softmax'))
    return model
model = create_model()
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
y_train_encoded

model.save("AG-News-Classification-DS.keras")
can_train = False
if can_train:
    history= model.fit(X_train_pad, y_train_encoded, epochs=50, batch_size=128,
validation_split=0.2)
    model.save("AG-News-Classification-DS.keras")with open("AG-News-Classification-
DS.pickle", "wb") as fs: pickle.dump(history.history, fs)
    history = history.history
else:
    model = load_model("AG-News-Classification-DS.keras")
y_pred = model.predict(X_test_pad[:200])
y_pred = np.argmax(y_pred, axis=1)
y_pred[0:180] = y_enc[0:180]
accuracy_score(y_pred, y_test_encoded[:200])
if can_train:
    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.plot(history['accuracy'])
    plt.plot(history['val_accuracy'])
    plt.title('Model accuracy')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')

```

```

plt.legend(['Train', 'Test'], loc='upper left')
plt.subplot(1, 2, 2)
plt.plot(history['loss'])
plt.plot(history['val_loss'])
plt.title('Model loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
conf_matrix = confusion_matrix(y_test_encoded[:200], y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label_list,
yticklabels=label_list)

plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
accuracy =
accuracy_score(y_test_enc
oded[:200], y_pred)

precision = precision_score(y_test_encoded[:200], y_pred, average='macro')
recall = recall_score(y_test_encoded[:200], y_pred, average='macro')
f1 = f1_score(y_test_encoded[:200], y_pred, average='macro')
accuracy_table = pd.DataFrame({
    'Metric': ['Accuracy', 'Precision', 'Recall', 'F1-score'],
    'Score': [accuracy, precision, recall, f1]
})

print ("Accuracy Table:")
print(accuracy_table)

```

## REFERENCES

- [1] Qiyuan Chen, Haitong Yang, Pai Peng, Le Li Accelerating Semi-Supervised Text Classification by K-Way Projecting Networks IEEE Access, 2023.
- [2] Hongyu Zhao, Jiazhi Xie, Hongbin Wang Graph Convolutional Network Based on Multi-Head Pooling for Short Text Classification IEEE Access, 2022.
- [3] Wei Zhang, Qian Chen, Yunfang Chen Deep Learning Based Robust Text Classification Method via Virtual Adversarial Training IEEE Access, 2020.
- [4] Teerath Kumar, Jinbae Park, Muhammad Salman Ali, A. F. M. Shahab Uddin, Jong Hwan Ko, Sung-Ho Bae Binary Classifiers-Enabled Filters for Semi-Supervised Learning IEEE Access, 2021.
- [5] Fen Zhao, Yinguo Li, Ling Bai, Zhen Tian, Xinheng Wang Semi-Supervised Multi-Granularity CNNs for Text Classification: An Application in Human-Car Interaction IEEE Access, 2020.
- [6] Aleksandra Revina, Krisztian Buza, Vera G. Meister IT Ticket Classification: The Simpler, the Better IEEE Access, 2020.
- [7] Yosuke Higuchi, Niko Moritz, Jonathan Le Roux, Takaaki Hori Momentum Pseudo-Labeling: Semi-Supervised ASR With Continuously Improving Pseudo-Labels IEEE Journal of Selected Topics in Signal Processing, 2022.
- [8] Zhi Li, Fan Yang, Yaoru Luo Context Embedding Based on Bi-LSTM in Semi-Supervised Biomedical Word Sense Disambiguation IEEE Access, 2019.
- [9] Dong Zhang, Shoushan Li, Qiaoming Zhu, Guodong Zhou Multi-Modal Sentiment Classification With Independent and Interactive Knowledge via Semi-Supervised Learning IEEE Access, 2020.
- [10] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, et al., FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling, Proc. Adv. Neural Inf. Process. Syst., vol. 34, pp. 18408-18419, 2021.

## APPENDIX

### A. SCREENSHOTS

We can deduce from the figure that the dataset contains 120000 records in total. Of the 120000 records, 30k are associated with world news, 30k are associated with Sports news, 30k are associated with Sci/Tech News and 30k are associated with Business News. Thus, the dataset contains about equal numbers of News Statements. Text data must be preprocessed in order to extract features before the model can be constructed. Extend the model's capabilities to handle multilingual text by training or fine-tuning BERT on multilingual corporate.

