A

PROJECT REPORT

On

MultiDL Hindi News Categorisation

Submitted in partial fulfillment of the requirements for the award of the degrees

Of

BACHELOR OF TECHNOLOGY in INFORMATION TECHNOLOGY

Submitted by:

Devdeep Sarkar
(300103321033)

Subhodeep Sarkar
(300103321050)

Guided by:

Mrs. Subhashini Spurjeon (Asst. Professor)

ASPIRE TO EXCEL

BHILAI INSTITUTE OF TECHNOLOGY DURG

DEPARTMENT OF INFORMATION TECHNOLOGY

UGC Autonomous Institution

(Affiliated to CSVTU, Approved by AICTE, NBA &NAAC ACCREDITED)

DURG– 491001, CHHATTISGARH, INDIA www.bitdurg.ac.in

SESSION: 2023-24

CANDIDATE'S DECLARATION

We hereby declare that the project entitled "**MultiDL Hindi News Categorisation"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in **Devdeep Sarkar, Subhodeep Sarkar** completed under the supervision of **Mrs. Subhashini Spurjeon, Asst. Professor, BIT DURG** is an authentic work.

Further, we declare that we have not submitted this work for the award of any other degree elsewhere.

Signature and name of the student(s) with date

CERTIFICATE by PROJECT Guide

It is certified that the above statement made by the students is correct to the best of our knowledge.

Signature of Guide with dates and their designation

CERTIFICATE BY THE EXAMINERS

This is to certify that the Major Project work entitled "**MultiDL Hindi News Categorisation**" is carried out by **Devdeep Sarkar and Subhodeep Sarkar** in partial fulfillment for the award of degree of **Bachelor of Technology** in **Information Technology**, **Chhattisgarh Swami Vivekanand Technical University**, **Durg** during the academic year 2023-2024.

Mrs. Subhashini Spurjeon
Internal Guide

Prof. Dr. Ani Thomas
HOD

External Examiner

ACKNOWLEDGEMENTS

We wish to acknowledge with a deep sense of hearty gratitude and indebtedness to **Mrs. Babita Verma (Assistant Professor)** of Information Technology, who gave us this opportunity to experience project work & her valuable suggestions during this project have been invaluable. We take this opportunity to voice & record our sincerest gratefulness toward our esteemed Supervisor **Mrs. Subhashini Spurjeon (Asst. Professor)** & Co–Supervisor **Dr. Ani Thomas** under whose able guidance the project work has been brought to completion.

Our heart leaps up in thankfulness for his benevolence & time for valuable suggestions, constructive criticism & active interest in the successful completion of this project work.

We are also thankful to all our honorable teachers of the Information Technology Department and our parents whose valuable support helped us and kept us motivated all through.

Devdeep Sarkar
Subhodeep Sarkar
B.Tech. VI Year
Discipline of InformationTechnology
BIT DURG

ABSTRACT

The surge in content on Hindi news platforms calls for better automated methods to organize and access it. This study aims to explore Machine Learning (ML) and Natural Language Processing (NLP) techniques to classify Hindi news articles efficiently, a field that has received limited attention in existing literature. We want to assess how well different ML and NLP models can categorize news articles accurately based on their content and context. We acquire our dataset from various online news portals to assess the effectiveness of our proposed ML techniques. Each article in the dataset is assigned specific news labels. Our classification process involves two stages: primary and secondary. The primary classification categorizes the data into binary labels – India and International – as outlined in this study. Subsequently, the secondary classification aims to further categorize the data into multi-label classes. Various algorithms, including Support Vector Machine, Logistics Regression, TF-IDF with Naïve Bayes, convolutional neural network, Bidirectional LSTM, Recurrent Neural Network, and Bert algorithm were employed for the classification of news articles. Given the complexity of the Hindi Language and the ever-changing nature of news, our goal is to contribute to the improvement of Hindi language processing techniques and offer insights into developing hybrid models for challenging text classification tasks.

Keywords: *Hindi News Platforms, Automated Methods, Machine Learning, Natural Language Processing, Classification Techniques, Binary Labels, Algorithms, Hybrid Models, Text Classification Tasks*

TABLE OF CONTENTS

List Of Figures

List Of Tables

CHAPTER-1
INTRODUCTION

Hindi underwent significant evolution over the past millennium, emerging as a prominent global language today. Its presence on online platforms is crucial, serving as a means of expression for news outlets, government bodies, and various sectors[1]. While Unicode facilitates online reading and writing, challenges persist, notably in Information Extraction and text classification. Text classification is particularly vital due to the abundance of uncategorized data. Leveraging natural language processing and machine learning, Hindi news articles can be effectively analysed and categorized, covering diverse topics such as world affairs, sports, politics, and economy.

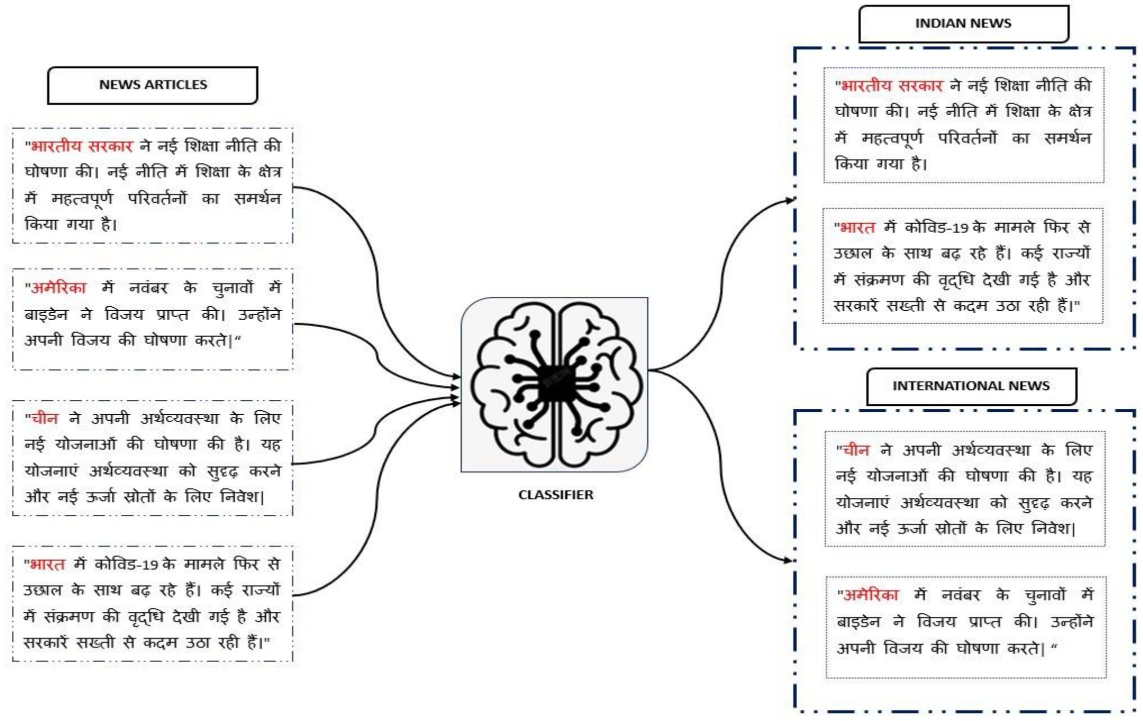
Categorization of Hindi news articles involves hierarchical classification into Indian and International at the primary level, and further subdivision into multiple categories like politics, crime, science, and sports as secondary classification. However, contextual ambiguity in Hindi text poses a challenge, necessitating the development of efficient classification techniques[2].

Currently, manual categorization of news stories is common in India, highlighting the need for automated classification systems based on contextual understanding[3].

For initial level classification, the system utilizes two predefined classes: देश (India) and वदेश (International). Pre-processing techniques like tokenization, stop word removal, feature extraction, and POS tagging are essential to prepare the dataset for machine learning models. Tokenization involves breaking text into tokens, which can be characters, subwords, or words. Stop word removal eliminates irrelevant tokens, while stemming reduces words to their root form. These processed data are then fed into classifiers.

A study evaluates various classifier models, including Support Vector Machine, TF-IDF with Naïve Bayes, Logistic Regression, convolutional neural network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Bidirectional

LSTM, and Bert, using a dataset of 3320 Hindi news articles. Results indicate superior performance of SVM and regression models in terms of accuracy, precision, recall, and F1-score.



"भारतीय सरकार ने नई शिक्षा नीति की घोषणा की। नई नीति में शिक्षा के क्षेत्र में महत्वपूर्ण परिवर्तनों का समर्थन किया गया है।

"अमेरिका में नवंबर के चुनावों में बाइडेन ने विजय प्राप्त की। उन्होंने अपनी विजय की घोषणा करते|"

"चीन ने अपनी अर्थव्यवस्था के लिए नई योजनाओं की घोषणा की है। यह योजनाएं अर्थव्यवस्था को सुदृढ़ करने और नई ऊर्जा स्रोतों के लिए निवेश|

"भारत में कोविड-19 के मामले फिर से उछाल के साथ बढ़ रहे हैं। कई राज्यों में संक्रमण की वृद्धि देखी गई है और सरकारें सख्ती से कदम उठा रही हैं।"

"भारतीय सरकार ने नई शिक्षा नीति की घोषणा की। नई नीति में शिक्षा के क्षेत्र में महत्वपूर्ण परिवर्तनों का समर्थन किया गया है।

"भारत में कोविड-19 के मामले फिर से उछाल के साथ बढ़ रहे हैं। कई राज्यों में संक्रमण की वृद्धि देखी गई है और सरकारें सख्ती से कदम उठा रही हैं।"

"चीन ने अपनी अर्थव्यवस्था के लिए नई योजनाओं की घोषणा की है। यह योजनाएं अर्थव्यवस्था को सुदृढ़ करने और नई ऊर्जा स्रोतों के लिए निवेश|

"अमेरिका में नवंबर के चुनावों में बाइडेन ने विजय प्राप्त की। उन्होंने अपनी विजय की घोषणा करते| "

Figure 1.1: Classification system with binary classes

Chapter-2
LITERATURE REVIEW

.

In the realm of text classification, various studies have been conducted to explore the efficacy of different machine learning algorithms and techniques. Shah et al. (2020) undertook a study focusing on text classification for BBC News. They evaluated the performance of three classification algorithms: K-nearest neighbor, random forest, and logistic regression. Through rigorous testing and analysis, they determined that the logistic regression classifier utilizing the TF-IDF Vectorizer feature achieved the highest accuracy. This algorithm exhibited stability, particularly in scenarios with limited data. Conversely, K-nearest neighbor algorithms demonstrated the lowest accuracy among the three methods, while random forest showed reasonable performance. The findings underscored the importance of accuracy in selecting machine learning algorithms for specific datasets [1].

