Heaven’s Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

Leveraging Traditional Machine Learning Classifiers and Transformer Model for Bangla Fake News Detection

Author

Mst. Sadia Khatun

Roll No. 1803051

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Supervised by

Dr. Md. Shahid Uz Zaman

Professor

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

ACKNOWLEDGEMENT

First of all, I would like to thank almighty Allah, for his grace and blessings as well as for providing me with the diligence and enthusiasm along the way to accomplishing my thesis work.

I also want to express my sincere gratitude, admiration and heartfelt appreciation to my supervisor **Dr. Md. Shahid Uz Zaman**, Professor, Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi. Throughout the year, he has not only provided me with the technical instructions and documentation to complete the work, but he has also continuously encouraged me, offered me advise, assisted me, and cooperated sympathetically whenever he deemed necessary. His constant support was the most successful tool that helped me to achieve my result. Whenever I was stuck in any complex problems or situation he was there for me at any time of the day. Without his sincere care, this work not has been materialized in the final form that it is now at the present.

I am also grateful to respected **Prof. Dr. Ali Hossain**, Head of the Department of Computer Science & Engineering and all the respective teachers of Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi for their valuable suggestions and inspirations from time to time.

Finally, I would like to convey my thanks to my parents, friends, and well-wishers for their true motivations and many helpful aids throughout this work.

May 23, 2024 — Mst. Sadia Khatun

RUET, Rajshahi

Heaven's Light is Our Guide

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Rajshahi University of Engineering & Technology, Bangladesh

CERTIFICATE

*This is to certify that this thesis report entitled **"Leveraging Traditional Machine Learning Classifiers and Transformer Model for Bangla Fake News Detection"** submitted by **Mst. Sadia Khatun, Roll:1803051** in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Department of Computer Science & Engineering of Rajshahi University of Engineering & Technology, Bangladesh is a record of the candidate own work carried out by him under my supervision. This thesis has not been submitted for the award of any other degree.*

Supervisor

____

Dr. Md. Shahid Uz Zaman
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

External Examiner

____

Prof. Dr. Md. Al Mamun
Professor
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204

ABSTRACT

Many individuals in today's digital age rely on the internet for nearly every purpose, including reading and viewing news on social networking sites. However, on occasion, these platforms distribute false information, which can have serious negative effects on both society and consumers. Because of the layout of online platforms, fake news is easily circulated. Fake news can be combated with tools, but most of them are only available to English speakers, which leaves millions of Bangla speakers out in the cold. The BanFakeNews dataset was first used in this study to create an efficient model for spotting fake news in Bangla. This dataset has 1,300 bogus and 48,000 legitimate news pieces. We used undersampling to remedy the imbalance. TF-IDF and BoW are two feature extraction techniques that are used to test several machine learning classifiers and Transformer model, including Bert. With an accuracy of 95.19%, our best-performing model was logistic regression with TF-IDF features. With 94.04% accuracy, Bangla-Bert-Base performs nearly identically. We attempted to obtain additional data by visiting Bangla fact-checking websites that flag bogus news and collect actual news headlines from YouTube, which offers a wealth of instantly accessible content because these articles were brief. Next, we used this scraped dataset which contains 1913 fake and 1995 Authentic news titles to evaluate several model detection capabilities for false information.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Introduction

The primary issue that the thesis addresses is outlined at the beginning of this chapter. After that, it discusses the significance of this particular subject and the factors that led to this line of inquiry in the research. It then provides a summary of the precise aims and objectives of the study. After that, it gives readers a summary of the thesis's structure and organization and a road map for the following chapters. Finally, the chapter ends with a conclusion

1.2 Problem Statement

Information that is erroneous or deceptive yet is portrayed as actual news is called fake news. False narratives, manipulated pictures and videos, misleading headlines and skewed reporting are only a few instances of fake news. Since social media and various other online platforms are so widely used in today's digital age, fake news can spread quickly and have grave implications. False information can seriously hurt individuals as well as communities when it is widely circulated. Consequently, there has been a growing interest in developing methods to identify and prevent the spread of fake news on social media platforms.

With roughly 240 million native speakers, Bengali is the sixth most spoken native language and the seventh most spoken language overall in the world[1], so the creation of an effective way to find and remove erroneous data is therefore urgently required. Researchers have been looking at the automatic detection of bogus news for several years. However, there has been limited

research attention given to identifying fake news in low-resource languages like Bangla.
Our research focuses on determining the authenticity of Bangla news articles using a combination of classical supervised learning[2] and transfer learning [3] strategies.

1.3 Motivation

False information is rapidly proliferating on social media and posing major issues worldwide, especially in Bangladesh. False information regarding health remedies such as consuming specific leaves to keep safe was spread by people during the COVID-19 outbreak. False allegations on social media sites like Facebook have also been harmful. Recently, on the occasion of Eid, someone claimed on Facebook that Bkash was giving out a 7,999 tk special Eid bonus to celebrate Eid-ul-Fitr. But as it turned out, this post was fraudulent. A similar fraudulent advertisement was made in another thread, claiming that Aarong was awarding cash awards in addition to a ”Ramadan Gift.” Unfortunately, a lot of individuals spread and believed these posts. Such posts should be taken into consideration because they could mislead people into disclosing personal information by clicking on dangerous links. We are proposing a model that will more precisely identify fake news by utilizing Transformers and various machine learning classifiers in order to stop the spread of false information.

1.4 Research Challenges

When dealing with Bangla text data, several challenges arise. Some of these challenges include:

- Limited availability of labeled and balanced datasets.
- Absence of comprehensive natural language processing (NLP). [4] tools and resources designed for handling Bangla text.
- To enhance the accuracy of the proposed model, extensive data preprocessing is necessary.
- Developing a model that performs better in predicting the target attribute.

1.5 Objectives of the Research

To address these challenges, our research aims to achieve the following objectives:

- Learning and studying about natural language processing (NLP)[4].
- Implement undersampling techniques to balance the dataset.
- Present a comparative analysis based on numerous feature extraction approaches with traditional machine learning models.
- Incorporating different types of pre-trained models to enhance the classification system.
- Designing and implementing a model to detect fake news in Bangla.
- Assess the capability of the models in detecting fake news using the scraped dataset from additional sources of data, such as Bangla fact-checking websites and YouTube.
- Conducting a comparative analysis to evaluate the performance of various classifiers.

1.6 Thesis Organization

The report is divided into 6 chapters including this chapter: ***Introduction*** where all the essential topics are discussed which are needed for comprehending the research work. The rest of the works are organized in the following manner:

Chapter 2

Topic - Background

This chapter offers an essential overview of the theory and mathematics behind machine learning algorithms, deep learning methods, transfer learning techniques and feature extraction methods.

Chapter 3

Topic - Literature Review

This chapter summarizes some noteworthy contributions in the field of Bangla fake news detection.

Chapter 4

Topic - Proposed Methodology & Implementation

This chapter covers the dataset and proposed methodology, along with a thorough exploration of data preprocessing, data scraping techniques, the proposed architecture, and the hyperparameters utilized for model training.

Chapter 5

Topic - Result & Performance Analysis

In this chapter, evaluate the proposed architecture, discuss the results, and their performance, and compare it to similar approaches. The measurements used to judge our model's effectiveness are also described here.

Chapter 6

Topic - Conclusion

In this concluding chapter, I summarize the findings of my research. I have made an effort to acknowledge the constraints I encountered in my research and suggest potential directions for future investigations.

1.7 Conclusion

The introductory chapter provided a glimpse of what will be explored in this research. We shall talk about the details in the following chapters.

Chapter 2

Background

2.1 Introduction

The background chapter serves as an introduction to the research problem. This chapter offers a brief overview of the theoretical foundations relevant to our work. It covers a variety of machine learning algorithms, such as Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, and Gradient Boosting. Additionally, it explores deep learning architectures, including RNN, and discusses the concept of transfer learning and the BERT model. Furthermore, the chapter elaborates on the fundamental operations of natural language processing (NLP), covering aspects such as text preprocessing and feature extraction from text data.

2.2 Machine learning

Just like with other types of unstructured data, like images, sound, and structured data, machine learning techniques are also used for textual data. Various NLP jobs mainly rely on supervised machine learning techniques like classification. The most common steps in any machine learning strategy for NLP, are capturing features from data, a learner model that can learn from these features, and then evaluating the model and modification.

2.2.1 Logistic Regession

The most used supervised machine learning model is logistic regression. For categorization jobs, it is employed. Depending on a threshold value, it can categorize labels into two groups, such as 0 or 1. This classifier can also be used to classify text.

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{2.1}$$

Because it is simple to comprehend and apply, logistic regression is a good option for data science and machine learning. The impact of each feature on the probability of the positive class is represented by the model's coefficients, which make it simple to analyze and comprehend the correlations between the variables. Large datasets can be handled with comparatively little memory and processing overhead thanks to the computational efficiency of logistic regression. In cases where classes are linearly separable, results using Logistic Regression can be quite accurate. A straight line can be transformed into a recognizable S-shaped curve using the sigmoid function.

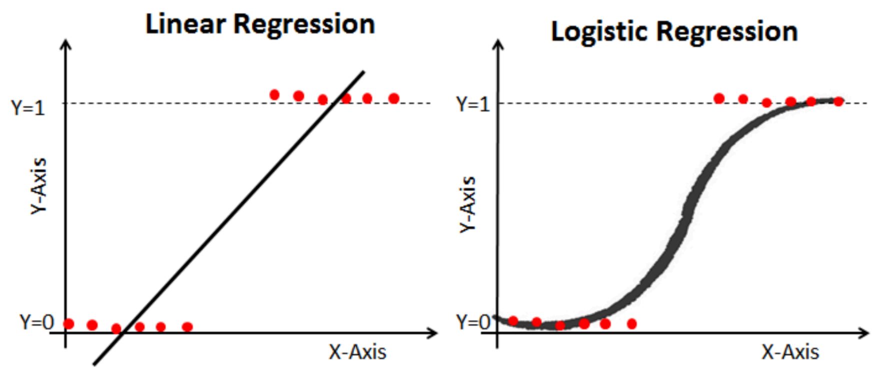

Figure 2.1: Linear Regression vs Logistic Regression [5]

2.2.2 Support vector Machine

The SVM algorithm divides n-dimensional space into several classes by finding an ideal line, or decision boundary, called a hyperplane. In the future, this will make it easier to accurately classify newly collected data points into relevant classes. Support vector machines (SVM) achieve this by choosing critical points that are essential for defining the hyperplane and are called support vectors. The algorithm's name, Support Vector Machine, is hence suitable.

2.2.3 Multinomial Naive Bayes

The Multinomial Naive Bayes algorithm operates on the foundation of Bayes' Theorem. It uses prior probabilities and conditional probabilities to compute the posterior probabilities. This

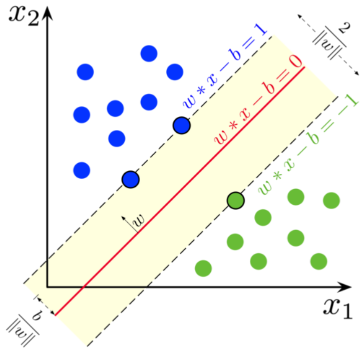

Figure 2.2: SVM with Hyperplane [6]

approach for supervised machine learning is mostly employed in the field of text classification. When dealing with high-dimensional data, it produces good results. The detection of spam emails is one of the most often utilized applications for this algorithm. It is a well-liked option for these kinds of classification assignments because of its efficaciousness in managing vast feature spaces.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{2.2}$$

2.2.4 Random Forest

Several decision trees that have been trained on different portions of the original dataset make up the ensemble classifier known as Random Forest. Random Forest is a type of ensemble learning model that combines several models to make judgments as a group. As opposed to depending only on one decision tree, this method creates many decision trees. Every tree is built separately using a different chunk of training data and a randomly selected set of characteristics. This divergence ensures that the trees capture various aspects of the data, hence preventing overfitting. Every decision tree in the Random Forest produces a unique forecast when faced with a new data point for prediction. The final forecast is then derived by aggregating results via voting or by summing the forecasts of each tree separately. The Random Forest approach reduces the possibility of overfitting by combining predictions from several decision trees, increasing the model's overall accuracy and robustness. Because Random Forest selects subsets of characteristics at random for each tree, it effectively handles datasets with a large number of features. This helps mitigate some of the drawbacks associated with high dimensionality.