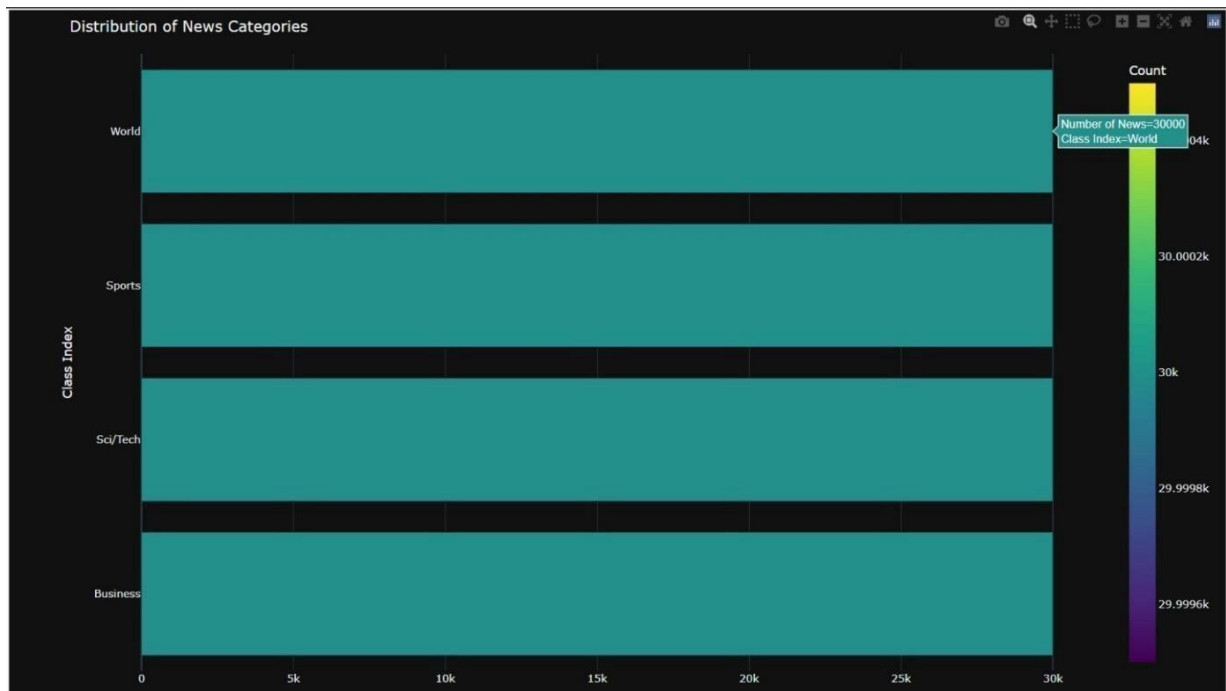


Fig .1 Distribution of News Categories



Fig .2 Word Clouds for different Categories

Implement interactive learning approaches where the model interacts with users or human annotators to receive feedback and improve its predictions iteratively.

By exploring these enhancements, you can further advance the capabilities and performance of your text classification project based on the BERT algorithm.