Alzoubi et al. (2023) delved into text classification using five different methods and meticulously examined the results. They explored the impact of data preparation by assessing both raw and preprocessed data. Techniques such as morphological examination, standardization, and simplification were employed on the data. The study compared the effectiveness of approaches including Naive Bayes (NB), Support Vector Machines (SVM), Long Short-Term Memory (LTSM), Logistic Regression (LR), and Random Forest (RF). LTSM emerged as the most efficient method in terms of training time and accuracy, highlighting the significance of coherence between training set size, categories, and normalized data. Additionally, the choice of feature extraction method, particularly TF-IDF, significantly influenced the outcomes, with data simplification demonstrating the most significant impact among all preparation steps [2].

McCallum et al. (2001) discussed advanced methods for selecting important features in text classification, such as mutual information and information gain. They highlighted Naive Bayes as a popular algorithm for such tasks but noted its data sparsity issue. Yin et al. (2017) emphasized the growing prevalence of deep neural networks (DNNs), particularly Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), in natural language processing (NLP) tasks due to their strong expressive capability and reduced need for feature engineering [3].

Umer et al. (2022) underscored the effectiveness of CNN models for classifying both short and long texts, particularly in scenarios with balanced classes, owing to their comparable performance and computational efficiency [4].

M.Arora et al. (2022) explored three distinct feature engineering techniques—CountVectorizer, TF-IDF, and Word2Vec—in preprocessing data for machine learning algorithm training. Their experimentation revealed that Multinomial Naïve Bayes with CountVectorizer yielded superior accuracy among the tested combinations [5].

Kulkarni et al. (2022) investigated various deep learning models for Marathi text classification and found that simple single-layer models based on CNN and LSTM, combined with FastText embeddings, outperformed BERT-based models [6].

Khuntia, M. et al. (2023) focused on news headline classification using LSTM networks, word embedding, cosine similarity index, and BERT. They observed varying accuracy across different classes, with a notable decrease in accuracy for politics and health categories due to insufficient training data [7].

Chapter-3
Problem Identification

Challenges:

1. ***Limited Availability of Hindi Datasets:*** One of the primary challenges in developing a hybrid classification model for Hindi news articles is the scarcity of high-quality labeled datasets in the Hindi language. Training accurate models requires substantial and diverse data, which might be lacking in this context.
2. ***Language Complexity:*** Hindi language, like any other language, poses its own set of challenges in terms of syntax, semantics, and morphology. Handling the intricacies of Hindi text processing, including word tokenization, stemming, and part-of-speech tagging, can be challenging.
3. ***Model Adaptation:*** Many existing classification models are developed and optimized for English text. Adapting these models to work effectively with Hindi text may require significant modifications and fine-tuning.
4. ***Class Imbalance:*** The distribution of news articles across different categories might not be balanced, leading to biased models that perform well on majority classes but poorly on minority classes.
5. ***Computational Resources:*** Deep learning techniques like Bert, LSTM, and CNN are computationally intensive and may require substantial computational resources for training and inference, especially when dealing with large datasets.

Opportunities:

1. ***Unmet Demand:*** The scarcity of Hindi news classification models presents an opportunity to fill a significant gap in the market. Developing accurate and efficient models for Hindi text classification can cater to the needs of Hindi-speaking audiences and news agencies.
2. ***Emerging Technologies:*** The advancements in natural language processing (NLP) and deep learning techniques offer promising avenues for developing

sophisticated models capable of handling complex tasks such as Hindi news classification.

3. ***Language Diversity:*** Hindi is one of the most widely spoken languages globally, presenting a vast potential audience for automated news classification systems. Meeting the needs of Hindi speakers can have wide-reaching societal and economic impacts.

Hardware and Software Requirements:

Software Requirements:

1. ***Pandas:*** For data manipulation and analysis, Pandas provides powerful data structures and functions.
2. ***NLTK:*** Natural Language Toolkit (NLTK) is a popular library for text processing tasks such as tokenization, stemming, and part-of-speech tagging.
3. ***Keras:*** Keras is a high-level neural networks API, which can be used for building and training deep learning models.
4. ***Scikit-learn:*** Scikit-learn is a versatile machine learning library that offers various algorithms for classification, including SVM (Support Vector Machines).
5. ***Numpy:*** Numpy is a fundamental package for scientific computing in Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

Hardware Requirements:

1. ***High-performance Computing (HPC) Resources:*** Given the computational demands of deep learning models, access to high-performance computing resources such as GPUs (Graphics Processing Units) or TPUs (Tensor Processing Units) may be beneficial for training large-scale models efficiently.

2. ***Sufficient Memory:*** Training and running deep learning models often require significant memory resources, so systems with ample RAM are preferred.
3. ***Storage:*** Adequate storage space is necessary for storing large datasets, model checkpoints, and intermediate outputs generated during training.

Chapter - 4
Methodology

In this section, we discuss the methods implemented to carry out the various activities in the dataset cleaning and preprocessing stage and analyze the machine learning model for classifying Hindi newswire articles.

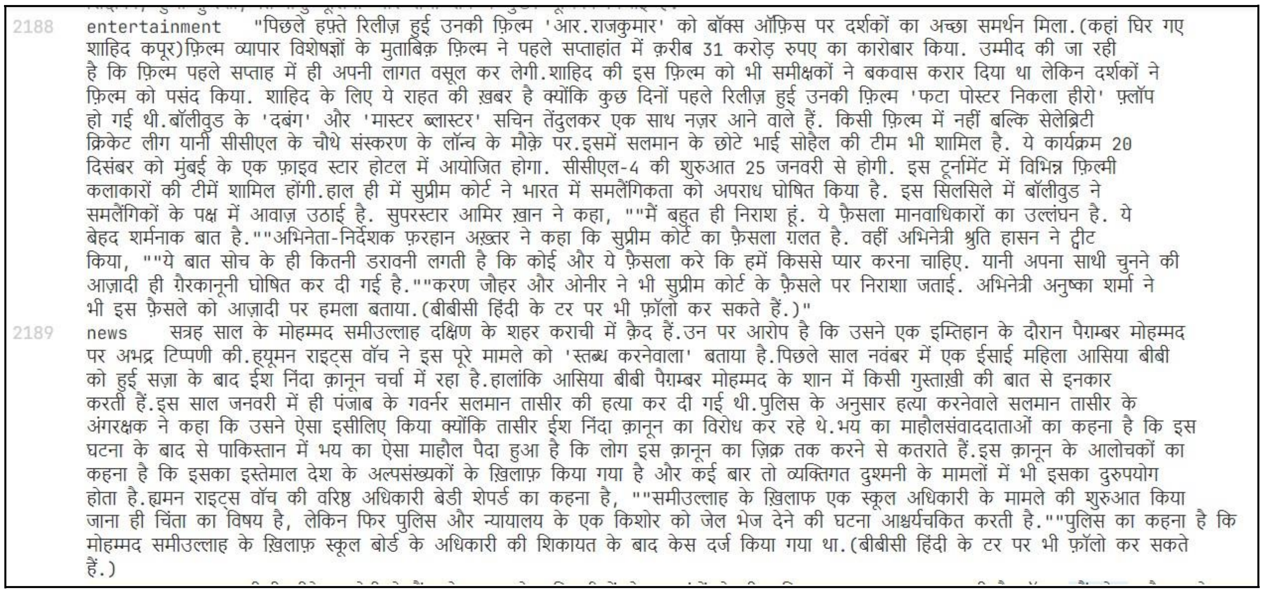
4.1. Dataset cleaning and preparation:

One of the most essential elements of NLP and machine learning is data. The System requires News Articles from Various Categories to Implement the Classification Model. Thus, these data were gathered from GitHub and Kaggle's BBC Hindi News Dataset. Some of the data were also extracted from the IIT Patna Disaster dataset. This dataset is typically utilized for more than just training. A single training data set that has previously been processed is usually divided into many segments to assess how well the model was trained. Typically, a testing data set is kept apart from the data for this particular reason. The overall dataset is about 3000 articles, and approximately 20 MB of data was used to train and test the models.

4.1.1. Text Pre-processing and feature extraction: Pre-processing raw data is a crucial step in data preparation, improving the accuracy and efficacy of a machine learning model.

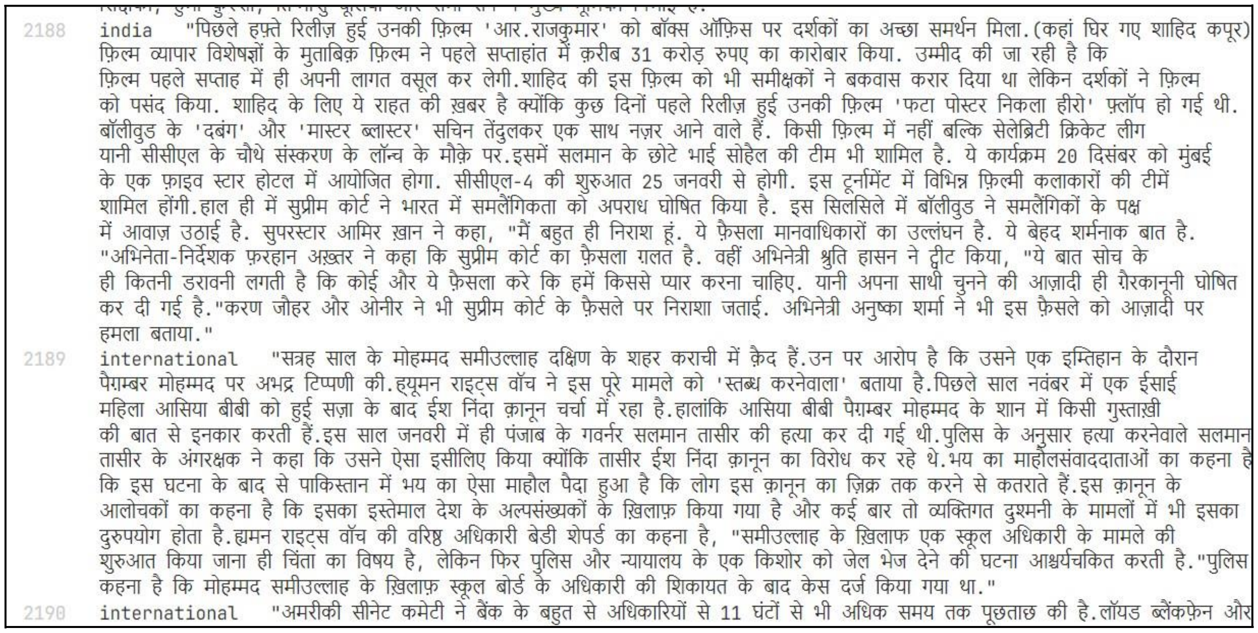
To get the dataset ready for the classification model to be applied, the following procedures were taken:

i) ***Dataset Cleaning:*** Cleaning a dataset is the process of eliminating extraneous and noisy data. The main objective is to create a uniform format for the dataset. The data was cleaned by following the steps: Cleaning the dataset started with deleting undesired, irrelevant, and empty ad lines. There are no longer any blank lines in the article; the advertisement lines have been removed, and multi-row articles have been converted into one row. The next step is to enclose each article within double quotes.



```
entertainment   "पिछले हफ़्ते रिलीज़ हुई उनकी फ़िल्म 'आर.राजकुमार' को बॉक्स ऑफ़िस पर दर्शकों का अच्छा समर्थन मिला.(कहां घिर गए
शाहिद कपूर)फ़िल्म व्यापार विशेषज्ञों के मुताबिक़ फ़िल्म ने पहले सप्ताहांत में क़रीब 31 करोड़ रुपए का कारोबार किया. उम्मीद की जा रही
है कि फ़िल्म पहले सप्ताह में ही अपनी लागत वसूल कर लेगी.शाहिद की इस फ़िल्म को भी समीक्षकों ने बकवास करार दिया था लेकिन दर्शकों ने
फ़िल्म को पसंद किया. शाहिद के लिए ये राहत की ख़बर है क्योंकि कुछ दिनों पहले रिलीज़ हुई उनकी फ़िल्म 'फटा पोस्टर निकला हीरो' फ़्लॉप
हो गई थी.बॉलीवुड के 'दबंग' और 'मास्टर ब्लास्टर' सचिन तेंदुलकर एक साथ नज़र आने वाले हैं. किसी फ़िल्म में नहीं बल्कि सेलेब्रिटी
क्रिकेट लीग यानी सीसीएल के चौथे संस्करण के लॉन्च के मौक़े पर.इसमें सलमान के छोटे भाई सोहैल की टीम भी शामिल है. ये कार्यक्रम 20
दिसंबर को मुंबई के एक फ़ाइव स्टार होटल में आयोजित होगा. सीसीएल-4 की शुरुआत 25 जनवरी से होगी. इस टूर्नामेंट में विभिन्न फ़िल्मी
कलाकारों की टीमें शामिल होंगी.हाल ही में सुप्रीम कोर्ट ने भारत में समलैंगिकता को अपराध घोषित किया है. इस सिलसिले में बॉलीवुड ने
समलैंगिकों के पक्ष में आवाज़ उठाई है. सुपरस्टार आमिर ख़ान ने कहा, ""मैं बहुत ही निराश हूं. ये फ़ैसला मानवाधिकारों का उल्लंघन है. ये
बेहद शर्मनाक बात है.""अभिनेता-निर्देशक फ़रहान अख़्तर ने कहा कि सुप्रीम कोर्ट का फ़ैसला ग़लत है. वहीं अभिनेत्री श्रुति हासन ने ट्वीट
किया, ""ये बात सोच के ही कितनी डरावनी लगती है कि कोई और ये फ़ैसला करे कि हमें किससे प्यार करना चाहिए. यानी अपना साथी चुनने की
आज़ादी ही ग़ैरकानूनी घोषित कर दी गई है.""करण जौहर और ओनीर ने भी सुप्रीम कोर्ट के फ़ैसले पर निराशा जताई. अभिनेत्री अनुष्का शर्मा ने
भी इस फ़ैसले को आज़ादी पर हमला बताया.(बीबीसी हिंदी के टर पर भी फ़ॉलो कर सकते हैं.)"
news    सत्रह साल के मोहम्मद समीउल्लाह दक्षिण के शहर कराची में क़ैद हैं.उन पर आरोप है कि उसने एक इम्तिहान के दौरान पैग़म्बर मोहम्मद
पर अभद्र टिप्पणी की.ह्यूमन राइट्स वॉच ने इस पूरे मामले को 'स्तब्ध करनेवाला' बताया है.पिछले साल नवंबर में एक ईसाई महिला आसिया बीबी
को हुई सज़ा के बाद ईश निंदा क़ानून चर्चा में रहा है.हालांकि आसिया बीबी पैग़म्बर मोहम्मद के शान में किसी गुस्ताख़ी की बात से इनकार
करती हैं.इस साल जनवरी में ही पंजाब के गवर्नर सलमान तासीर की हत्या कर दी गई थी.पुलिस के अनुसार हत्या करनेवाले सलमान तासीर के
अंगरक्षक ने कहा कि उसने ऐसा इसीलिए किया क्योंकि तासीर ईश निंदा क़ानून का विरोध कर रहे थे.भय का माहौलसंवाददाताओं का कहना है कि इस
घटना के बाद से पाकिस्तान में भय का ऐसा माहौल पैदा हुआ है कि लोग इस क़ानून का ज़िक्र तक करने से कतराते हैं.इस क़ानून के आलोचकों का
कहना है कि इसका इस्तेमाल देश के अल्पसंख्यकों के ख़िलाफ़ किया गया है और कई बार तो व्यक्तिगत दुश्मनी के मामलों में भी इसका दुरुपयोग
होता है.ह्यमन राइट्स वॉच की वरिष्ठ अधिकारी बेडी शेपर्ड का कहना है, ""समीउल्लाह के ख़िलाफ एक स्कूल अधिकारी के मामले की शुरुआत किया
जाना ही चिंता का विषय है, लेकिन फिर पुलिस और न्यायालय के एक किशोर को जेल भेज देने की घटना आश्चर्यचकित करती है.""पुलिस का कहना है कि
मोहम्मद समीउल्लाह के ख़िलाफ़ स्कूल बोर्ड के अधिकारी की शिकायत के बाद केस दर्ज किया गया था.(बीबीसी हिंदी के टर पर भी फ़ॉलो कर सकते
हैं.)
```

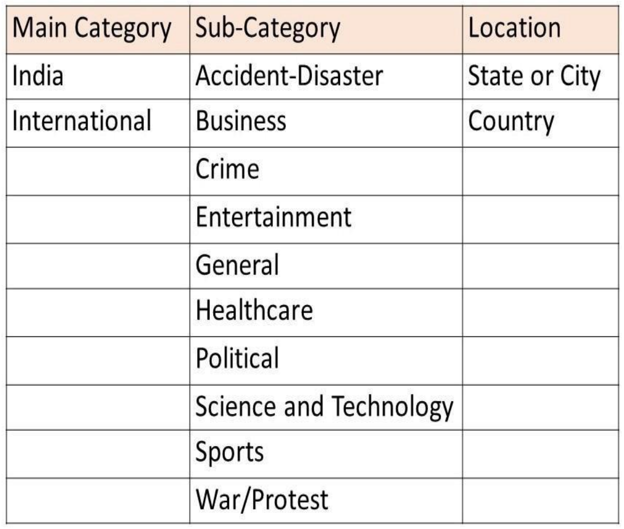
Figure 4.1: Dataset Before removing blank spaces and unwanted lines



```
india   "पिछले हफ़्ते रिलीज़ हुई उनकी फ़िल्म 'आर.राजकुमार' को बॉक्स ऑफ़िस पर दर्शकों का अच्छा समर्थन मिला.(कहां घिर गए शाहिद कपूर)
फ़िल्म व्यापार विशेषज्ञों के मुताबिक़ फ़िल्म ने पहले सप्ताहांत में क़रीब 31 करोड़ रुपए का कारोबार किया. उम्मीद की जा रही है कि
फ़िल्म पहले सप्ताह में ही अपनी लागत वसूल कर लेगी.शाहिद की इस फ़िल्म को भी समीक्षकों ने बकवास करार दिया था लेकिन दर्शकों ने फ़िल्म
को पसंद किया. शाहिद के लिए ये राहत की ख़बर है क्योंकि कुछ दिनों पहले रिलीज़ हुई उनकी फ़िल्म 'फटा पोस्टर निकला हीरो' फ़्लॉप हो गई थी.
बॉलीवुड के 'दबंग' और 'मास्टर ब्लास्टर' सचिन तेंदुलकर एक साथ नज़र आने वाले हैं. किसी फ़िल्म में नहीं बल्कि सेलेब्रिटी क्रिकेट लीग
यानी सीसीएल के चौथे संस्करण के लॉन्च के मौक़े पर.इसमें सलमान के छोटे भाई सोहैल की टीम भी शामिल है. ये कार्यक्रम 20 दिसंबर को मुंबई
के एक फ़ाइव स्टार होटल में आयोजित होगा. सीसीएल-4 की शुरुआत 25 जनवरी से होगी. इस टूर्नामेंट में विभिन्न फ़िल्मी कलाकारों की टीमें
शामिल होंगी.हाल ही में सुप्रीम कोर्ट ने भारत में समलैंगिकता को अपराध घोषित किया है. इस सिलसिले में बॉलीवुड ने समलैंगिकों के पक्ष
में आवाज़ उठाई है. सुपरस्टार आमिर ख़ान ने कहा, "मैं बहुत ही निराश हूं. ये फ़ैसला मानवाधिकारों का उल्लंघन है. ये बेहद शर्मनाक बात है.
"अभिनेता-निर्देशक फ़रहान अख़्तर ने कहा कि सुप्रीम कोर्ट का फ़ैसला ग़लत है. वहीं अभिनेत्री श्रुति हासन ने ट्वीट किया, "ये बात सोच के
ही कितनी डरावनी लगती है कि कोई और ये फ़ैसला करे कि हमें किससे प्यार करना चाहिए. यानी अपना साथी चुनने की आज़ादी ही ग़ैरकानूनी घोषित
कर दी गई है."करण जौहर और ओनीर ने भी सुप्रीम कोर्ट के फ़ैसले पर निराशा जताई. अभिनेत्री अनुष्का शर्मा ने भी इस फ़ैसले को आज़ादी पर
हमला बताया."
international   "सत्रह साल के मोहम्मद समीउल्लाह दक्षिण के शहर कराची में क़ैद हैं.उन पर आरोप है कि उसने एक इम्तिहान के दौरान
पैग़म्बर मोहम्मद पर अभद्र टिप्पणी की.ह्यूमन राइट्स वॉच ने इस पूरे मामले को 'स्तब्ध करनेवाला' बताया है.पिछले साल नवंबर में एक ईसाई
महिला आसिया बीबी को हुई सज़ा के बाद ईश निंदा क़ानून चर्चा में रहा है.हालांकि आसिया बीबी पैग़म्बर मोहम्मद के शान में किसी गुस्ताख़ी
की बात से इनकार करती हैं.इस साल जनवरी में ही पंजाब के गवर्नर सलमान तासीर की हत्या कर दी गई थी.पुलिस के अनुसार हत्या करनेवाले सलमान
तासीर के अंगरक्षक ने कहा कि उसने ऐसा इसीलिए किया क्योंकि तासीर ईश निंदा क़ानून का विरोध कर रहे थे.भय का माहौलसंवाददाताओं का कहना है
कि इस घटना के बाद से पाकिस्तान में भय का ऐसा माहौल पैदा हुआ है कि लोग इस क़ानून का ज़िक्र तक करने से कतराते हैं.इस क़ानून के
आलोचकों का कहना है कि इसका इस्तेमाल देश के अल्पसंख्यकों के ख़िलाफ़ किया गया है और कई बार तो व्यक्तिगत दुश्मनी के मामलों में भी इसका
दुरुपयोग होता है.ह्यमन राइट्स वॉच की वरिष्ठ अधिकारी बेडी शेपर्ड का कहना है, "समीउल्लाह के ख़िलाफ एक स्कूल अधिकारी के मामले की
शुरुआत किया जाना ही चिंता का विषय है, लेकिन फिर पुलिस और न्यायालय के एक किशोर को जेल भेज देने की घटना आश्चर्यचकित करती है."पुलिस
कहना है कि मोहम्मद समीउल्लाह के ख़िलाफ़ स्कूल बोर्ड के अधिकारी की शिकायत के बाद केस दर्ज किया गया था."
international   "अमरीकी सीनेट कमेटी ने बैंक के बहुत से अधिकारियों से 11 घंटों से भी अधिक समय तक पूछताछ की है.लॉयड ब्लैंकफ़ेन और
```