2.2.5 GradientBoosting

A well liked machine learning method called gradient boosting turns several poor prediction models usually decision trees into one powerful predictive model. Gradient Boosting produces trees consecutively, with each tree fixing the mistakes of the preceding one, in contrast to Random Forest, which builds trees independently. Gradient boosting is based on the iterative fitting of new trees to the ensemble's residuals, with each iteration progressively improving predictions.

2.3 Deep learning

Deep learning is a subarea of machine learning that try replicates the complex decision-making skills of the human brain via multi-layered neural networks known as deep neural networks. The term ”deep learning” was actually coined by Rina Dechter[7] in 1986[8]. Indeed, there are differences between deep learning and machine learning, despite deep learning being a subset of machine learning. Machine learning typically relies on structured, labeled data and requires manual feature extraction. On the other hand, deep learning algorithms can handle unstructured data, like text and images, and have the ability to automatically extract useful features from it, making deep learning ideal for tasks with complicated unstructured data. Deep-learning algorithms are quite complex, and there are different types of Deep-learning architectures these include Convolutional neural networks (CNNs) [9] predominantly used for computer vision tasks, Recurrent neural networks (RNNs)[10] typically utilized for Natural language processing and sequential data analysis, Transformer models[11] etc. We will discuss these architectures in greater detail later in this chapter. The concept of deep learning originated from biological neurons. we will discuss it in the next subsection.

2.3.1 Biological neuron

Before we discuss artificial neurons, let's take a quick look at a biological neuron.
Biological neurons are vital parts of the nervous system in every living creature, such as humans. A biological neuron generally consists of dendrites, a cell body (soma), and an axon. Dendrites acquire signals from other neurons and afterward, combine them into the cell body[13]. If the combined signal crosses a specific limit, an action potential is generated along the axon, allowing the neuron to communicate with neighboring neurons via synaptic connections. This

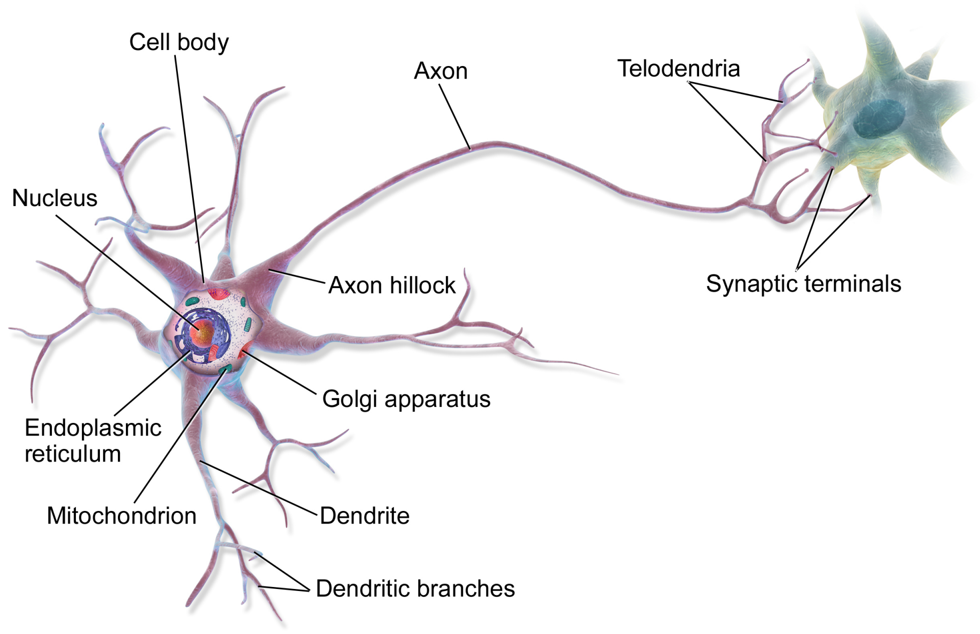

Figure 2.3: Biological neuron [12]

complicated network of neurons is the foundation of information processing in the brain and serves as an inspiration for artificial neural networks.

2.3.2 Artificial neural networks

Frank Rosenblatt[14] invented the Perceptron in 1957, which is one of the earliest and simplest artificial neural network architectures.

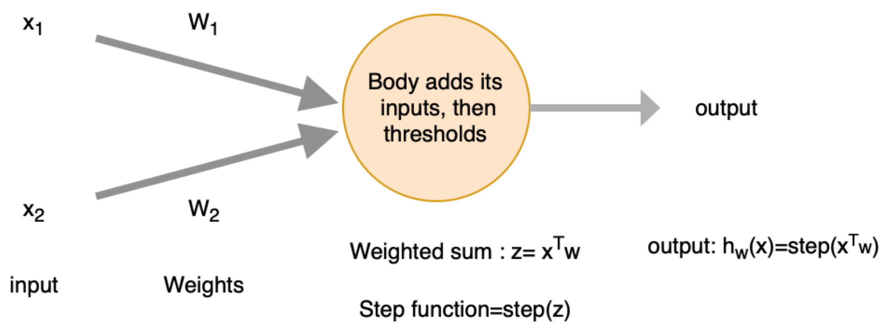

Figure 2.4: The Perceptron[15]

The Heaviside step function is the most common one utilized by perceptrons. It calculates a linear combination of the inputs and, if the result reaches a particular threshold, it returns the positive class. Otherwise, it returns a negative class.

The multilayer perceptron (MLP) is an evolved version of the single-layer perceptron. It's also known as Artificial Neural Networks (ANNs). Artificial Neural Networks (ANNs) are mathematical models based on the structure and operation of biological neural networks seen in the human brain. ANNs are composed of interconnected nodes referred to as artificial neurons or perceptrons, which are organized into layers. An ANN has three primary types of layers: input, hidden, and output. In an ANN, data passes through the network's components from the

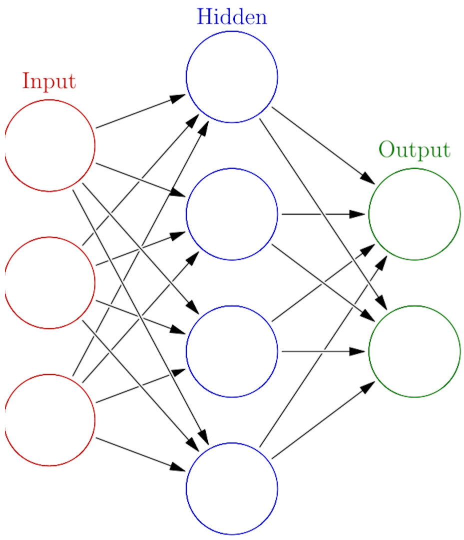

Figure 2.5: Artificial Neural Network[16]

input layer, where data is provided, through one or more hidden layers, where computations are conducted, and finally to the output layer, which generates the network's predictions or classifications. Every link between neurons in adjacent layers is assigned a weight. Throughout the learning phase, these weights are altered based on how well the network performs on a training dataset, generally using optimization techniques like gradient descent.ANNs can learn difficult patterns and correlations in data. ANNs have been effectively used in a variety of applications, including image and speech recognition, natural language processing, and medical diagnosis.

2.3.3 RNN

Text data and time series are examples of sequence data that is processed or classified using recurrent neural networks, or RNNs. RNNs' hidden state, which holds onto some information about a sequence, is its essential component. Nevertheless, RNNs have trouble storing information for later stages when working with lengthy sequences. This causes problems such as the exploding gradient problem (gradients becoming too huge) or the vanishing gradient problem (gradients becoming too small during backpropagation). Advanced RNNs, such as Long Short-Term Memory (LSTM), BiLSTM, and GRU, have been created to overcome these difficulties. RNNs can handle lengthy sequences with more effectiveness thanks to their sophisticated structures that assist reduce issues like vanishing or expanding gradients.

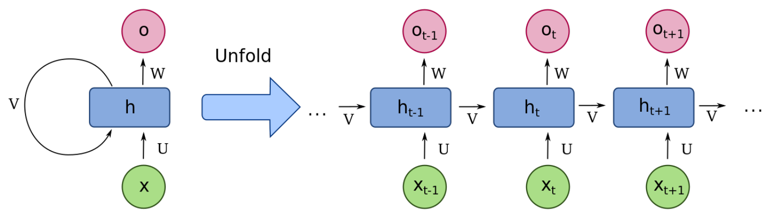

Figure 2.6: Recurrent neural network[17]

2.3.4 Transfer Learning

Training a model for one task and then applying its learned parameters to other tasks in the other domain is known as transfer learning. Natural language processing (NLP) is seeing a rise in the use of transfer learning after introducing transformer architecture in the paper "Attention All You Need"[18]. But transfer learning was very popular for computer vision in those days nowadays as well. In this case huge amount of text data is fed into the model and the model learns from this data. Then these pretraining weights or parameters are used for the start point of another kind of NLP task. Bert's model uses the encoder part of this transformer architecture. Figure 2.7 show the concept of Transfer learning.

Bert

The encoder part of the transformer architecture is used by Bert (Bidirectional Encoder Representations from Transformers)[20]. Bert is trained using an unsupervised method on vast volumes of text input, which is the fundamental idea. Two approaches can be used for pre-training: either randomly mask a portion of the text's tokens, and then ask the model to predict the tokens that are masked, or it may involve predicting the following sentence. It can be concluded that this model understands language context based on pre-training. It can therefore be

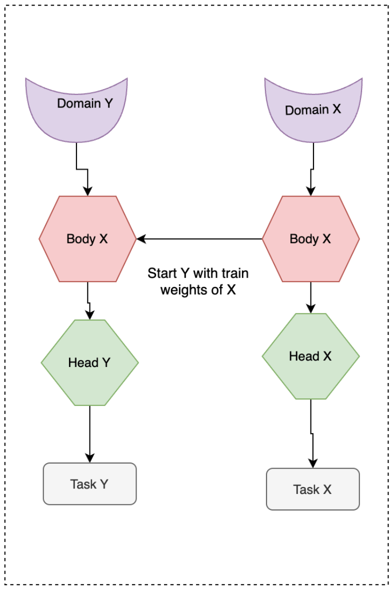

Figure 2.7: transfer learning [19]

used for a variety of downstream NLP tasks. The Bert base consists of 12 attention heads, 12 transformer blocks with a hidden size of 768, and 110 million parameters in total.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.3}$$

Where:

Q represents the query matrix.

K represents the key matrix.