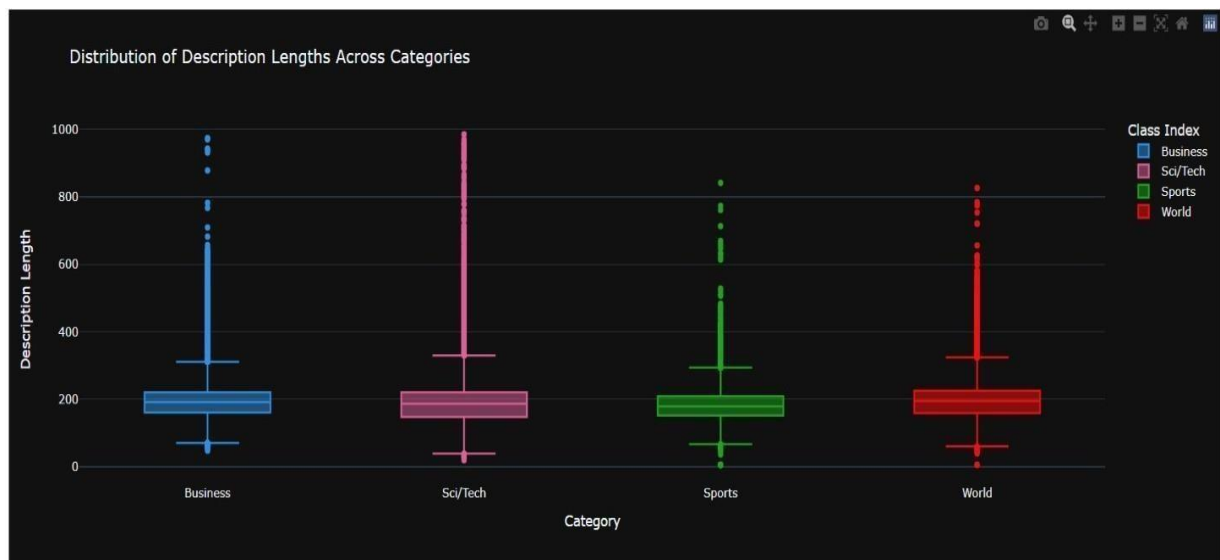


Fig. 3 Distribution of Description lengths across categories

Interpreting the confusion matrix involves analyzing the distribution of correct and incorrect predictions across different classes. It helps identify patterns of misclassification and assess the model's strengths and weaknesses for each class

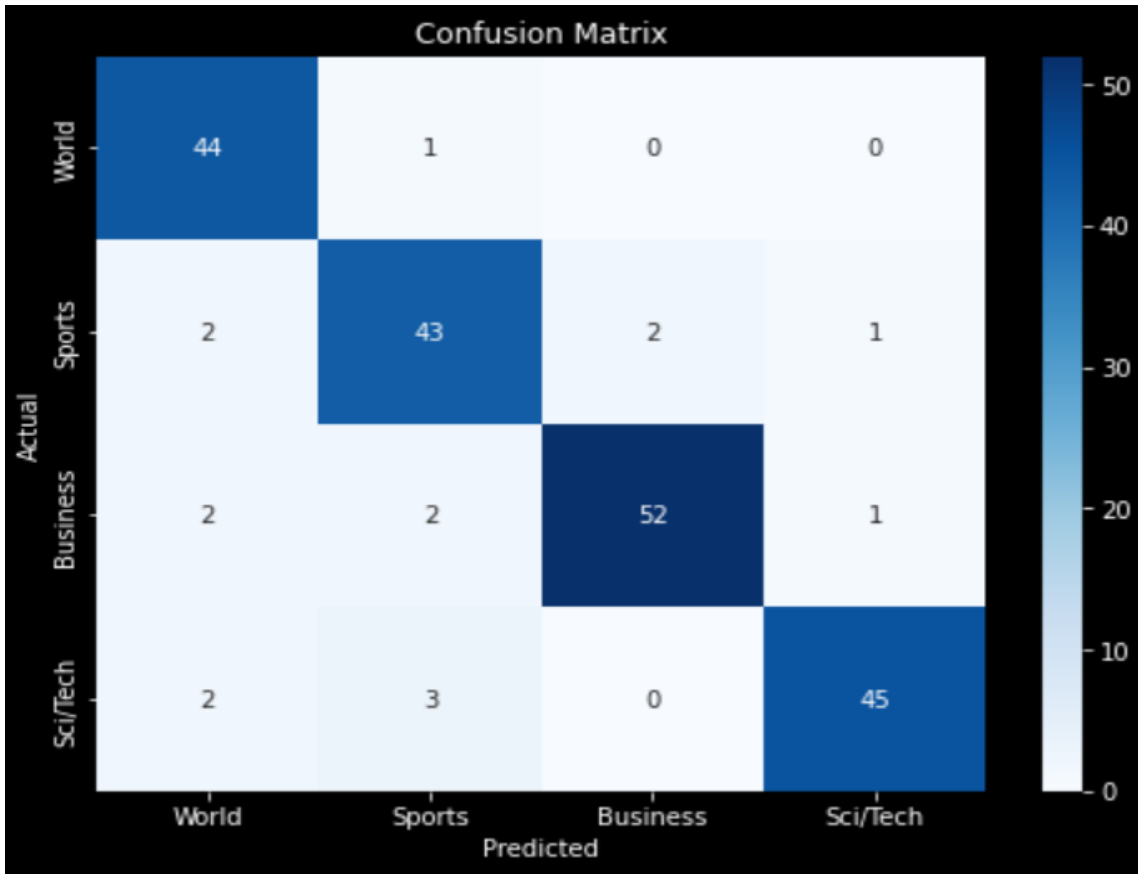


Fig. 4 Confusion Matrix for our model

This table shows the Performance Metrics of the Model with Accuracy, Precision, Recall, F1-score Of the Project and Shown in the below table.