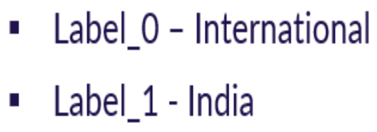
Figure 4.2: Dataset Before removing blank spaces and unwanted lines

ii) ***Dataset Classification:*** Table shows the categories and subcategories created in the dataset for the classification process. Figure shows the categories and subcategories converted to Hindi Language. Then, the articles are manually tagged in each data row according to the type of categories and sub-categories. Each article unit is categorized into three columns, as shown in figure.



Main Category	Sub-Category	Location
India	Accident-Disaster	State or City
International	Business	Country
	Crime	
	Entertainment	
	General	
	Healthcare	
	Political	
	Science and Technology	
	Sports	
	War/Protest	

Table 4.1: Summary of Categorization



- Label_0 – International
- Label_1 - India

Figure 4.3: Classification to tag news articles in Hindi

iii) ***Preprocessing:*** Machine Learning and NLP techniques can efficiently organize Hindi Newswire articles into various sections. Tokenization, Embedding, stopword removable, Text filtering & cleaning, and vectorization are techniques used in binary classification.

4.2. Binary Classification of News:

The system used the binary classification strategy to divide the Hindi Newswire articles into India and International classes. Figure shows the block diagram of the overall categorization system. Two modules comprise the entire system. Machine and deep learning algorithms will be combined with the preprocessing methods mentioned in this work to create a text classification model for news articles. The first module covers preparing and preprocessing the dataset, and the second covers creating a comprehensive machine-learning model. This work focuses on classifying Hindi texts because little research has been done.

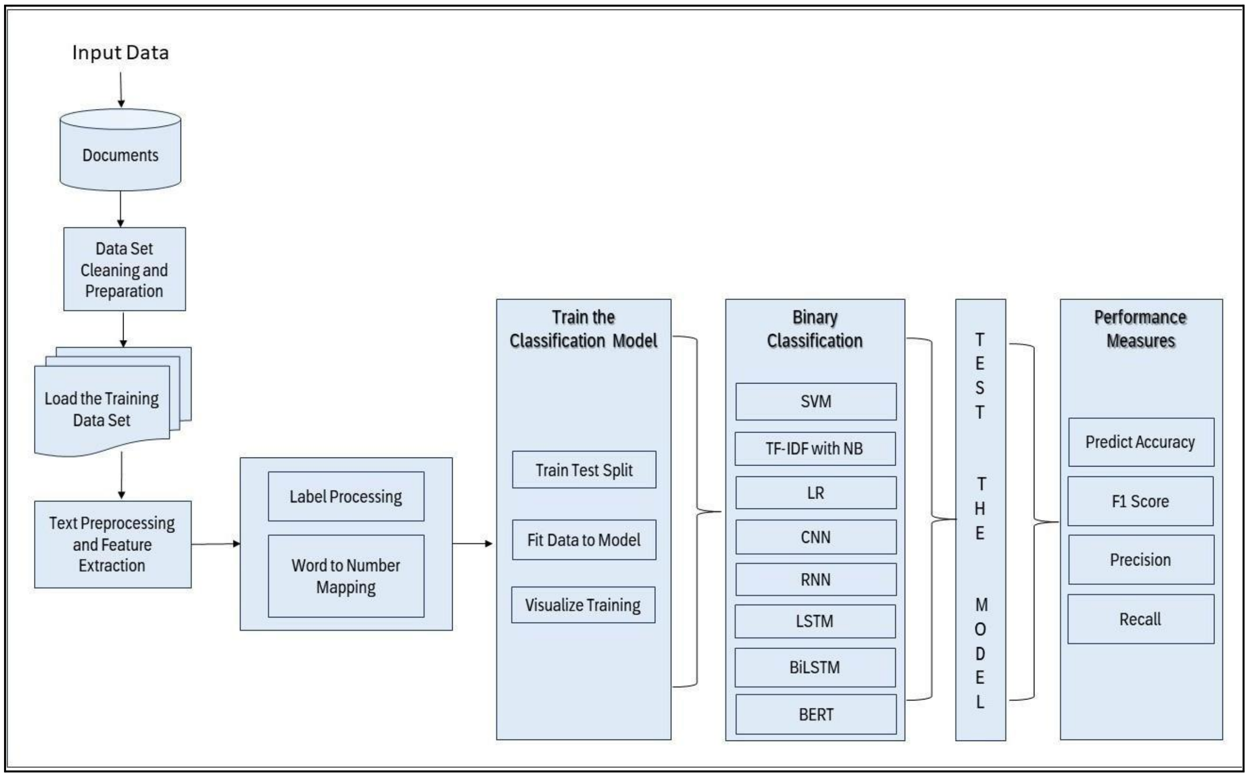


Figure 4.4: Block Diagram for Newswire Article Binary Classification

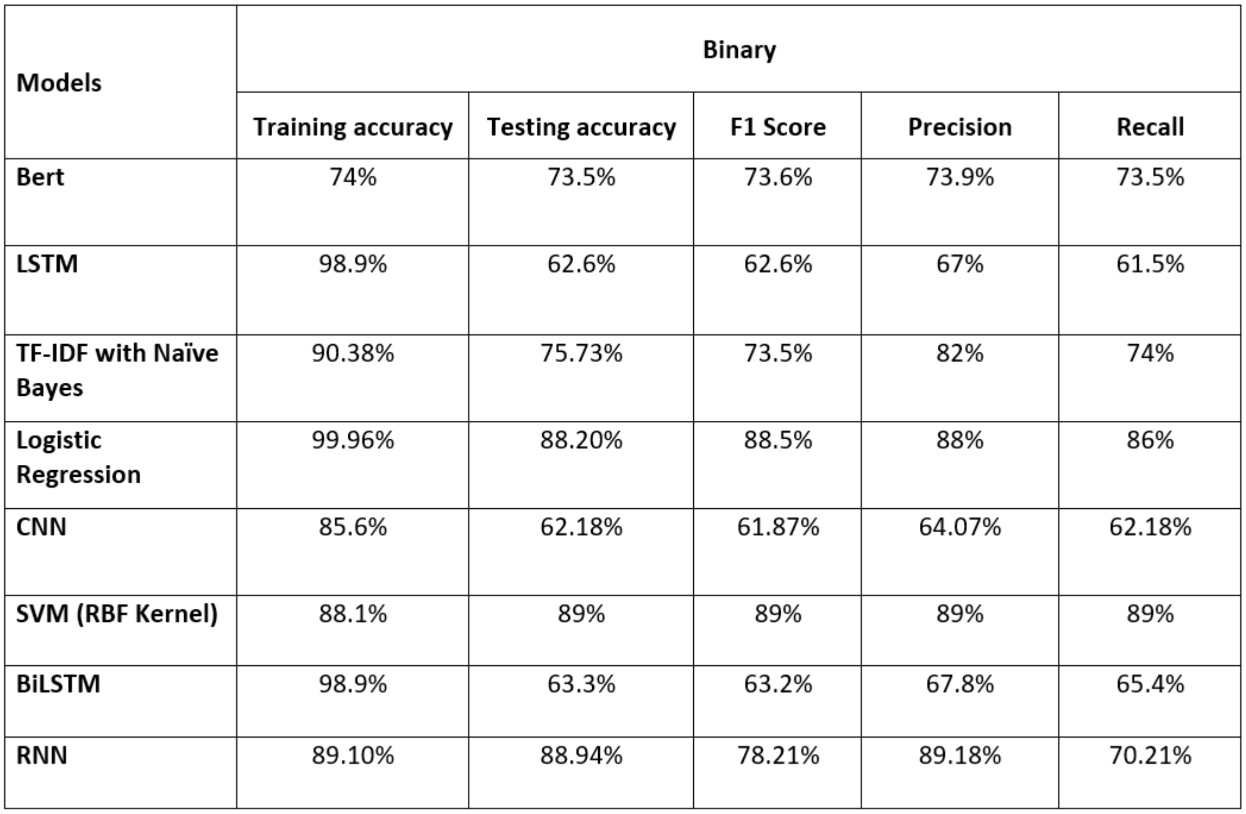
The proposed binary classification model was developed using several algorithms, including the RNN, LSTM, BiLSTM, BeRT technique, CNN, NB, LR, and SVM classifier. The SVM model is prepared with the Radial Basis Function kernel and hyperparameters to capture the linear relationships between the words and the news classes. It also helps make the classification very fast and practical. Stopwords and tokenizer methods are used in data preprocessing, and for feature extraction, TF-IDF vectorizer is implemented with specified parameters. The SVM model performs about 89% and above testing accuracy on the sample data predictions. The LR is used on vectorized training data. Two functions can be utilized in this statistical technique, which will work well for binary classification. The logistic function, the sigmoid function, and binary values. The LR performs 99.96%

on the training dataset and 88.20% on testing the dataset. The multinomial NB classifier with a TF-IDF vectorizer with a specified alpha value of 0.7 is used. At first, Naive Bayes relies on the size of the dataset. Still, as we add more data, its performance eventually reaches a plateau, and more training data is needed to improve the Naive Bayes classifier's performance. However, we are getting improved accuracy if we increase the dataset. Our dataset's performance is recorded as 75.73% of testing accuracy. Text classification issues involving binary and multi-class data are handled by CNNs[12]. Pre-trained Word embeddings are used to create an embedding matrix. Model Accuracy of 62.18% is calculated. Despite its remarkable effectiveness in gathering features for relation extraction, CNN only gathers local features and ignores the long-distance dependency between the nominal pairings. RNN can be the solution to this problem. The model is prepared using binary cross-entropy loss, RMSprop optimizer, and accuracy and precision metrics. Tokenizer is also used in the initial preprocessing of the dataset. The testing results are 88.94%. Interpreting an RNN's output can be challenging, particularly with complex inputs like natural language. Because of this, it could be challenging to comprehend how the network generates its predictions.