V represents the value matrix.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.4}$$

A list of words or subwords is fed into BERT. With pre-trained embeddings, it turns each token into a numerical representation. In the BERT paradigm, the input sequence passes through several transformer layers. These tiers record relationships between tokens and contextual data by performing some mathematical operation like positional encoding, and self-attention mechanism. The input sequence starts with the addition of a unique token ([CLS]). The comprehension of the entire sequence is summed up in the token's final output representation. A classification head processes the [CLS] token's output representation. Fully linked layers make up this head,

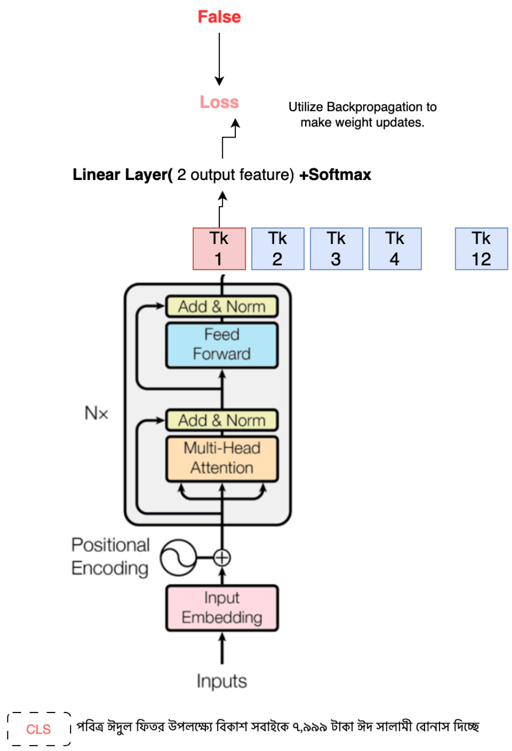

Figure 2.8: Text classification training [18]

which converts the [CLS] token representation into a probability distribution across all possible classes. Using a loss function, the model is trained by determining the actual labels with the expected probability distribution. Backpropagation is used to edit the model's parameters to minimize the loss and enhance classification performance.

2.4 Natural language processing

Natural language processing (NLP) is concerned with allowing computers to understand, decode, and generate human language in a meaningful and contextually appropriate manner. Natural Language Processing (NLP) offers a wide range of applications in many businesses and domains. NLP applications include Text classification, Text-to-image generation, Named Entity Recognition (NER), Parsing, Sentiment Analysis, Machine Translation, Question Answering, Virtual Assistants, Text Summarization, speech recognition, natural-language understanding, and natural-language generation[21]. The next section describes the fundamental NLP roles used in this research.

2.4.1 Tokenization

Tokenization, in Natural Language Processing (NLP), is the process of dividing a sequence of text into smaller fragments known as tokens. These tokens might be as short as individual letters or as long as the whole words. This method is important because it helps machines comprehend human language by splitting it down into fragments that are easier to handle. In natural language processing, the following kinds of tokenization are often used: word tokenization, character tokenization, and subword tokenization.

2.4.2 Text cleaning

Text preprocessing is a crucial part of the NLP pipeline. Text cleaning, also known as text preprocessing, is the act of preparing raw text data for analysis by eliminating noise and unnecessary details. While text cleaning, particular symbols, and tags used in HTML, as well as other non-linguistic elements, are removed from the initial dataset since they provide no information to the model or just act as noise for the model. Some common approaches used in text cleaning are Lowercasing, Removing Special Characters and Punctuation, Removing Stopwords, Stemming, and Lemmatization, Handling Numerical Data, etc.

Punctuation

Punctuation removal is a text cleaning method that removes punctuation signs from text data. While punctuation is crucial for reading, it is generally unimportant or even disruptive for certain natural language processing (NLP) tasks. Removing punctuation from text reduces noise, simplifies tokenization, and improves model performance. some commonly used punctuation marks in Bangla include:
, , | , ” ” , ’ ’, ! , : ,- , ;, ?

Stopwords

Stopword removal is a text preprocessing procedure that removes common words, known as stopwords, from a piece of text. Stopwords are words that appear frequently in a language but typically do not carry significant meaning.

Stemming

Stemming is a text normalization technique in natural language processing (NLP) that reduces words to their root or base form, also known as the stem. The stem may not necessarily be a genuine word, but it captures the essence of the term.

2.5 Feature extraction methods for text classification

Machine learning algorithms and deep learning methods are incapable of understanding raw text data because they can only work with numerical values. To ensure effective processing, textual information must be converted into numerical representations. That is why feature extraction is a crucial step in text classification, where the goal is to convert raw text data into a suitable format for various learning algorithms. Effective feature extraction methods are required for developing accurate and efficient text categorization models. Various frequently used feature extraction methods in text classification tasks include Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), word embeddings, etc.

2.5.1 Bag of words

Bag of words (BoW) is a conventional text representation approach that is commonly used in NLP applications to extract features from input data. Represent the text under consideration as a collection of words, ignoring order and context. BoW assigns words unique integer IDs ranging from 1 to $|V|$, where V represents the vocabulary set. Every single document is then transformed into a vector of $|V|$ dimensions. We just score each word in V based on how many times it appears in the document.

2.5.2 TF-IDF term

TF-IDF, short for term frequency-inverse document frequency, is a method designed to quantify the importance of a specific word relative to other words within a document and across an entire corpus. It achieves this by combining two key metrics: TF (term frequency) and IDF (inverse document frequency). TF quantifies how frequently a term or word appears within a particular document, while IDF (inverse document frequency) measures the importance of the term throughout a corpus. The TF-IDF (Term Frequency-Inverse Document Frequency) formula combines term frequency (TF) and inverse document frequency (IDF) to determine the importance of a term in a document relative to a corpus.

Mathematically, the TF-IDF formula is given by:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

Where: - t term or word. - d document. - D entire corpus.

The TF (Term Frequency) for term t in document d is calculated as:

$$\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

The IDF (Inverse Document Frequency) for term t in corpus D is calculated as:

$$\text{IDF}(t, D) = \log \left(\frac{\text{Total number of documents in corpus } D}{\text{Number of documents containing term } t} \right)$$

The TF-IDF score represents the importance of term t in document d relative to the entire corpus D. It is the product of its term frequency and inverse document frequency.

2.6 Conclusion

In conclusion, this chapter has given a general overview of text categorization, highlighting the importance of machine learning models such as BERT and natural language processing (NLP). Even if the area has advanced recently, issues like model interpretability and domain adaption still exist. However, this foundational work guides our study toward improving text categorization methods and their real-world applications in the next chapters.

Chapter 3

Literature Review

3.1 Introduction

The previous chapter presented a brief overview of the research. This chapter summarizes the literature review related to this research.This chapter begins with some examples of fake news and its impact. A literature survey is provided in the next section. Finally, we conclude.

3.2 Related works

I've read several works on the issue of spotting false news and the following are some of the notable contributions:
The study of [22] Mugdha et al. evaluated the performance of fake news detection using nine different machine learning algorithms including Support Vector Machine (linear), Logistic Regression, Multi-Layer Perceptron, Random Forest, Voting Ensemble Classifier, Gaussian Naive Bayes, Multinomial Naive Bayes, AdaBoost, Gradient Boosting. Their investigation showed that Gaussian Naive Bayes performed the best among all classifiers, with Logistic Regression coming in second spot with a performance of 87.42%. To train the classifiers, the researchers employed 269 Bangla fake news and 269 actual news pieces, they gathered the data for their study from several sources, and then extracted features from the data using a technique called Term Frequency (TF) Inverse Document Frequency (IDF). [23]Employed a convolutional neural network (CNN) to detect bogus news to demonstrate their work. The English dataset was used for research, and a 98% performance score was obtained. [24]Fake news is classified using a variety of deep learning algorithms, such as Convolutional Neural Networks

(CNN), Long short-term memory (LSTM), Bidirectional LSTM (Bi-LSTM), and LR, as well as other types of classical machine learning methods, such as KNN, RF, SVM, etc. Using Glove and FastText pre-trained word embedding, fraud news can be identified. The study employed the BanFakeNews[25] dataset and found that the Bi-LSTM tracked 96% of the data appropriately.RI Rasel et al.[26] experimented with multiple Machine Learning models(LR, KNN, SVM, DT, MNB, and Adaboost) with performance scores up to 95.2% average accuracy and recall of 83.3% for fake news. They used the BanFakeNews dataset and also gathered 500 fake news via the Internet Archive, In total, there were 2.3k false news and 2.3k actual news. This study also looked at DNN models including CNN-LSTM, Bi-LSTM, LSTM, and CNN-BiLSTM. Additionally, They investigate transformer models (Bangla-Bert-Base and mBert) with accuracy of 93.3% and 93.8% respectively. BiLSTM was the best-performing model, with a 95.9% accuracy rate. AJ Keya et al.'s research [27] 1.3k cases of fake news were gathered from the "BanFakeNews"[25] dataset. used a transfer learning-based method for text augmentation to boost the volume of fake news data. They created synthetic data by inserting and substituting words into a pre-trained multilingual BERT base uncased model. They created 2,700 fake news content items, resulting in a total of 4,000 fake news data items. To achieve a balanced augmented dataset, they performed undersampling on the authentic news data. Their proposed best model, AugFake-BERT, trained on the augmented dataset, achieved an accuracy of 92.45%, precision of 92.86%, recall of 91.23%, and an F1-score of 91.85%.

3.3 Conclusion

We have tried to highlight some significant advancements in the field of Bangla fake news detection in this chapter.

Chapter 4

Proposed Methodology & Implementation

4.1 Introduction

The chapter ”Proposed Methodology and Implementation” provides a complete picture of how we are carrying out our research. In this chapter, we will talk about datasets, preprocessing of datasets, and the adoption of machine learning and transfer learning frameworks.

4.2 Methodology

Figure 4.1 illustrates the methodology we proposed. Two datasets were employed in this study: the BanFakeNews[25] dataset, which was gathered from Kaggle, and the other was created by web scraping from Fact check websites including Boom [28], jachai[29], and YouTube [30] news video titles. Following that, data went through several cleaning stages in the preparation phase. Next, the dataset was divided into two groups: 20% was used for testing, and the remaining 80% for training. For feature extraction, techniques such as Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) so that machine learning classifiers could utilize them. Attention-based feature extraction was carried out within the BERT model's architectural framework. To detect fake news, our dataset was run through the BERT model and several machine-learning classifiers. Lastly, the test data was used to assess the model's performance.

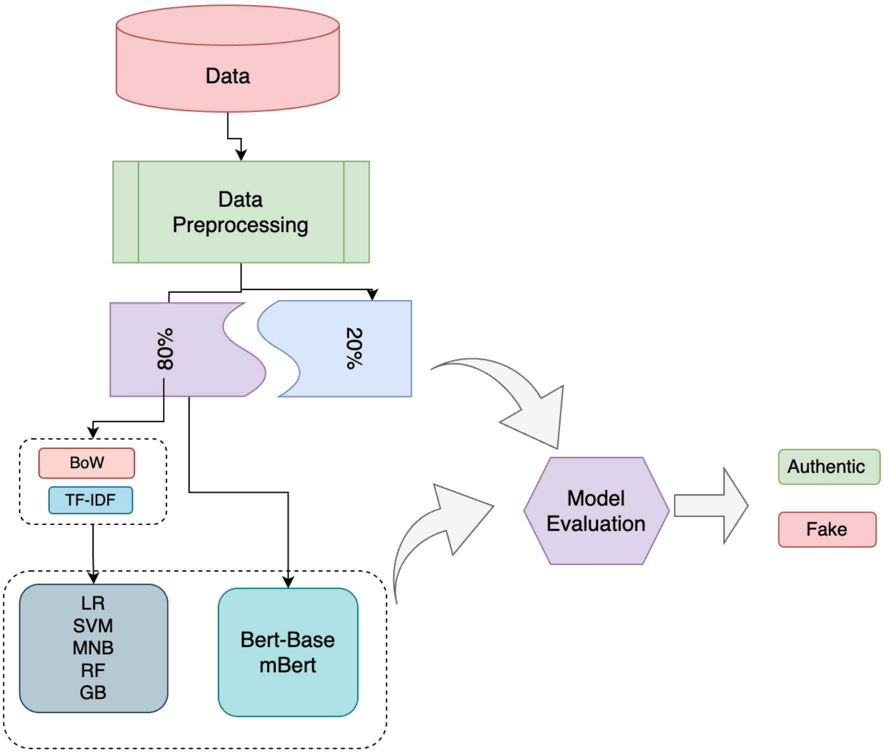

Figure 4.1: A summary of the proposed methodology

4.3 Dataset Descriptions

4.3.1 BanFakeNews

The "BanFakeNews[25]" dataset, is accessible to everyone on Kaggle. This dataset contains 48,678 examples of True news, reflecting the abundance of True news available across numerous news portals. Conversely, it contains only 1,299 cases of fake news, indicating how rare such events are.