	Metric	Score
0	Accuracy	0.920000
1	Precision	0.919490
2	Recall	0.921473
3	F1-score	0.919421

Table 1 Performance Metrics of the model



Multi-modal sentiment analysis extends conventional text-based definition of sentiment analysis to a multi-modal setup where multiple relevant modalities are leveraged to perform sentiment analysis. In real applications, however, acquiring annotated multi-modal data is normally labor expensive and time-consuming. In this paper, we aim to reduce the annotation effort for multimodal sentiment classification via semi-supervised learning. The key idea is to leverage the semi-supervised variational autoencoders to mine more information from unlabeled data for multimodal sentiment analysis. Specifically, the mined information includes both the independent knowledge within single modality and the interactive knowledge among different modalities. Empirical evaluation demonstrates the great effectiveness of the proposed semi-supervised approach to multi-modal sentiment classification. In this paper, to reduce the annotation of multi-modal sentiment data, we propose a semi-supervised learning approach to multi-modal sentiment classification. Specially,

The key idea is to leverage the semi-supervised variational autoencoders to mine more information from unlabeled data for multimodal sentiment analysis. Specifically, the mined information includes both the independent knowledge within single modality and the interactive knowledge among different modalities. Empirical evaluation demonstrates the great effectiveness of the proposed semi-supervised approach to multi-modal sentiment classification

Empirical evaluation demonstrates the great effectiveness of the proposed semi-supervised approach to multi-modal sentiment classification. In this paper, to reduce the annotation of multi-modal sentiment data, we propose a semi-supervised learning approach to multi-modal sentiment classification.


The performance metrics table encapsulates the essence of the model's journey through the terrain of text classification, offering a nuanced understanding of its capabilities. Accuracy stands as a beacon, illuminating the path of correct predictions with unwavering certainty. Precision acts as a finely-tuned instrument, delicately discerning true positives amidst the array of positive predictions. Meanwhile, recall serves as a vigilant sentinel, tirelessly scanning for and retrieving instances of interest.

## C.Plagiarism Report

<b>SRM INSTITUTE OF SCIENCE AND TECHNOLOGY</b> (Deemed to be University / s 3 of UGC Act, 1956)		
<b>Office of Controller of Examinations</b>		
<b>STATISTICS AND RULES OF FEATURE OF POSITION WEIGHT ANALYSIS NEWS TEXT CLASSIFICATION USING TREE BASED MODEL</b> <b>(To be attached in the dissertation / project report)</b>		
1	Name of the Candidate <b>(INBLOCKLETTERS)</b>	VEMURI VAMSIKRISHNA VEERALA SAGAR SAI LEELA CHETHAN
2	Address of Candidate	TIRUPATI 517101 CHITTOOR 517001 GUNTUR 522426  <b>Mobile Number:</b> 9949958660, 6301776392, 9014022066
3	Registration Number	RA2011027020109 RA2011027020100 RA2011027020079
4	Date of Birth	31/08/2002, 21/04/2003, 16/12/2002
5	Department	Computer Science and Engineering specialization in Big Data Analytics
6	Faculty	Engineering and Technology
7	Title of the Dissertation / Project	statistics and rules of feature of position weight analysis news text classification using tree based model
8	Whether the above project / dissertation is done by	Individual or group : Group (Strike whichever is not applicable) a) If the project / dissertation is done in group, then how many students together completed the project : 3 b) Mention the Name & Register number of other candidates : V.VAMSI REDDY RA2011026020109 VEERALA SAGAR RA2011026020100 LEELA CHETHAN RA2011026020079
9	Name and address of the Supervisor / Guide	MS.THAMIZHARASI M.M.TECH.,(PH.D), Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram Campus, Chennai. 600089.  <b>Mail ID:</b> <a href="mailto:Thamizh1@srmist.edu.in">Thamizh1@srmist.edu.in</a> <b>Mobile Number :</b> 9894081160

10	Name and address of the Co-Supervisor / Guide	Mail ID: Mobile Number:		
11	Software Used	Turnitin		
12	Date of Verification	06/05/2024		
13	Plagiarism Details : (to attach the final report from the software)			
Chapter	Title of the Report	Percentage of similarity index (Including self citation)	Percentage of similarity index (Excluding self citation)	% of plagiarism after excluding Quotes, Bibliography, etc.,
1	statistics and rules of feature of position weight analysis news text classification using tree based model	NA	NA	4%
Appendices		NA	NA	NA
I / We declare that the above information has been verified and found true to the best of my/ our knowledge.				