Ordinary RNN models perform poorly and cannot support text with complicated sequence memory. Among the several RNN variations, LSTM and BiLSTM are the most often utilized models. A sequential model is created, and an embedding layer for word embeddings is added. Dropout for overfitting and Dense layers for classification were used. model is used with binary cross entropy loss and Adam optimizer and gets 62% testing accuracy results. BERT uses transfer learning to understand the context of textual data fully. Attention masks are created, and inputs, masks, and labels are given to PyTorch tensors. For training and validation sets, batch size and data loaders are created, and then a pre-trained BERT model is used for sequence classification, and good results in testing accuracy of 73% are obtained.

Approx: 3000+

MB: 20MB data



Models	Binary				
	Training accuracy	Testing accuracy	F1 Score	Precision	Recall
Bert	74%	73.5%	73.6%	73.9%	73.5%
LSTM	98.9%	62.6%	62.6%	67%	61.5%
TF-IDF with Naïve Bayes	90.38%	75.73%	73.5%	82%	74%
Logistic Regression	99.96%	88.20%	88.5%	88%	86%
CNN	85.6%	62.18%	61.87%	64.07%	62.18%
SVM (RBF Kernel)	88.1%	89%	89%	89%	89%
BiLSTM	98.9%	63.3%	63.2%	67.8%	65.4%
RNN	89.10%	88.94%	78.21%	89.18%	70.21%

Table 4.2: COMPARISON RESULTS OF DIFFERENT MACHINE AND DEEP LEARNING MODELS

4.3. Training the hybrid model:

We thoroughly compared several machine learning (ML) and natural language processing (NLP) models for the binary categorization of Hindi news items in Table 2 of our study. By carefully analyzing parameters like recall, accuracy, precision, and F-score, we were able to identify the best models in each category. Building on these discoveries, our goal was to create hybrid models that further enhance classification performance by combining the advantages of several different techniques. BiLSTM-CNN, CNN-BiLSTM, SVM-Logistic Regression, BiLSTM-CNN, and SVM-BiLSTM are some of our hybrid model combinations. Our objective is to develop hybrid models that take use of both individual strengths and shortcomings, thereby improving the efficacy of text categorization in the Hindi language domain, by combining various algorithmic capabilities.

Models	**Hybrid Binary Classification**			
	Training Accuracy	**Testing Accuracy**	**F1 Score**	**Recall**
BiLSTM-CNN	99.93%	74.63%	77.12%	77.95%
CNN-BiLSTM	99.96%	55.16%	70.31%	96.77%
Logistic Regression-SVM	96.82%	85.22%	85.95%	85.71%
SVM-Logistic Regression	96.85%	85.08%	85.83%	85.71%
SVM-BiLSTM	55.19%	51.77%	68.22%	99.99%

Table 4.3: COMPARISON RESULTS OF HYBRID MODELS

In our study, we conducted a comprehensive comparison of several machine learning (ML) and natural language processing (NLP) models for the binary categorization of Hindi news items. Table 2 summarizes the performance metrics of these models, including training accuracy, testing accuracy, F1 score, and recall. Based on our analysis, we identified the best-performing models in each category, which served as the foundation for developing hybrid models aimed at further enhancing classification performance.

The BiLSTM-CNN hybrid model exhibited a training accuracy of 99.93% and a testing accuracy of 74.63%, with an F1 score of 77.12% and a recall of 77.95%. This hybrid model leverages the strengths of both BiLSTM and CNN architectures, combining their capabilities in sequence modeling and feature extraction, respectively. The CNN-BiLSTM model, on the other hand, achieved a training accuracy of 99.96% but demonstrated lower testing accuracy at 55.16%. Despite this, it exhibited a high recall of 96.77%, indicating its effectiveness in correctly identifying positive instances.

Moreover, we explored hybrid combinations involving traditional machine learning algorithms such as Logistic Regression and Support Vector Machines (SVM). The Logistic Regression-SVM hybrid model achieved a training accuracy of 96.82% and a testing accuracy of 85.22%, with an impressive F1 score and recall of 85.95% and 85.71%, respectively. Similarly, the SVM-Logistic Regression hybrid model demonstrated comparable performance with a training accuracy of 96.85% and a testing accuracy of 85.08%.

However, our analysis also revealed challenges in hybridization, as exemplified by the SVM-BiLSTM model. Despite incorporating both SVM and BiLSTM components, this hybrid model exhibited lower performance, with a training accuracy of 55.19% and a testing accuracy of 51.77%. While its F1 score was relatively high at 68.22%, its recall was notably lower at 99.99%, indicating a high rate of false negatives.

In summary, our study highlights the effectiveness of hybrid models in improving the efficacy of text categorization in the Hindi language domain. By leveraging the strengths of diverse algorithmic capabilities, these hybrid models offer promising avenues for achieving more accurate and robust classification results, thereby addressing the limitations of individual models and enhancing the automation of news categorization processes in Hindi language publications.

One significant challenge lies in effectively integrating different algorithmic components within hybrid models. As evident from our analysis, not all combinations yield improved performance, and in some cases, hybridization may even lead to diminished results. Therefore, identifying compatible architectures and methodologies for hybridization is crucial to ensure synergistic effects and avoid conflicts between individual components.

Furthermore, the scarcity of labeled data in the Hindi language poses a significant challenge to model development and evaluation. Limited datasets hinder the training of accurate and robust models, particularly for complex tasks such as news categorization. Addressing this challenge requires concerted efforts to curate and annotate large-scale datasets specific to the Hindi language, thereby enabling more comprehensive training and validation of hybrid models.

Additionally, the computational complexity associated with deep learning architectures poses practical challenges, particularly in resource-constrained environments. Training and fine-tuning hybrid models often require substantial computational resources, including high-performance computing infrastructure and specialized hardware accelerators. Mitigating these challenges necessitates the optimization of model architectures, training algorithms, and resource utilization strategies to achieve a balance between performance and computational efficiency.

Despite these challenges, hybrid models offer several promising opportunities for advancing text categorization in the Hindi language domain. By combining complementary strengths and mitigating individual shortcomings, hybrid models can enhance the accuracy, robustness, and scalability of text classification systems. Moreover, the development of hybrid models fosters interdisciplinary collaboration and innovation, drawing upon insights from diverse fields such as machine learning, natural language processing, and linguistics.

In conclusion, our study underscores the significance of hybrid models in advancing text categorization in the Hindi language domain. Through careful analysis and experimentation, we have identified promising hybrid combinations and laid the groundwork for future research and development in this area. By addressing key challenges and leveraging emerging opportunities, hybrid models have the potential to revolutionize the automation of news categorization processes and facilitate access to timely and relevant information for Hindi-speaking audiences worldwide.

4.4. Algorithms:

4.4.1. Machine and Deep Learning Models

i) BERT:

1. Import necessary libraries:

    - **pandas** for data manipulation.
    - **tensorflow** for GPU device availability check.
    - **torch** for PyTorch framework.
    - **train_test_split** from **sklearn.model_selection** for splitting data.
    - **ElectraTokenizer** from **transformers** for tokenization.
    - **pad_sequences** from **tensorflow.keras.preprocessing.sequence** for padding sequences.
    - **TensorDataset**, **DataLoader**, **RandomSampler**, **SequentialSampler** from **torch.utils.data** for creating data loaders.
    - **BertForSequenceClassification**, **AdamW**, **BertConfig** from **transformers** for BERT model and optimizer.
    - **get_linear_schedule_with_warmup** from **transformers** for creating a scheduler.
    - **numpy** for numerical operations.
    - **time** and **datetime** for timing operations.
    - **plotly.express** for visualization.
    - Various metrics from **sklearn.metrics** for evaluation.

2. Read data from CSV files (**india_news.csv** and **international_news.csv**) using **pd.read_csv**.
3. Make the length of both data frames equal by trimming the longer one.
4. Add a label column to each data frame (**india_news** and **international_news**) with values 1 and 0 respectively.
5. Concatenate the two data frames (**india_news** and **international_news**) into a single data frame **news**.
6. Check for GPU availability.
7. Split the data into training and testing sets using **train_test_split**.
8. Tokenize the sentences using ElectraTokenizer and encode them.
9. Pad the sequences to a maximum length.
10. Create attention masks.
11. Split the data into training and validation sets.
12. Convert inputs, masks, and labels to PyTorch tensors.
13. Define batch size and create data loaders for training and validation sets.
14. Load the pre-trained BERT model for sequence classification.
15. Set up the optimizer and scheduler.
16. Define functions for calculating accuracy, formatting time, and clipping gradients.
17. Train the model for the specified number of epochs.
18. Evaluate the model on the validation set after each epoch.
19. Store the training loss values for plotting.

20. After training, plot the training loss.
21. Tokenize the test sentences, pad sequences, create attention masks, and convert to tensors.
22. Create a data loader for the test set.
23. Put the model in evaluation mode and make predictions on the test set.
24. Calculate accuracy and various evaluation metrics (F1 score, precision, recall).
25. Print the classification report.

ii) LSTM:

1. Import necessary libraries:
    - **pandas** for data manipulation.
    - **numpy** for numerical operations.
    - **matplotlib.pyplot** for plotting.
    - **seaborn** for visualization styling.
    - **nltk** for natural language processing tasks.
    - **re** for regular expressions.
    - **Tokennizer** from **keras.preprocessing.text** for tokenization.
    - **pad_sequences** from **keras.preprocessing.sequence** for sequence padding.
    - Various layers and utilities from **keras** for building and compiling the LSTM model.
2. Read the India and international news datasets from CSV files.