To mitigate the class imbalance, we performed under-sampling based on the label column. Specifically, we selected the same number of authentic news samples as fake news instances. This under-sampling technique ensured a more balanced representation of both classes in the dataset. In This research, we worked with 1299 Authentic samples and 1299 Fake samples.

A balanced dataset is necessary for accurate fake news detection; otherwise, biased models, poor performance metrics (such as low sensitivity for the minority class), misleading evaluation

Table 4.1: Contents of Dataset

File Name	Authentic-48k.csv	Fake-1k.csv
Number of Contents (Rows)	48,678	1299
Columns	7	7
Features	articleID, domain, date, category, headline, content, label	articleID, domain, date, category, headline, content, label

	articleID	domain	date	category	headline	content	label
0	1	jagonews24.com	2018-09-19 17:48:18	Education	হট্টগোল করায় বাকৃবিতে দুইজন বরখাস্ত, ৬ জনকে শোকজ	গত ১৭ সেপ্টেম্বর বাংলাদেশ কৃষি বিশ্ববিদ্যালয়ে ...	1
1	2	jagonews24.com	2018-09-19 17:48:19	National	মালয়েশিয়ায় কর্মী পাঠানোর ব্যবস্থা নেয়ার সুপারিশ	বাংলাদেশের বৃহৎ শ্রমবাজার মালয়েশিয়ায় আবার শ্রম...	1
2	3	jagonews24.com	2018-09-19 17:48:20	National	প্রেমের প্রস্তাবে রাজি না হওয়ায় স্কুলছাত্রীকে ...	নরসিংদীর মনোহরদীতে প্রেমের প্রস্তাবে রাজি না হ...	1
3	4	jagonews24.com	2018-09-19 17:48:21	Crime	মেডিয়েশনই মামলাজট নিরসনের পথ : বিচারপতি আহমেদ ...	সুপ্রিম কোর্টের হাইকোর্ট বিভাগের বিচারপতি আহমে...	1
4	5	jagonews24.com	2018-09-19 17:48:21	National	টকশোতে বক্তব্য দিতে গিয়ে জাপা নেতার মৃত্যু	মাদারীপুর সদরের উপজেলার লেকেরপাড়ে একটি বেসরকার...	1
5	6	jagonews24.com	2018-09-19 17:48:23	Entertainment	সোনালী দিনের নায়িকা সুচন্দার জন্মদিন আজ	বিশিষ্ট অভিনেত্রী ও চলচ্চিত্র নির্মাতা কোহিনূর...	1
6	7	jagonews24.com	2018-09-19 17:48:23	Finance	হস্তান্তরিত বস্ত্রকলে সরকারের সহায়তার দাবি	শ্রমিক-কর্মচারীদের ব্যবস্থাপনায় হস্তান্তরিত বস...	1

Figure 4.2: Authentic-48k.csv

	articleID	domain	date	category	headline	content	label
0	1	channeldhaka.news	2019-03-14T13:34:14+00:00	International	মুরগির হামলায় শেয়াল নিহত	বাংলায় একটা প্রবাদ আছে, শেয়ালের কাছে মুরগী বর্...	0
1	2	earki.com	সেম্বর ১৭, ২০১৮	Miscellaneous	বিটিভিতে যেবার আমি ইন্টারভিউ দিতে গেলাম	BTV থেকে লোকজন আসছে, ইন্টারভিউ নিবে। চারজনের টি...	0
2	3	earki.com	২০:৩৯, জানুয়ারি ১৪, ২০১৯	Miscellaneous	বিদেশ থেকে উন্নতমানের বিরোধীদল আমদানি করার পরা...	অদ্ভুত বিরোধীদলহীনতায় ভুগছে সরকার। এ এক অন্যরক...	0
3	4	channeldhaka.news	2018-06-30T15:56:47+00:00	Sports	অবসর নেয়ার ঘোষণা দিলেন মেসি !	রাশিয়া বিশ্বকাপ নকআউট পর্বে ফ্রান্সের সাথে ৪-৩...	0
4	5	motikontho.wordpress.com	2013-03-05T21:55:45+00:00	Miscellaneous	মাদারফাকার নহে, ব্রাদারফাকার: সাকা | দৈনিক মতি...	নিজস্ব মতিবেদক'মাদারফাকার নহে, আমি ব্রাদারফাকা...	0
5	6	channeldhaka.news	2018-12-24T18:17:50+00:00	Miscellaneous	বিয়ের পিড়িতে বসছেন মিয়া খলিফা ! ছেলে কুমিল্লার	বিয়ের সানাই বাজতে চলেছে শীঘ্রই ! সব জল্পনা কল...	0
6	7	earki.com	2019-03-14T02:33:32+00:00	Miscellaneous	জুম্মার নামাজে সবচেয়ে বেশি মসজিদে যায় নোয়াখ...	এক গবেষণা থেকে জানা গেছে, বাংলাদেশের অন্যান্য ...	0

Figure 4.3: Fake-1k.csv

results, and ultimately inaccurate predictions may result. Our goal was to train our models on a more representative collection of data for both classes by correcting the imbalance in the dataset through under-sampling..

For this experiment, we combined headline and content attributes and used them for further analysis.

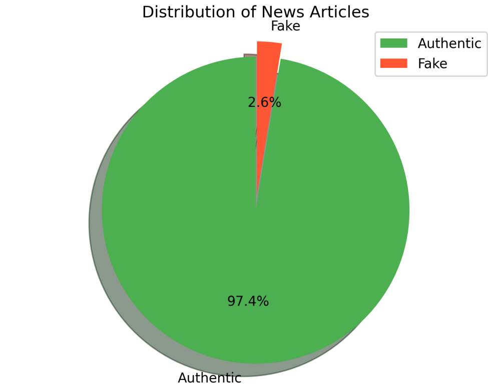

Figure 4.4: Default label ratio

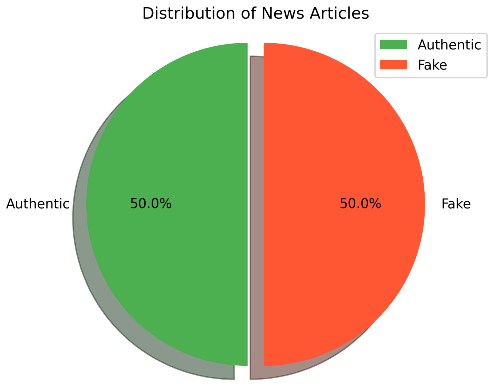

Figure 4.5: After performing undersampling label ratio

4.3.2 Dataset Construction

Using a tool called Beautiful Soup in Python, we obtained bogus Bangla news headlines from fact-checking websites such as Boom[28] and Jachai [29]. These websites analyze news stories and postings from Facebook to determine whether they are accurate before marking them. Using this method, we found 1922 fraudulent posts and headlines. 1913 fake articles remained after deleting duplicates, English titles, and extremely brief titles. We searched YouTube[30] for actual news headlines as their data is easily reusable through a feature known as the YouTube API. We selected 5,000 news video titles from several Bangla news channels on YouTube, including Independent, ATN Bangla News, Desh TV News, and Somoy TV Figure 4.6 Data collection website and Youtube channel. Once we removed duplicates, English titles, and extremely brief titles, we were left with actual news from 1995 real news headlines. After combining the two sets, we had 3908 samples in total, this dataset contains two columns: the text and whether it's fake or real. Figure 4.7 illustrates the whole process of data creation steps.

Figure 4.6: Data collection website and Youtube channel

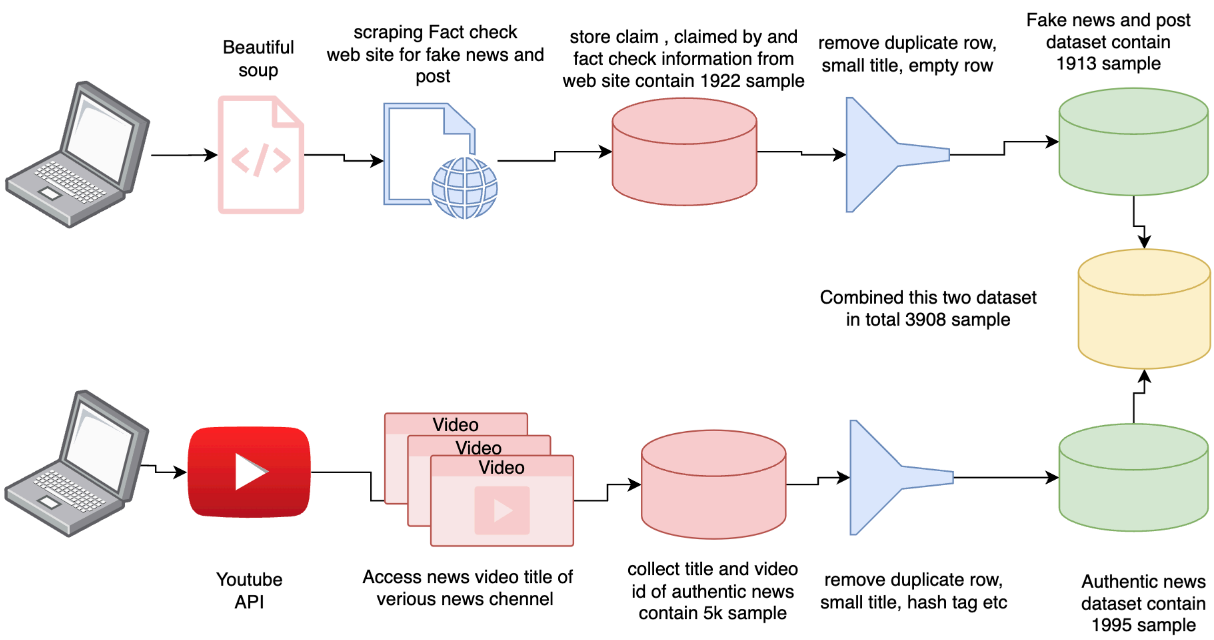

Figure 4.7: Dataset collection steps

A simple representation of our dataset is in Figure 4.8. We divided the labels into two categories: actual (1) and fake (0), with 1995 and 1913 cases respectively being labeled as real and fake. Our study was limited to examining the news headlines.

Figure 4.9 shows Frequency vs text the length of Authentic news and fake news.