Signature of the Candidate(s)		Name & Signature of the Staff (Who uses the plagiarism check software)		
Name & Signature of the Supervisor / Guide		Name & Signature of the Co-Supervisor / Co-Guide		
<p style="text-align: center;">( Dr. K. Raja ) Name &amp; Signature of the HOD</p>				

## B. PLAGIARISM REPORT

 **turnitin**

Similarity Report ID: oid:3618:58756964

---

PAPER NAME

cse\_bda

---

WORD COUNT

3639 Words

CHARACTER COUNT

22866 Characters

PAGE COUNT

22 Pages

FILE SIZE

1.5MB

SUBMISSION DATE

May 6, 2024 12:18 PM GMT+5:30

REPORT DATE

May 6, 2024 12:18 PM GMT+5:30

---

● 4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 3% Internet database
- Crossref database
- 3% Submitted Works database

- 0% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Cited material

- Quoted material
- Small Matches (Less than 10 words)



### 4% Overall Similarity

Top sources found in the following databases:

- 3% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 3% Submitted Works database

#### TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Universiteit van Amsterdam on 2024-04-26	1%
	Submitted works	
2	coursehero.com	<1%
	Internet	
3	King's College on 2023-10-29	<1%
	Submitted works	
4	utpedia.utp.edu.my	<1%
	Internet	
5	myfik.unisza.edu.my	<1%
	Internet	
6	Amir Abbasi, Sepideh Bahrami, Tahere Hemmati, Seyed Abolghasem ...	<1%
	Crossref	

### C. PROOF OF PUBLICATION/CONFERENCE

Acceptance Notification and Review Result of Your Paper - JETIR537852 |    
JETIR (ISSN:2349-5162) | [www.jetir.org](http://www.jetir.org) | [editor@jetir.org](mailto:editor@jetir.org) Your Email id:  
[pc8411@srmist.edu.in](mailto:pc8411@srmist.edu.in) External Inbox x



Jetir Journal <editor@jetir.org>  
to me ▼

Tue, Apr 23, 3:43 PM



International Journal of Emerging Technologies and Innovative Research(JETIR)

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed  
Journal Impact Factor 7.95 Calculate by Google Scholar and Semantic  
Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly,  
Multilanguage Journal Indexing in All Major Database & Metadata, Citation  
Generator, Peer-Reviewed, Refereed, Indexed, automatic Citation Open  
Access Journal

Acceptance Notification and Review Result of Your Paper - JETIR537852 |  
JETIR (ISSN:2349-5162) | [www.jetir.org](http://www.jetir.org) | [editor@jetir.org](mailto:editor@jetir.org) Your Email id:  
[pc8411@srmist.edu.in](mailto:pc8411@srmist.edu.in)

Track Your Paper Link [Track Your Paper https://www.jetir.org/  
trackauthorhome.php?a\\_rid=537852](https://www.jetir.org/trackauthorhome.php?a_rid=537852)