3. Ensure that both datasets have the same length by truncating the longer one.
4. Combine the datasets and assign labels (**0** for international news and **1** for India news).
5. Clean the text data by removing stopwords and stemming the words using the Porter Stemmer.
6. Tokenize the cleaned text data and convert it into one-hot representations.
7. Pad the sequences to ensure uniform length.
8. Split the data into training and testing sets.
9. Build the LSTM model:
    - Define the embedding dimension.
    - Create a Sequential model.
    - Add an Embedding layer for word embeddings.
    - Add a Bidirectional LSTM layer for sequence processing.
    - Add Dropout layers to prevent overfitting.
    - Add Dense layers for classification.
    - Compile the model with binary cross-entropy loss and Adam optimizer.
10. Train the model on the training data.
11. Predict labels for the test data and evaluate the model:
    - Make predictions on the test data.
    - Convert predicted probabilities to binary predictions.
    - Calculate accuracy, precision, recall, and F1-score using sklearn metrics.

12. Print the evaluation metrics and classification report.

13. Test the model with a sample string and predict its label.

iii) CNN

1. Import necessary libraries:
    - **punctuation** from **string** for string operations.
    - **listdir** from **os** for listing directory contents.
    - Various modules from **numpy** for numerical operations.
    - **pd** for data manipulation using Pandas.
    - **keras** for building neural network models.
    - **Tokenizer** from **keras.preprocessing.text** for tokenization.
    - **pad_sequences** from **keras.preprocessing.sequence** for sequence padding.
    - **Sequential**, **Dense**, **Dropout**, **Flatten**, **Embedding**, **Conv1D**, and **MaxPooling1D** from **keras.layers** for building layers of the neural network.
    - **confusion_matrix**, **accuracy_score**, **precision_score**, **recall_score**, and **f1_score** from **sklearn.metrics** for evaluation metrics.
    - **matplotlib.pyplot** for plotting.
2. Read data from CSV files (**india_news.csv** and **international_news.csv**) using **pd.read_csv**.
3. Preprocess the data:

- Combine the two data frames (**india_news** and **international_news**) into a single data frame **news**.
- Encode labels using one-hot encoding.
- Create train and test data sets using a given split ratio.
- Tokenize documents and clean them by removing punctuation and filtering based on word frequency.

4. Load pre-trained word embeddings and create an embedding matrix.
5. Define and compile the Convolutional Neural Network (CNN) model using Keras.
6. Train the CNN model on the training data.
7. Predict the labels for test data and evaluate the model:
    - Make predictions on the test data.
    - Convert predicted probabilities to class labels.
    - Calculate precision, recall, and F1-score using sklearn metrics.
    - Print precision, recall, and F1-score.

iv) Logistic Regression

1. **Import Libraries**:
    - Import necessary libraries such as NumPy, Matplotlib, pandas, NLTK, and os.
2. **Load Data**:
    - Read the Indian and international news data from CSV files.

- Assign labels to the news articles.
- Concatenate the international and Indian news dataframes.

3. **Data Preprocessing**:
    - Split the data into features (X) and labels (y).
    - Define a list of stop words to be removed from the text data.
    - Define a custom tokenizer function.
    - Initialize CountVectorizer with specified parameters and fit it on the training data.
    - Transform the training and testing sets into bag-of-words representation.
4. **Model Training**:
    - Initialize Logistic Regression model with specified hyperparameters.
    - Train the Logistic Regression model on the vectorized training data.
5. **Model Evaluation**:
    - Make predictions on both training and testing data using the trained model.
    - Calculate accuracy scores for both training and testing sets.
    - Calculate F1 score for the testing set.
6. **Print Results**:
    - Print the train accuracy and test accuracy.
    - Print the F1 score for the testing set.

v) Naïve Bayes

1. Import necessary libraries:
    - **numpy** for numerical operations.
    - **pandas** for data manipulation.
    - **matplotlib.pyplot** for plotting.
    - **nltk** for natural language processing tasks.
    - **os** for operating system related tasks.
    - **pickle** for serializing and deserializing Python objects.
    - **WordCloud** from **wordcloud** for generating word clouds.
    - **warnings** to handle warnings gracefully.
2. Read data from CSV files (**india_news.csv** and **international_news.csv**) using **pd.read_csv**.
3. Concatenate the two data frames (**india_news** and **international_news**) into a single data frame **news**.
4. Set up features **X** as news articles and labels **y** as binary labels.
5. Define stop words to be removed from text data.
6. Generate a word cloud for news articles labeled as international using **WordCloud**.
7. Split the data into training and testing sets (**xtrain**, **xtest**, **ytrain**, **ytest**) using **train_test_split**.
8. Define a custom tokenizer function **my_tokenizer** to split text into words.

9. Initialize a TF-IDF vectorizer (**TfidfVectorizer**) with custom tokenizer and stop words, and fit it to training data (**xtrain_tf**).
10. Transform training and testing data into TF-IDF feature matrices (**xtrain_tf**, **xtest_tf**).
11. Initialize a Multinomial Naive Bayes classifier (**MultinomialNB**) with specified alpha value and fit it to the training data.
12. Make predictions on the testing data and print classification report.
13. Evaluate accuracy and F1-score on both training and testing data.

vi) BiLSTM

1. **Import Libraries**:
    - Import necessary libraries for data manipulation, visualization, NLP, and deep learning.
2. **Load Data**:
    - Read the data from CSV files for international and Indian news.
3. **Preprocess Data**:
    - Assign labels to news articles.
    - Concatenate international and Indian news.
    - Split the data into features (input) and labels (output).
    - Reset index for proper concatenation.
4. **NLP Preprocessing**:
    - Tokenize news articles.

- Remove stopwords and perform stemming.
- Create a corpus of preprocessed text.
- Perform one-hot encoding on the corpus.
- Pad sequences to ensure uniform length for input to the model.

5. **Model Creation**:
    - Build a sequential model.
    - Add an embedding layer.
    - Add a Bidirectional LSTM layer.
    - Add a dense layer with sigmoid activation for binary classification.
    - Compile the model with appropriate loss function and optimizer.
6. **Model Training**:
    - Split the data into training and testing sets.
    - Train the model on the training data for a specified number of epochs.
    - Validate the model on the testing data.
7. **Model Evaluation**:
    - Make predictions on the testing set.
    - Convert predicted probabilities to binary predictions based on a threshold of 0.5.
    - Calculate various evaluation metrics such as accuracy, precision, recall, and F1-score on the testing set.
    - Calculate accuracy on the training set for comparison.

8. **Print Results**:

    - Print the accuracy, precision, recall, and F1-score on the testing set.
    - Print the accuracy on the training set.
    - Print a classification report containing precision, recall, F1-score, and support.

vii) SVM

1. **Import Libraries**:

    - Import necessary libraries such as SVM from scikit-learn, NumPy, pandas, and model selection.

2. **Load Data**:

    - Read the Indian and international news data from CSV files.
    - Assign labels to the news articles.
    - Concatenate the international and Indian news data frames.

3. **Data Preprocessing**:

    - Split the data into features (X) and labels (y).
    - Define a list of stop words to be removed from the text data.
    - Define a custom tokenizer function.
    - Initialize TF-IDF vectorizer with specified parameters.

4. **SVM Model Training**:

    - Split the dataset into training and testing sets using train_test_split.
    - Perform TF-IDF vectorization on the training and testing sets.

- Initialize an SVM classifier with the RBF kernel and specified hyperparameters.
- Train the SVM classifier on the TF-IDF transformed training data.

5. **Model Evaluation**:
   - Make predictions on the test data using the trained SVM classifier.
   - Calculate precision and recall scores for the SVM classifier with the RBF kernel.
   - Calculate train score, test accuracy, and F1 score for the SVM classifier with the RBF kernel using cross-validation and appropriate scikit-learn functions.
   - Print classification report containing precision, recall, F1-score, and support.
6. **Print Results**:
   - Print precision and recall scores for the SVM classifier with the RBF kernel.
   - Print train score, test accuracy, and F1 score for the SVM classifier with the RBF kernel.
   - Print classification report.

viii) RNN

1. Import necessary libraries:
   - **pandas** for data manipulation.
   - **numpy** for numerical operations.

- **matplotlib.pyplot** for plotting.
- **seaborn** for visualization.
- **train_test_split** from **sklearn.model_selection** for splitting data.
- **LabelEncoder** from **sklearn.preprocessing** for label encoding.
- Various modules from **keras** for building neural network models.
- **RMSprop** optimizer from **tensorflow.keras.optimizers**.
- **Tokenizer** from **keras.preprocessing.text** for tokenization.
- **sequence** from **keras.preprocessing** for sequence padding.
- **to_categorical** from **tensorflow.keras.utils** for one-hot encoding.

2. Read data from CSV files (**india_news.csv** and **international_news.csv**) using **pd.read_csv**.
3. Concatenate the two data frames (**india_news** and **international_news**) into a single data frame **news**.
4. Preprocess the data:
    - Encode labels using **LabelEncoder**.
    - Split the data into training and testing sets using **train_test_split**.
5. Tokenize the text data using **Tokenizer** and convert text sequences to sequences of integers.
6. Pad sequences to ensure uniform length.
7. Define a function **RNNmodel** to build the Recurrent Neural Network (RNN) model.

8. Compile the model with binary cross-entropy loss, RMSprop optimizer, and accuracy and precision metrics.
9. Train the model on the training data.
10. Evaluate the model on the testing data and print the accuracy.
11. Calculate precision, recall, and F1-score using predictions from the model.

4.4.1. Hybrid Models

i) BiLSTM-CNN:

1. **Import necessary libraries:**
    - TensorFlow and Keras for deep learning functionalities.
    - sklearn for model stacking.
    - numpy for numerical operations.
    - Other necessary libraries for data manipulation and preprocessing.
2. **Preprocess the data:**
    - Tokenize the input text data and convert it into sequences of integers.
    - Pad sequences to ensure uniform length.
    - Split the data into training and testing sets.
3. **Define individual models:**
    - Build a BiLSTM model using sequential layers, incorporating Bidirectional LSTM layers followed by dropout layers.
    - Construct a CNN model with convolutional layers, pooling layers, and dropout layers.
4. **Train individual models:**
    - Train the BiLSTM model on the training data.
    - Train the CNN model on the training data.
5. **Predictions from individual models:**

- Obtain predictions from the BiLSTM model on the testing data.
- Obtain predictions from the CNN model on the testing data.