Figure 4.8: Dataset Visualization

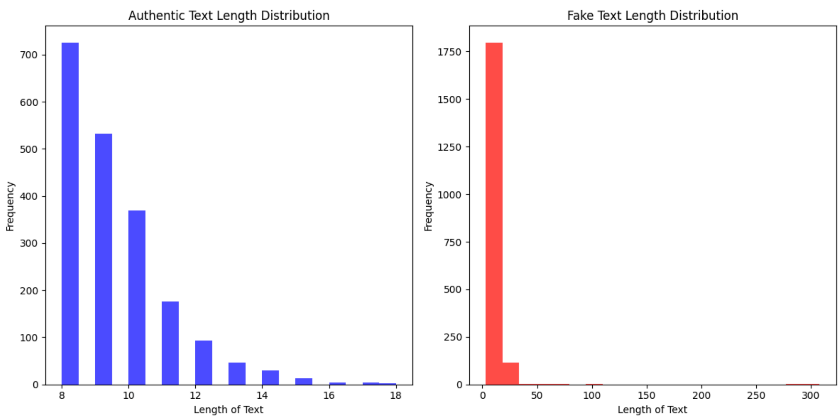

Figure 4.9: text length of each class Visualization

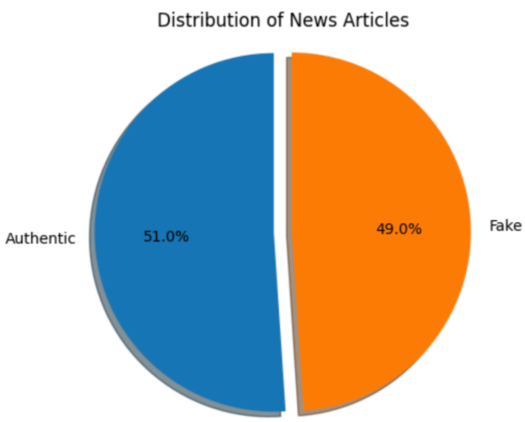

Figure 4.10: Label ratio of constructed dataset

4.4 Dataset Preprocessing

Data preprocessing is an important step in preparing raw text data for analysis in the context of Bangla fake news identification. This section describes the processes performed to clean and standardize the collected datasets, assuring that they are suitable for future analysis.

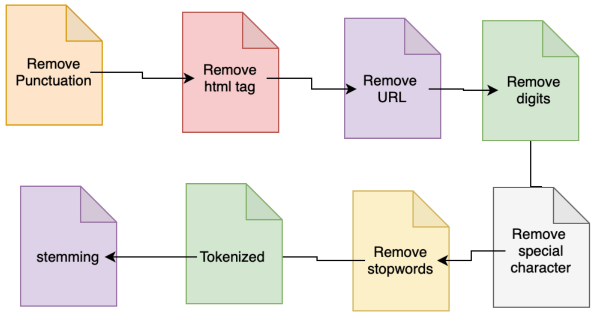

Figure 4.11: Data Preprocessing steps

During the preprocessing phase, our data goes through several cleaning steps. Initially, we removed punctuation marks, numeric values, English alphabets, URLs, HTML tags, special characters, and stop words from the raw dataset. This preprocessing process aimed to eliminate noise and irrelevant information from the text. Following the cleaning phase, we performed tokenization, splitting the text into individual words or tokens. Finally, we applied stemming to reduce words to their root or base form. Through these preprocessing steps, we converted the original dataset into a clean, tokenized, and stemmed format, ready for subsequent analysis.

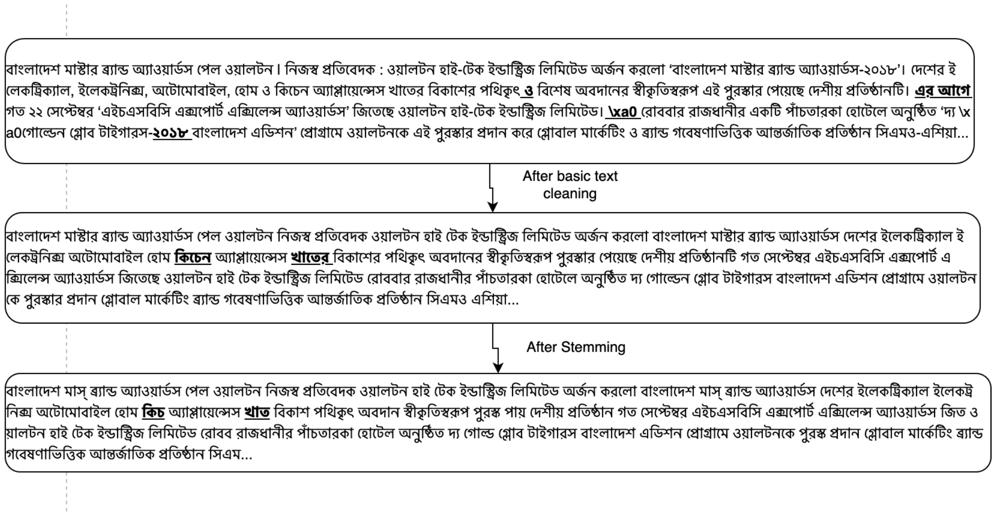

Figure 4.12: Text Cleaning

4.5 Feature Extraction

Because they are limited to working with numerical values, machine learning algorithms, and deep learning techniques are unable to comprehend text data. Textual data needs to be represented numerically in order for processing to be done efficiently. Feature extraction, which aims to transform text data into a format that is appropriate for a variety of learning algorithms, is hence an essential stage in the text classification process.

4.5.1 Bag Of words

The NLP technique known as ”Bag of Words” (BoW) counts the frequency of terms in a document to represent text. It treats every word separately, ignoring context and word order. The result is a vector where each dimension corresponds to a word, and the value represents its frequency in the document.

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [1, 1, 0, ..., 6, 7, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 3, ..., 1, 1, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

Figure 4.13: BoW matrix

4.5.2 TF-IDF

A method for determining a word’s importance within a text and throughout a corpus of documents is called term frequency-inverse document frequency, or TF-IDF. It combines the two primary metrics TF, which determines a word’s frequency of occurrence in a document, and IDF, which assesses a word’s significance throughout the corpus. TF evaluates local significance within a document, whereas IDF evaluates global significance throughout the entire corpus.

```
array([[0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.05254023, 0.05254023, 0.        , ..., 0.25364813, 0.29673673,
        0.        ],
       ...,
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.17444044, ..., 0.0361471 , 0.03624652,
        0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
        0.        ]])
```

Figure 4.14: TF-IDF matrix

4.6 Machine Learning Classifiers

Using this approach, we cleaned the dataset by performing stemming and removing stopwords, digits, special characters, and punctuation. BoW and TF-IDF are employed in feature extraction. Divide the dataset 80:20 for training and testing after the first steps. then conducted trials using a variety of machine learning classifiers, including Multinominal Naive Bayes, Support Vector Machine, and Logistic Regression, Random Forest, GradientBoosting. Found suitable hyperparameters For Machine learning classifiers via randomized search.

4.7 Bert Models

4.7.1 Bangla-Bert-Base

Bangla Bert, which was developed especially for the Bengali language, is used in this work. 'sagorsarker/Bangla-bert-base' [31] is the name of this model, which has been pre-trained using Bengali Wikipedia Dump Dataset and Bengali corpus from OSCAR4.

4.7.2 mBert

104 widely-used languages with large amounts of Wikipedia data were used in the m-BERT training process, including Bengali. The model that we utilized in our dataset was the 'bert-base-multilingual-cased'[32] model, which we adjusted for batch size, learning rate, and epochs.

4.8 Conclusion

This chapter provides a detailed discussion of the research methodology we suggest. It provides a thorough explanation of every stage, beginning with data collection, dataset construction, and analytical preparation. We also describe how we take significant features out of the data. We also explain the Transformer models and machine learning techniques we employ. Each step is carefully planned to ensure our analysis is accurate and reliable.

Chapter 5

Result & Performance Analysis

5.1 Introduction

In this study, two distinct feature extraction methodologies were investigated to improve the performance of machine learning models on the BanFakeNews and Custom datasets. These methodologies were applied to five diverse ML models, enabling a comprehensive evaluation. The outcomes of each ML classifier and Bert models were meticulously examined and thoroughly discussed in this section. On the test data, several performance metrics, including recall, f1-score, accuracy, and precision, were used to evaluate the effect of these strategies on model performance.

5.2 Evaluation Metrics

Precision, recall, accuracy, F1-score, and the confusion matrix are among the assessment measures that are frequently used to evaluate the performance of models. The phrases True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) must be understood in the context of categorization before diving into the explanation of these assessment metrics.

True Positives (TP): These are instances where the model correctly predicts the positive class (1) and the actual value of the data point is also positive (1).

True Negatives (TN): These are instances where the model correctly predicts the negative class (0) and the actual value of the data point is also negative (0).

False Positives (FP): When the model predicts the positive class (1) but the actual value of the data point is negative (0), this is known as a false positive. The model predicted the positive class, which is the reason why the term ”positive” is used, and the term ”false” is used because the model produces an incorrect prediction.

False Negatives (FN): While the model predicts the negative class (0) but the data point’s true value is positive (1), this is known as a false negative. As with false positives, the term ”false” denotes an incorrect prediction produced by the model, while ”negative” denotes the prediction of the negative class by the model.

In summary, these metrics are used to quantify the performance of classification models by comparing their predictions with the actual labels of the data points. Now that we are aware of these metrics, we can go on to discuss each evaluation indicator and how it helps determine the efficacy of the model.

5.2.1 Confusion Matrix

A crucial evaluation tool for binary classification models is the confusion matrix, which provides a summary of the model’s predictions concerning the actual data labels. True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are its four fundamental components. These parameters allow for the computation of important performance metrics including Precision, Recall, Accuracy, and F1-score, which provide for a thorough and quantitative evaluation of the model’s effectiveness in differentiating between the two classes.

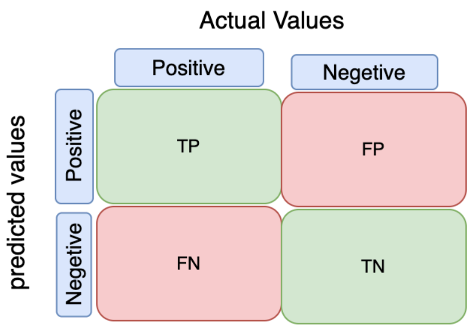

Figure 5.1: Confusion Matrix for the Binary Classification [33]

5.2.2 Accuracy

Especially in classification tasks, accuracy is a critical performance indicator for machine learning systems. It calculates the ratio of the algorithm's accurate predictions to all of the predictions made. It's necessary to remember that accuracy by itself might not give a whole view of the model's performance, particularly in cases where datasets are imbalanced. When evaluating classification problems that are well-balanced, non-skewed, and free of class imbalance, accuracy is a suitable option.

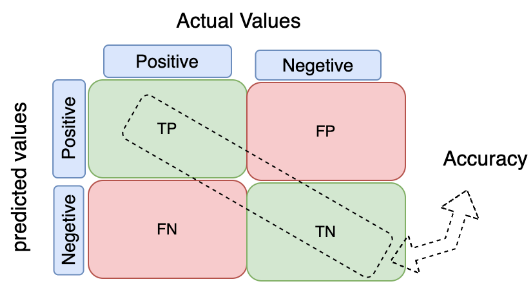

Figure 5.2: Accuracy[33]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

5.2.3 Precision

In binary classification tasks, precision is a performance indicator that is used to assess how well a model can identify positive examples (true positives) among all instances that the model predicts to be positive (both true positives and false positives). Put differently, precision quantifies how well the model predicts positive outcomes. A high precision score means that the model is more accurate in recognizing positive cases and produces fewer false positive predictions.

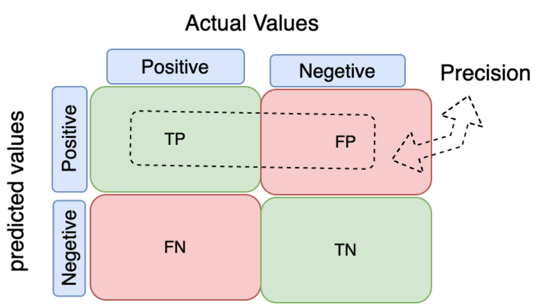

Figure 5.3: Precision[33]

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

5.2.4 Recall

Recall is another crucial performance parameter applied to binary categorization tasks; it is sometimes referred to as sensitivity or true positive rate. The assessment gauges the model's capacity to accurately detect positive examples, or true positives, among all genuinely positive occurrences, including false negatives.

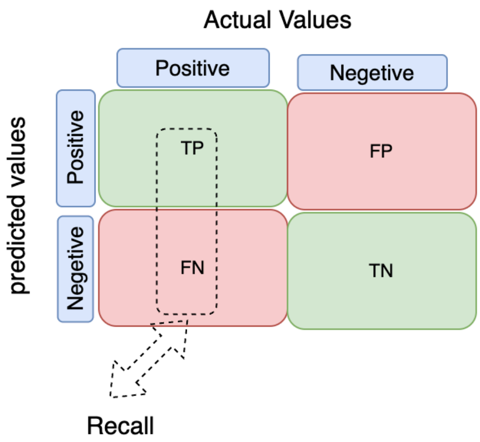

Figure 5.4: Recall[33]

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

When a model's recall value is high, it means it can accurately identify positive examples in the dataset. Put another way, the model is more likely to accurately classify a positive data point when it comes across one. Conversely, a low recall score indicates a tendency for the model to overlook positive cases and a reduced ability to correctly identify all of the positive cases.

5.2.5 F1-Score

The harmonic mean of recall and precision is the F1 score. Unlike ordinary averages, the harmonic mean is not sensitive to very big numbers. The F1 score helps the classifier maintain a balance between recall and precision. F1 score will be low if your recall is poor as well as low if your precision is poor. The F1-score will be near 1, suggesting that the model exhibits balanced performance concerning both the accuracy of positive predictions and the capacity to capture all positive cases, when precision and recall have similar values. The model's incapacity to strike a balance between the two criteria will be reflected in a lower F1 score if accuracy and recall diverge noticeably.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.4}$$

5.3 Experimental Results

The accuracy and F1-score for each classifier model are shown below, along with the confusion matrices and classification report of the top three classifiers.

5.3.1 BanFakeNews Dataset

Model Performance with TF-IDF Feature Extraction Method On BanFakeNews Dataset

To assess the performance of our model, the table 5.1 shows the accuracy, and F-1 score of all the classifiers with the feature extraction method. We used 20% of the dataset as our testing dataset and 80% as the training dataset. Here, we describe the performance of each of the classifiers-

Table 5.1: Model Performance for BoW feature Extraction approach on BanFakeNews Dataset

Model	Feature extractor	Accuracy%	F1%
LR	BoW	**94.04%**	**94.31%**
MNB	BoW	91.35%	91.71%
SVM	BoW	92.50%	92.71%
RF	BoW	89.62%	90.18%
GB	BoW	88.85%	89.38

The model performance table 5.1 presents the performance of different machine learning algorithms on the BanFakeNews dataset using the Bag-of-Words (BoW) feature extraction approach. With an astonishing accuracy of 94.04% and an F1 score of 94.31%, Logistic Regression (LR) stands out among these models, proving its resilience in differentiating between bogus and legitimate news stories. Moreover, Support Vector Machine (SVM) performs admirably, attaining an F1 score of 92.71% and an accuracy of 92.50%. But Random Forest (RF) and Gradient Boosting (GB) lag somewhat behind LR and SVM, suggesting that when using BoW feature extraction, the algorithm selection has a big impact on classification accuracy.

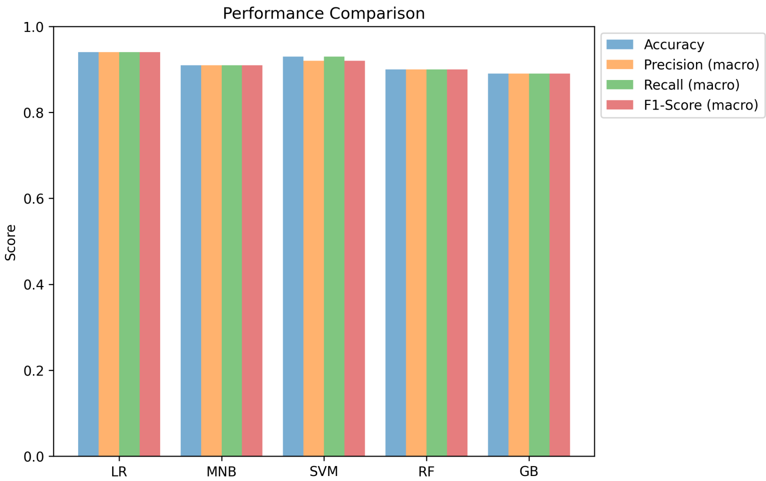

Figure 5.5: Comparison of Accuracy, Precision, Recall, F1-Score obtained applying Five classifiers LR, MNB, SVM, RF, GB for BoW feature Extraction approach on BanFakeNews Dataset

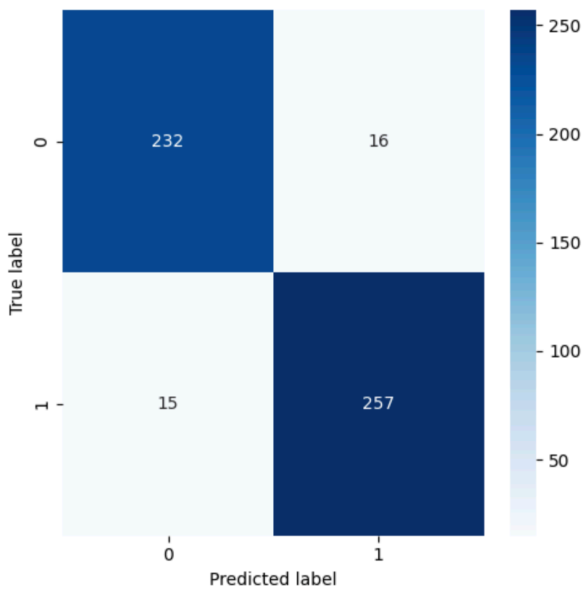

Figure 5.6: Confusion Matrix of LR (BoW) BanFakeNews Dataset

Figure 5.6 shows the confusion matrix of the Logistic Regression (LR) model trained on the BoW feature extraction for the BanFakeNews dataset and reveals its performance. It correctly identifies 232 fake news and 257 true news. However, it misclassifies 15 real news as fake and 16 fake news as true. These insights highlight LR's strengths and areas for improvement in distinguishing between fake and real news.

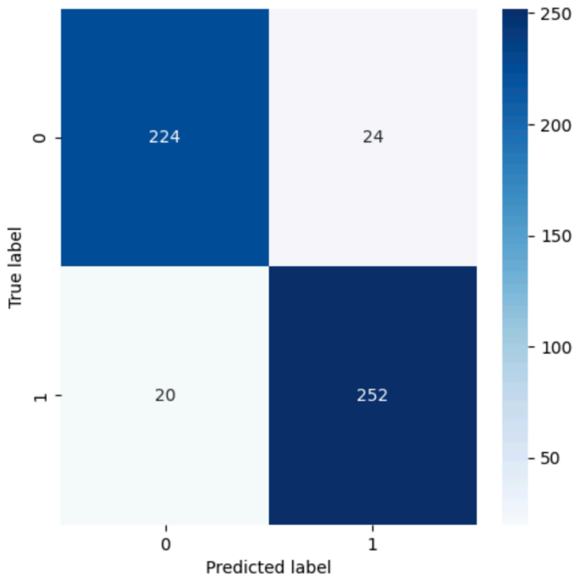

Figure 5.7: Confusion Matrix of MNB (BoW) BanFakeNews Dataset

Figure 5.7 shows the confusion matrix of the MNB model, trained on BoW features for the BanFakeNews dataset, revealing insights into its performance. It accurately identifies 224 fake news and 252 true news. However, it misclassifies 20 real news as fake and 24 fake news as true.

Figure 5.8 shows the confusion matrix of the Support Vector Machine (SVM) model trained on BoW features for the BanFakeNews dataset and reveals its performance. Out of the total instances, 233 fake news articles were correctly identified, along with 248 authentic news articles. However, 15 real news articles were misclassified as fake, and 24 fake news articles were misclassified as true. These results highlight both the SVM model's strengths and areas for improvement in accurately classifying news articles.

In Figure 5.9, titled "Classification report of LR (BoW) BanFakeNews Dataset", the LR (BoW) model demonstrates exceptional performance. With a precision of 0.94 for 'fake' news articles and 0.94 for recall, it effectively identifies fake articles. This success is attributed to the Bag-of-Words (BoW) feature extraction method, which captures essential textual features. Logistic regression, the chosen algorithm, excels in binary classification tasks like this. The model's

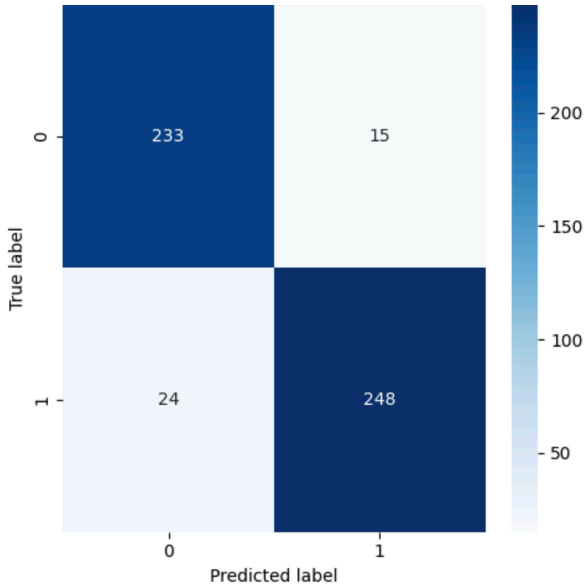

Figure 5.8: Confusion Matrix of SVM (BoW) BanFakeNews Dataset

Classification Report

	precision	recall	f1-score	support
0	0.94	0.94	0.94	248
1	0.94	0.94	0.94	272
accuracy			0.94	520
macro avg	0.94	0.94	0.94	520
weighted avg	0.94	0.94	0.94	520

Figure 5.9: Classification report of LR (BoW) BanFakeNews Dataset

consistent performance, with high precision, recall, and F1-score values of 0.96, 0.94, and 0.94, respectively, instills confidence in its reliability across the BanFakeNews Dataset.