6. **Stacking ensemble:**
    - Concatenate the predictions from both the BiLSTM and CNN models to form a new feature matrix.
    - Build a meta-learner model (e.g., Logistic Regression or SVM) using the concatenated predictions as input features and the true labels as the target variable.
    - Train the meta-learner model on the concatenated predictions from the training data.
7. **Make final predictions:**
    - Combine predictions from the individual models using the meta-learner model to make final predictions on the testing data.
8. **Evaluate performance:**
    - Calculate evaluation metrics such as accuracy, F1-score, precision, and recall for the stacked ensemble model.
9. **Optimization and tuning:**
    - Perform hyperparameter tuning for individual models and the meta-learner model to optimize performance.
    - Cross-validate the stacked ensemble model to ensure generalizability.
10. **Model deployment:**
    - Once satisfied with performance, deploy the stacked ensemble model for real-world applications, such as automated classification of Hindi news articles.

ii) CNN-BiLSTM:

1. **Import necessary libraries:**

- TensorFlow and Keras for deep learning functionalities.
- sklearn for model stacking.
- numpy for numerical operations.
- Other necessary libraries for data manipulation and preprocessing.

2. **Preprocess the data:**
   - Tokenize the input text data and convert it into sequences of integers.
   - Pad sequences to ensure uniform length.
   - Split the data into training and testing sets.
3. **Define individual models:**
   - Build a CNN model with convolutional layers, pooling layers, and dropout layers.
   - Construct a BiLSTM model using sequential layers, incorporating Bidirectional LSTM layers followed by dropout layers.
4. **Train individual models:**
   - Train the CNN model on the training data.
   - Train the BiLSTM model on the training data.
5. **Predictions from individual models:**
   - Obtain predictions from the CNN model on the testing data.
   - Obtain predictions from the BiLSTM model on the testing data.
6. **Stacking ensemble:**
   - Concatenate the predictions from both the CNN and BiLSTM models to form a new feature matrix.
   - Build a meta-learner model (e.g., Logistic Regression or SVM) using the concatenated predictions as input features and the true labels as the target variable.
   - Train the meta-learner model on the concatenated predictions from the training data.

7. **Make final predictions:**
    - Combine predictions from the individual models using the meta-learner model to make final predictions on the testing data.
8. **Evaluate performance:**
    - Calculate evaluation metrics such as accuracy, F1-score, precision, and recall for the stacked ensemble model.
9. **Optimization and tuning:**
    - Perform hyperparameter tuning for individual models and the meta-learner model to optimize performance.
    - Cross-validate the stacked ensemble model to ensure generalizability.
10. **Model deployment:**
    - Once satisfied with performance, deploy the stacked ensemble model for real-world applications, such as automated classification of Hindi news articles.

iii) Logistic Regression-SVM:

1. **Import necessary libraries:**
    - scikit-learn for machine learning functionalities.
    - numpy for numerical operations.
    - Other necessary libraries for data manipulation and preprocessing.
2. **Preprocess the data:**
    - Perform necessary data preprocessing steps, such as encoding categorical features and scaling numerical features.
    - Split the data into training and testing sets.
3. **Define individual models:**
    - Build a Logistic Regression model.
    - Construct a Support Vector Machine (SVM) model.

4. **Train individual models:**
    - Train the Logistic Regression model on the training data.
    - Train the SVM model on the training data.
5. **Predictions from individual models:**
    - Obtain predictions from the Logistic Regression model on the testing data.
    - Obtain predictions from the SVM model on the testing data.
6. **Stacking ensemble:**
    - Concatenate the predictions from both the Logistic Regression and SVM models to form a new feature matrix.
    - Build a meta-learner model (e.g., Logistic Regression or SVM) using the concatenated predictions as input features and the true labels as the target variable.
    - Train the meta-learner model on the concatenated predictions from the training data.
7. **Make final predictions:**
    - Combine predictions from the individual models using the meta-learner model to make final predictions on the testing data.
8. **Evaluate performance:**
    - Calculate evaluation metrics such as accuracy, F1-score, precision, and recall for the stacked ensemble model.
9. **Optimization and tuning:**
    - Perform hyperparameter tuning for individual models and the meta-learner model to optimize performance.
    - Cross-validate the stacked ensemble model to ensure generalizability.
10. **Model deployment:**

- Once satisfied with performance, deploy the stacked ensemble model for real-world applications, such as automated classification of Hindi news articles.

iv) SVM-Logistic Regression:

1. **Import necessary libraries:**
    - scikit-learn for machine learning functionalities.
    - numpy for numerical operations.
    - Other necessary libraries for data manipulation and preprocessing.
2. **Preprocess the data:**
    - Perform necessary data preprocessing steps, such as encoding categorical features and scaling numerical features.
    - Split the data into training and testing sets.
3. **Define individual models:**
    - Build a Support Vector Machine (SVM) model.
    - Construct a Logistic Regression model.
4. **Train individual models:**
    - Train the SVM model on the training data.
    - Train the Logistic Regression model on the training data.
5. **Predictions from individual models:**
    - Obtain predictions from the SVM model on the testing data.
    - Obtain predictions from the Logistic Regression model on the testing data.
6. **Stacking ensemble:**
    - Concatenate the predictions from both the SVM and Logistic Regression models to form a new feature matrix.

- Build a meta-learner model (e.g., Logistic Regression or SVM) using the concatenated predictions as input features and the true labels as the target variable.
- Train the meta-learner model on the concatenated predictions from the training data.

7. **Make final predictions:**
    - Combine predictions from the individual models using the meta-learner model to make final predictions on the testing data.
8. **Evaluate performance:**
    - Calculate evaluation metrics such as accuracy, F1-score, precision, and recall for the stacked ensemble model.
9. **Optimization and tuning:**
    - Perform hyperparameter tuning for individual models and the meta-learner model to optimize performance.
    - Cross-validate the stacked ensemble model to ensure generalizability.
10. **Model deployment:**
    - Once satisfied with performance, deploy the stacked ensemble model for real-world applications, such as automated classification of Hindi news articles.

v) SVM-BiLSTM:

1. **Import necessary libraries:**
    - TensorFlow and Keras for deep learning functionalities.
    - scikit-learn for machine learning functionalities.
    - numpy for numerical operations.
    - Other necessary libraries for data manipulation and preprocessing.
2. **Preprocess the data:**

- Tokenize the input text data and convert it into sequences of integers.
- Pad sequences to ensure uniform length.
- Perform necessary data preprocessing steps, such as encoding categorical features and scaling numerical features.
- Split the data into training and testing sets.

3. **Define individual models:**
    - Build a Support Vector Machine (SVM) model.
    - Construct a BiLSTM model using sequential layers, incorporating Bidirectional LSTM layers followed by dropout layers.
4. **Train individual models:**
    - Train the SVM model on the training data.
    - Train the BiLSTM model on the training data.
5. **Predictions from individual models:**
    - Obtain predictions from the SVM model on the testing data.
    - Obtain predictions from the BiLSTM model on the testing data.
6. **Stacking ensemble:**
    - Concatenate the predictions from both the SVM and BiLSTM models to form a new feature matrix.
    - Build a meta-learner model (e.g., Logistic Regression or SVM) using the concatenated predictions as input features and the true labels as the target variable.
    - Train the meta-learner model on the concatenated predictions from the training data.
7. **Make final predictions:**
    - Combine predictions from the individual models using the meta-learner model to make final predictions on the testing data.
8. **Evaluate performance:**

- Calculate evaluation metrics such as accuracy, F1-score, precision, and recall for the stacked ensemble model.

9. **Optimization and tuning:**
   - Perform hyperparameter tuning for individual models and the meta-learner model to optimize performance.
   - Cross-validate the stacked ensemble model to ensure generalizability.

10. **Model deployment:**
    - Once satisfied with performance, deploy the stacked ensemble model for real-world applications, such as automated classification of Hindi news articles.

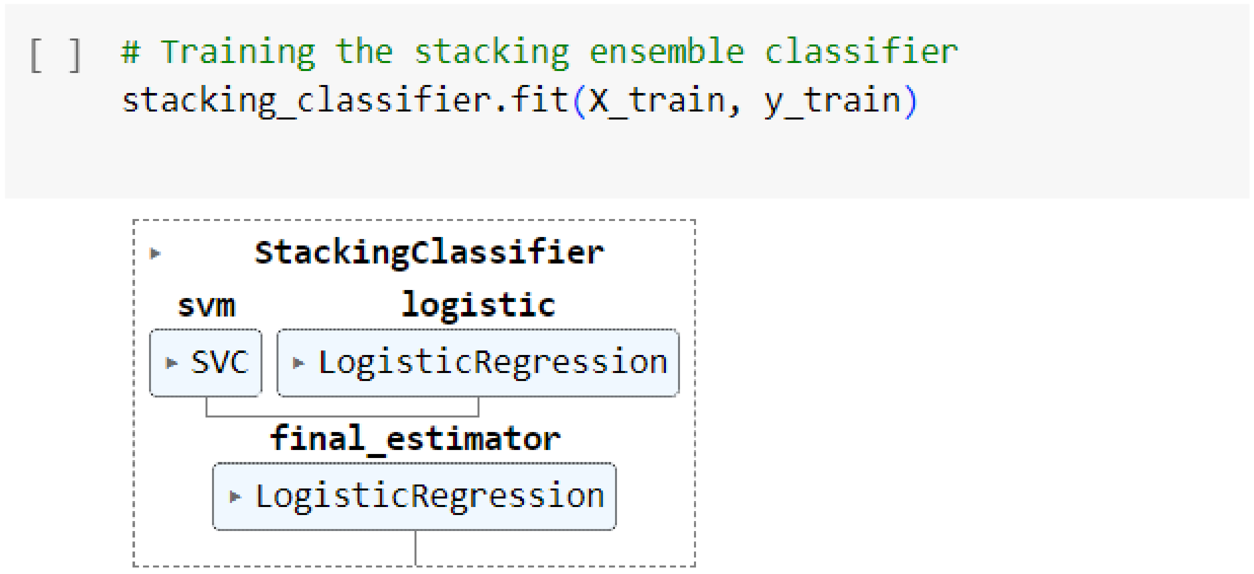
Chapter-5
RESULTS AND DISCUSSION

5.1. Analysis and assessment of performance

This section contains the proper outcomes after training and testing the binary classification system of our hybrid model.