Classification Report

	precision	recall	f1-score	support
0	0.91	0.94	0.92	248
1	0.94	0.91	0.93	272
accuracy			0.93	520
macro avg	0.92	0.93	0.92	520
weighted avg	0.93	0.93	0.93	520

Figure 5.10: Classification report of SVM (BoW) BanFakeNews Dataset

Model Performance with TF-IDF Feature Extraction Method On BanFakeNews Dataset

Table 5.2 shows Model Performance with Accuracy and F1 score on BanFakeNews Dataset

Table 5.2: Model Performance for TF-IDF feature Extraction approach on BanFakeNews Dataset

Model	Feature extractor	Accuracy%	F1%
LR	TF-IDF	**95.19%**	**95.36%**
MNB	TF-IDF	91.35%	91.71%
SVM	TF-IDF	**94.81%**	**94.95%**
RF	TF-IDF	90.38%	90.81%
GB	TF-IDF	86.73%	87.05

Transitioning to the TF-IDF feature extraction approach yields notable improvements across the machine learning models. Logistic Regression (LR) demonstrates enhanced performance, achieving an accuracy of 95.19% and an F1 score of 95.36%, surpassing its performance with BoW feature extraction. Similarly, Support Vector Machine (SVM) also exhibits significant improvements, with an accuracy of 94.81% and an F1 score of 94.95%. These results underscore the effectiveness of TF-IDF in capturing meaningful textual features, enabling models to better discern between fake and real news articles compared to the BoW approach.

Figure 5.12 shows the confusion matrix of the Logistic Regression (LR) model trained on the TF-IDF feature extraction for the BanFakeNews dataset and reveals its performance. It correctly identifies 238 fake news and 257 true news. However, it misclassifies 15 real news as fake and 10 fake news as true. These insights highlight LR's strengths and areas for improvement in distinguishing between fake and real news. Figure 5.13 shows the confusion matrix of the Support Vector Machine (SVM) model trained on TF-IDF features for the BanFakeNews dataset and reveals its performance. Out of the total instances, 239 fake news articles were correctly identified, along with 254 authentic news articles. However, 18 real news articles were misclassified as fake, and 9 fake news articles were misclassified as true. These results highlight both the SVM model's strengths and areas for improvement in accurately classifying

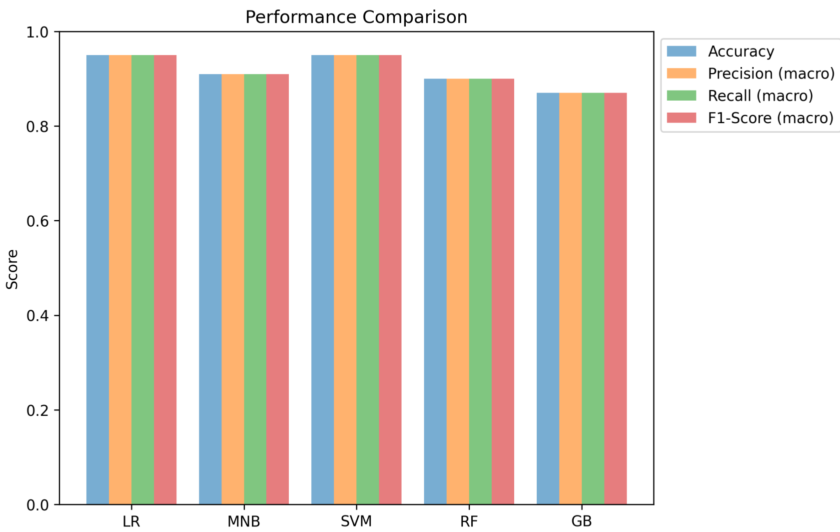

Figure 5.11: Comparison of Accuracy, Precision, Recall, F1-Score obtained applying Five classifiers LR, MNB, SVM, RF, GB for TF-IDF feature Extraction approach on BanFakeNews Dataset

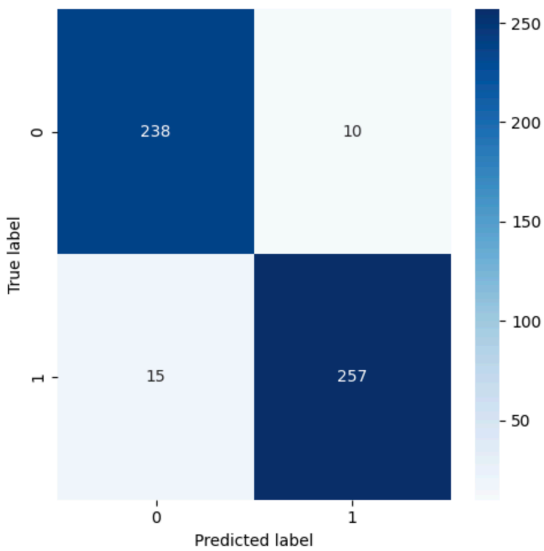

Figure 5.12: Confusion Matrix of LR (TF-IDF) BanFakeNews Dataset

news articles.

Figure 5.14 shows the confusion matrix of the MNB model, trained on TF-IDF features for the BanFakeNews dataset, revealing insights into its performance. It accurately identifies 226 fake news and 249 true news. However, it misclassifies 23 real news as fake and 22 fake news as true.

The Figure 5.15 ”Classification report of LR (TF-IDF) BanFakeNews Dataset”, we gain insights into the model’s performance. Notably, when classifying ’fake’ news articles, the model

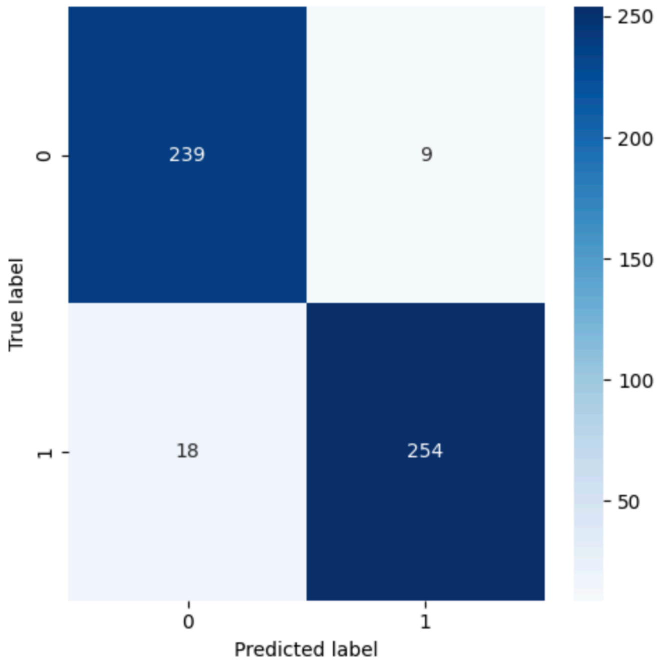

Figure 5.13: Confusion Matrix of SVM (TF-IDF) BanFakeNews Dataset

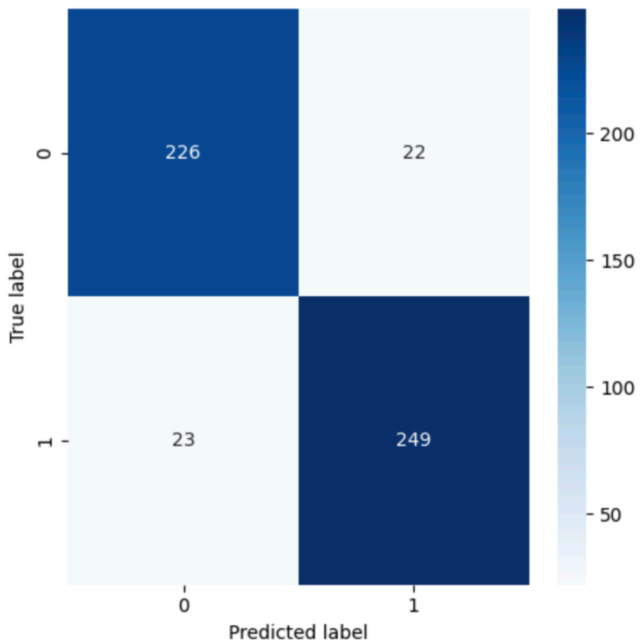

Figure 5.14: Confusion Matrix of MNB (TF-IDF) BanFakeNews Dataset

achieved a precision of 0.94, indicating that 94% of the articles predicted as fake were actually fake. The recall of 0.96 highlights the model's ability to identify 96% of the actual fake articles. The F1-score, a balance between precision and recall, stood at 0.95. This means the model effectively balances both precision and recall in identifying fake news. With 248 instances supporting this classification, these results provide confidence in the model's capability. Similarly, Classification Report

	precision	recall	f1-score	support
0	0.94	0.96	0.95	248
1	0.96	0.94	0.95	272
accuracy			0.95	520
macro avg	0.95	0.95	0.95	520
weighted avg	0.95	0.95	0.95	520

Figure 5.15: Classification report of LR (TF-IDF) BanFakeNews Dataset

for ’real’ news articles, the model demonstrated a precision, recall, and F1-score of 0.96, 0.94, and 0.95, respectively, with 272 instances. This suggests the model’s consistency in accurately classifying both types of articles within the BanFakeNews Dataset, bolstering its reliability in distinguishing between fake and real news.

Classification Report

	precision	recall	f1-score	support
0	0.93	0.96	0.95	248
1	0.97	0.93	0.95	272
accuracy			0.95	520
macro avg	0.95	0.95	0.95	520
weighted avg	0.95	0.95	0.95	520

Figure 5.16: Classification report of SVM (TF-IDF) BanFakeNews Dataset

Bert Model

Table 5.3 shows Bangla-bert-base and mBert model Performance with Accuracy and F1 score on BanFakeNews Dataset, among them Bangla-Bert-Base perform better than mBert. So That we just represent loss and Accuracy curve Figure 5.17, Confusion Matrix Figure 5.18, classification report Figure 5.19 of Bangla-Bert-Base below.

Table 5.3: Model Performance of Bert Model on BanFakeNews Dataset

Model	Accuracy%	F1%
Bangla-bert-base	**94.04%**	**93.98%**
mBert	91.92%	92.00%

The performance of the Bert model, a state-of-the-art transformer-based approach, presents competitive results in fake news detection. Bangla-bert-base achieves an accuracy of 94.04% and an F1 score of 93.98%, comparable to Logistic Regression with BoW feature extraction. Additionally, mBert exhibits slightly lower performance but remains competitive with other traditional models, with an accuracy of 91.92% and an F1 score of 92.00%. These findings suggest that transformer models offer promising avenues for text classification tasks, showcasing competitive performance alongside traditional machine learning algorithms when applied to the BanFakeNews dataset.