The complexity of the Hindi News Text Binary Classification system is profound, making its evaluation and analysis a challenging task. The content is intricate, requiring a deep understanding to capture the meaning in a sentence. Linking an incident with its associated argument set becomes even more challenging with increased distance. Various methods can be employed to evaluate and analyze the system's performance, each demanding a thorough understanding of the system's intricacies.

The research focused on classification using current techniques, such as RNN, LSTM, BiLSTM, BERT, CNN, NB, LR, and SVM classifiers, and finally training the hybrid model using various features of Table-2 ML and NLP techniques. This research needs a large Hindi dataset. With adequate training data, text classification is more straightforward and requires sufficient hand-labeled data to use supervised techniques[15]. Consequently, the best evaluation metric for classifying or defining the fundamental data categories would be an assessment made by a human assessor.



```
[ ]  # Training the stacking ensemble classifier
     stacking_classifier.fit(X_train, y_train)
```

Figure 5.1: Training of SVM-Logistic Regression Hybrid model

5.2. Experimental Setup and Results Obtained:

Using the hand-labeled approach, labeled data preparation was done for different classification models, and it was discovered that the accuracy of the classifier is better if labeled data is increased to train the model. Instead of classifying the raw data into multiclass news categories, this research defined two primary categories देश(India), वदेश(International) in the first phase and ten different news classes in secondary categorization. A total of 8 different classification models are used for binary classification, and Performance indicators are essential when assessing the effectiveness of machine learning algorithms and making judgments. After the model was successfully prepared and tested on the news dataset, the outputs were obtained and compared regarding F1 score, precision, and recall. These indicators are crucial for assessing algorithm performance, directing the optimization process, reporting outcomes, and discovering limitations. SVM, LR, and RNN models classify and give better results than the other models for our highly challenging news data.

From figure, the binary classification training and testing accuracy of the deep learning models for India and International class can be observed.

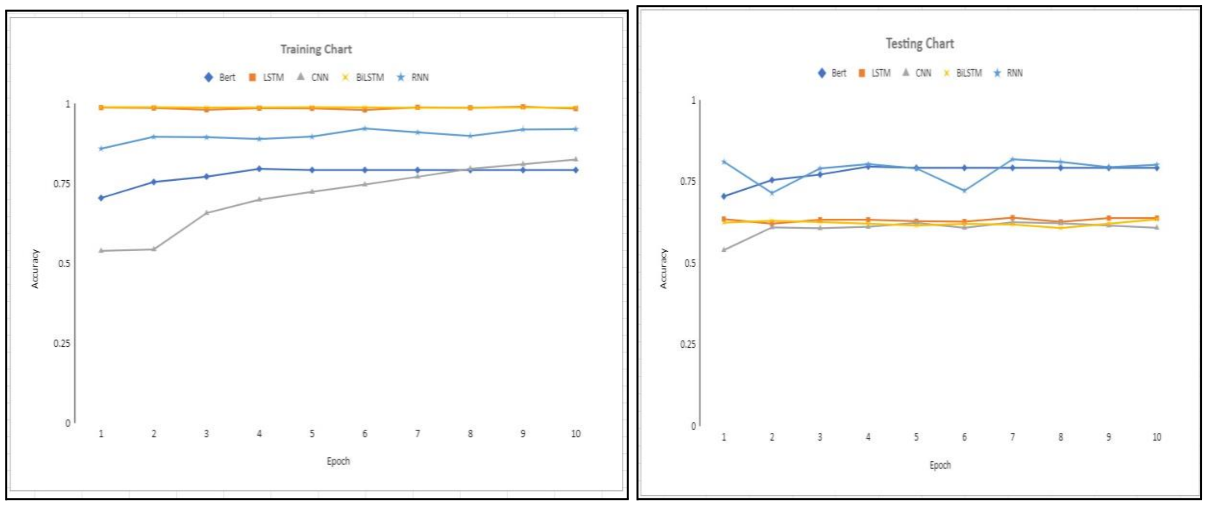


Figure 5.2. Training and testing accuracy graph for machine learning models

The graph shows the testing and training accuracy concerning algorithms for binary classification. Suppose the analysis in the diagram takes place by using the . In that case, the variation between training and validation data which is not the same.

Chapter-6
Conclusion And Scope of Future Work

In conclusion, the SVM, LR, and RNN outperformed all other algorithms regarding model performance measures in the binary text classification tasks examined, and in case of hybrid model the BiLSTM-CNN and LogisticRegression-SVM outperformed all other algorithms. Text classification has significantly impacted fields, including language identification, sentiment, and semantic analysis, news classification, and spam content detection. This work focuses on the classification of Hindi text because few studies have been done. In this research, eight separate classification techniques were applied, and the results were carefully examined and furthermore the five hybrid models were trained. NLP techniques were used on the news articles and pre-processed data to examine the effects of data preparation. Text Data were normalized by Stopwords, tokenizer in the preprocessing stage, and vectorizer for feature extraction.

The data were trained using the SVM, NB, LR, CNN, RNN, LSTM, BiLSTM, and Bert techniques. The performance of the various methods was analysed after the classification. The SVM, LR, and RNN methods were the most effective regarding accuracy, F1 score, precision, and recall. The results indicate that pre-processed datasets and coherence between the number of training sets and categories are important factors affecting how well the classifier works. Binary classification is applied in this work, having two categories only and, as of now, getting good results. The TF-IDF approach utilized in the feature extraction stage stopword and tokenizer used in data preprocessing significantly impact the results. Overall, 89% performance accuracy for SVM and for LogisticRegression-SVM 85.22% accuracy was attained in this study, significantly greater than the results of earlier text classification solutions suggested for the Hindi language. Currently, no available system classifies the Hindi text into predefined categories by considering newswire articles, processing them, and organizing random news articles. We plan to build a classification model using our own Hindi dataset, which uses NLP, Machine, and Deep Learning. This work can also be expanded further into the areas with the expansion of Hindi datasets and the development of a classification system for multiple classes with an innovative algorithm, providing better performance with outstanding results.

REFERENCES

[1] Pallathadka, H., Pallathadka, L. K., & Devi, T. K. 2022. “Importance of Hindi Language and Its Significance in Nation-Building. Integrated Journal for Research in Arts and Humanities”, 2(6), 92–98. https://doi.org/10.55544/ijrah.2.6.12

[2] Sahoo Kumar Sovan, Saha Saumajit, Ekbal Asif, Bhattacharyya Pushpak and Mathew Jimson 2019 “Event-Argument Linking in Hindi for Information Extraction in Disaster Domain” *CICLing 2019*.

[3] Ahmad Zishan, Sahoo Kumar Sovan, Ekbal Asif, Bhattacharyya Pushpak 2018 “A Deep Learning Model for Event Extraction and Classification in Hindi for Disaster Domain” *Proc. of ICON*-2018, Patiala, India. December 2018 c2018 NLPAI, pages 127–136.

[4] Shah, K., Patel, H., Sanghvi, D. et al. A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. Augment Hum Res 5, 12 (2020). https://doi.org/10.1007/s41133-020-00032-0.

[5] Alzoubi, Yehia Ibrahim, Ahmet E. Topcu, and Ahmed Enis Erkaya. 2023. "Machine Learning-Based Text Classification Comparison: Turkish Language Context" *Applied Sciences* 13, no. 16: 9428. https://doi.org/10.3390/app13169428.

[6] Mccallum A, Nigam K 2001 A comparison of event models for naive bayes text classification. Work LearnText Categ752:05

[7] Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative Study of CNN and RNN for Natural Language Processing. *ArXiv*. /abs/1702.01923.

[8] Umer Muhammad, Imtiaz Zainab, Ahmad Muhammad, Nappi Michele, Medaglia Carlo, Choi Gyu Sang Mehmood Arif 2022 “ Impact of convolutional neural network and FastText embedding on text classification” Multimedia Tools and Applications” August 2022, volume 82, 10.1007/s11042-022-13459-x.

[9] Arora, M., Dhingra, B., Gupta, D., Singh, D. (2022). Performance Comparison of Different Machine Learning Algorithms on Hindi News Classification. In: Khanna, A., Gupta, D., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1388. Springer, Singapore. https://doi.org/10.1007/978-981-16-2597-8_27

[10] Kulkarni, A., Mandhane, M., Likhitkar, M., Kshirsagar, G., Jagdale, J., Joshi, R. (2022). Experimental Evaluation of Deep Learning Models for Marathi Text Classification. In: Gunjan, V.K., Zurada, J.M. (eds) Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications. Lecture Notes in Networks and Systems, vol 237. Springer, Singapore. https://doi.org/10.1007/978-981-16-6407-6_53

[11] Khuntia, M., & Gupta, D. (2022). Indian News Headlines Classification using Word Embedding Techniques and LSTM Model. *Procedia Computer Science*, *218*, 899-907. https://doi.org/10.1016/j.procs.2023.01.070

[12] Soni, S., Chouhan, S.S. & Rathore, S.S. TextConvoNet: a convolutional neural network based architecture for text classification. Appl Intell 53, 14249–14268 (2023). https://doi.org/10.1007/s10489-022-04221-9

[13] Al-Qerem, A., Raja, M., Taqatqa, S., Sara, M.R.A. (2024). Utilizing Deep Learning Models (RNN, LSTM, CNN-LSTM, and Bi-LSTM) for Arabic Text Classification. In: Musleh Al-Sartawi, A.M.A., Al-Qudah, A.A., Shihadeh, F. (eds) Artificial Intelligence-Augmented Digital Twins. Studies in Systems, Decision and Control, vol 503. Springer, Cham. https://doi.org/10.1007/978-3-031-43490-7_22

[14] Wahdan, A., Hantoobi, S., Salloum, S. A., & Shaalan, K. (2020). A systematic review of text classification research based on deep learning models in Arabic language. *Int. J. Electr. Comput. Eng*, *10*(6), 6629-6643.

[15] S. Usmani and J. A. Shamsi, (2020) "News Headlines Categorization Scheme for Unlabelled Data," International Conference on Emerging Trends in Smart Technologies (ICETST), pp. 1-6.

[16] Mahato S., Thomas A. 2017 “Lexico-semantic analysis of essays in Hindi language” CEUR Workshop Proceedings 2017, 1819

[17] Thomas A., Kowar M. K., Sharma S. and Sharma H. R., "Exploring Text Semantics to Extract Key-Fragments for Model Answers," 2010 *International Conference on Advances in Recent Technologies in Communication and Computing*, Kottayam, India, 2010, pp. 255-257, doi: 10.1109/ARTCom.2010.110.