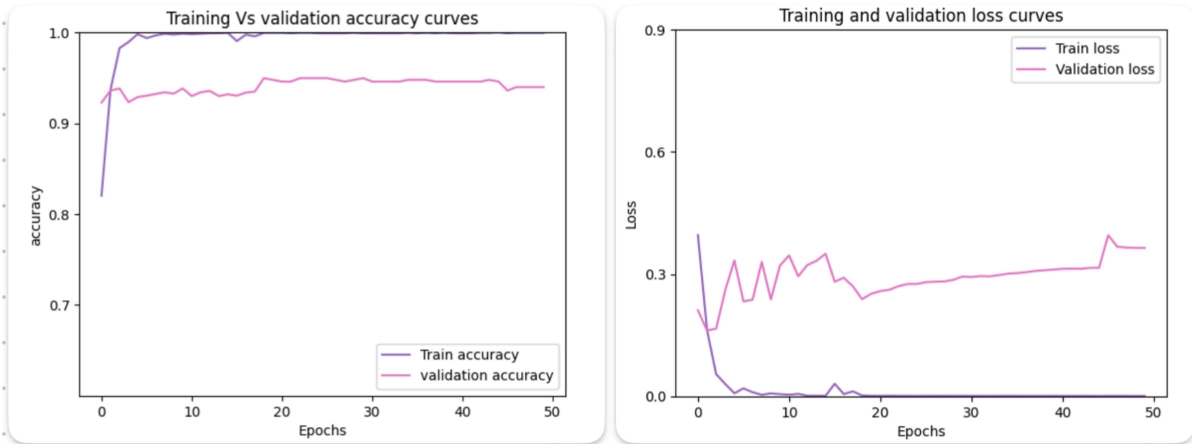

Figure 5.17: Accuracy and Loss curve of Bangla-bert-base on BanFakeNews Dataset

The model was trained using hyper-parameters: 256 sequence length, 50 epochs, 16 batch size, 2e-5 learning rate, and AdamW as an optimizer. We used CrossEntropyLoss as our loss function. Upon examining the Figure 5.17, we can see that the accuracy fluctuates throughout a few epochs. The model has reached the limit of its learning when the accuracy curve flattens out with time. We reach a maximum validation accuracy of 0.9420.

Figure 5.18 shows the confusion matrix of the Bangla-bert-base model for the BanFakeNews dataset reveals its performance. It correctly identified 247 fake news and 242 true news. However, it misclassified 15 real news as fake and 16 fake news as true. These insights underscore Bangla-bert-base effectiveness and areas for improvement in distinguishing between fake and authentic news articles.

Figure 5.19 shows the classification report for the Bangla-bert-base model on the BanFakeNews dataset reveals its strong performance. It achieved a precision, recall, and F1-score of 0.94 for both the 'fake' and 'real' classes, with supports of 262 and 258 instances, respectively. These results indicate the model's reliability in accurately classifying fake and real news articles.

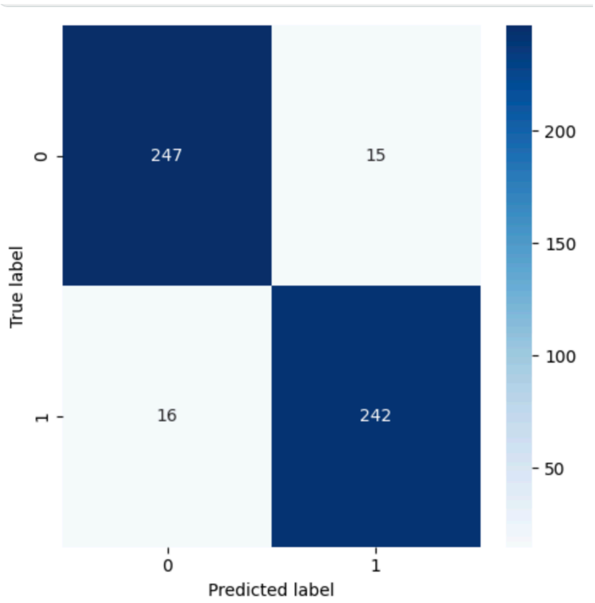

Figure 5.18: Confusion Matrix of Bangla-bert-base on BanFakeNews Dataset

Classification Report

	precision	recall	f1-score	support
0	0.94	0.94	0.94	262
1	0.94	0.94	0.94	258
accuracy			0.94	520
macro avg	0.94	0.94	0.94	520
weighted avg	0.94	0.94	0.94	520

Figure 5.19: Classification report of Bangla-bert-base on BanFakeNews Dataset

5.3.2 Scraped Dataset

Model Performance for BoW Feature Extraction Method with accuracy

Table 5.5 shows Model Performance with Accuracy and F1 score on Scraped Dataset

The model performance table 5.5 reveals insights into the effectiveness of different machine learning algorithms utilizing Bag-of-Words (BoW) feature extraction on a custom scraped dataset. Notably, Multinomial Naive Bayes (MNB) emerges as the top performer, achieving an accuracy of 82.74% and an F1 score of 83.02%, surpassing other models in classification accuracy. MNB's superior performance can be attributed to its inherent strength in handling count-based features like BoW, allowing it to effectively capture the distribution of words within documents and discern between fake and real news articles. Logistic Regression (LR) follows closely behind with an accuracy of 80.31% and an F1 score of 80.26%. However, Support Vector Machine (SVM) and Random Forest (RF) exhibit comparatively lower accuracy rates, suggesting that

Table 5.4: Best parameters of machine learning models using RandomizedSearch on Scraped Dataset

Model	Best-Parameter(RandomizedSearch)
LR (Bow)	$C = 0.5666566321361542$,, Solver = liblinear,penalty= l2, max_iter= 400
MNB (BoW)	alpha=0.6986584841970366
SVM (BoW)	$C = 1.6599452033620266$,, gamma=scale, kernel=rbf, max_iter = 600
RF(BoW)	min_samples_leaf = 3, min_samples_split=2,n_estimators = 215
GB (BoW)	subsample=0.9, n_estimators=200, min_samples_split=20, max_depth=6, learning_rate= 0.2

Table 5.5: Model Performance

Model	Feature extractor	Accuracy%	F1%
LR	BoW	80.31%	80.26%
MNB	BoW	**82.74%**	**83.02%**
SVM	BoW	76.85%	77.72%
RF	BoW	76.21%	73.88%
GB	BoW	73.91%	72.94

the choice of algorithm significantly influences performance when utilizing BoW feature extraction. The relatively poorer performance of SVM and RF may be due to their limitations in capturing the nuanced relationships between words present in the dataset, leading to less accurate classifications.

Figure 5.21 shows the confusion matrix of the Logistic Regression (LR) model trained on the

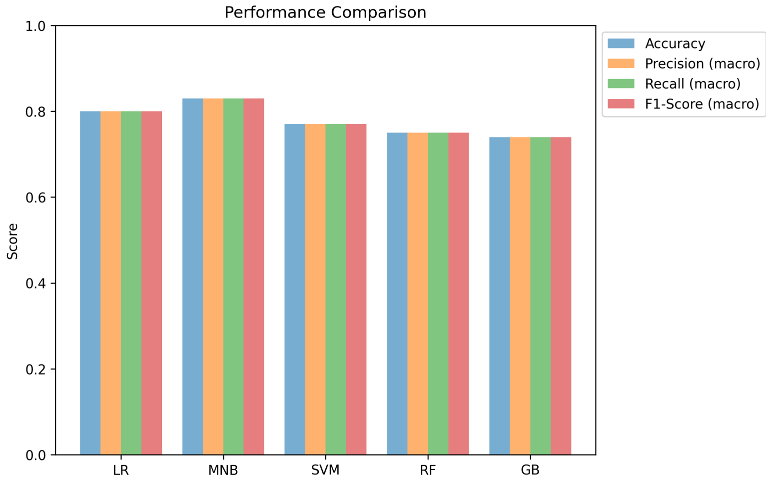

Figure 5.20: Comparison of Accuracy, Precision, Recall, F1-Score obtained applying Five classifiers LR, MNB, SVM, RF, GB for BoW feature Extraction approach on Scraped Dataset

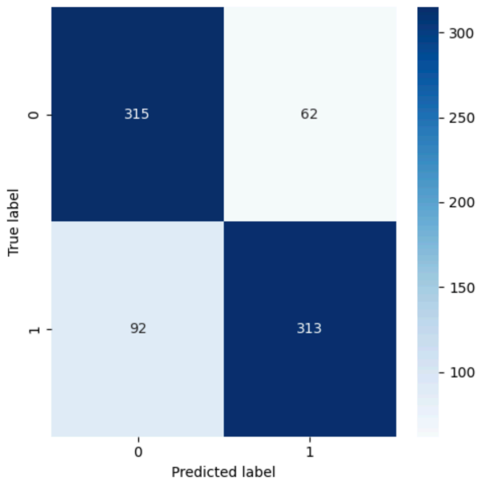

Figure 5.21: Confusion Matrix of LR (BoW) on Scraped Dataset

BoW feature extraction for the Scraped dataset and reveals its performance. It correctly identifies 315 fake news and 313 true news. However, it misclassifies 92 real news as fake and 62 fake news as true. These insights highlight LR's strengths and areas for improvement in distinguishing between fake and real news. Figure 5.22 shows the confusion matrix of the MNB model, trained on BoW features for the Scraped dataset, revealing insights into its performance. It accurately identifies 317 fake news and 330 true news. However, it misclassifies 75 real news as fake and 60 fake news as true. Figure 5.23 shows the confusion matrix of the Support Vector Machine (SVM) model trained on BoW features for the Scraped dataset and reveals its performance. Out of the total instances, 287 fake news articles were correctly identified, along with 314 authentic news articles. However, 91 real news articles were misclassified as fake, and 90

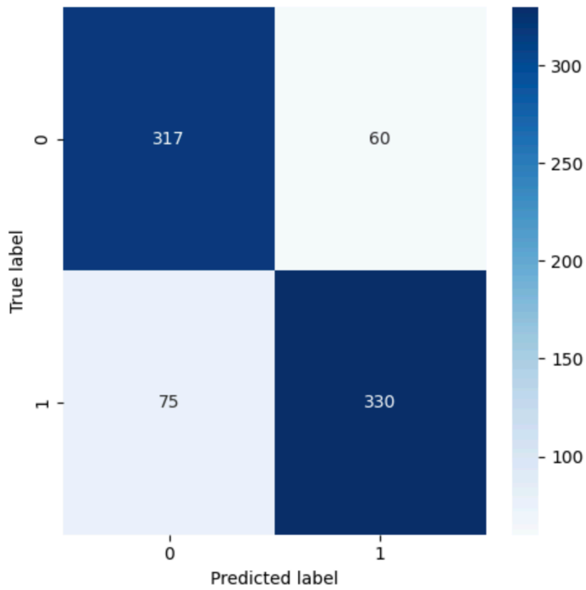

Figure 5.22: Confusion Matrix of MNB (BoW) on Scraped Dataset

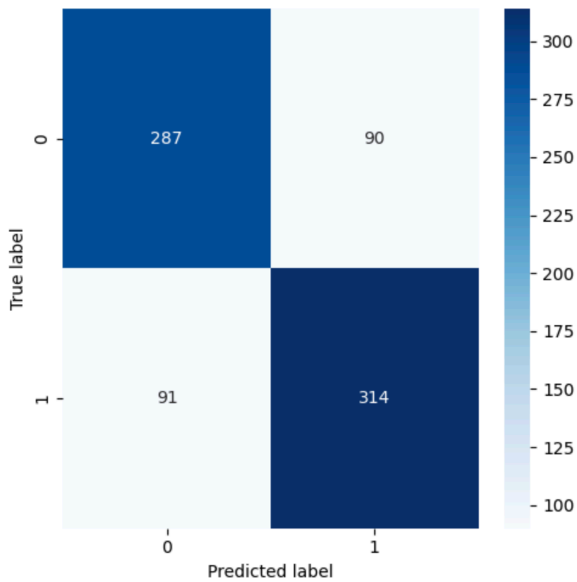

Figure 5.23: Confusion Matrix of SVM (BoW) on Scraped Dataset

fake news articles were misclassified as true. These results highlight both the SVM model's strengths and areas for improvement in accurately classifying news articles.

The Figure 5.24 summarizes the MNB model performance. It achieved a precision, recall, and F1-score of 0.81, 0.84, and 0.82, respectively, for the 'fake' class, with 377 instances supported. For the 'real' class, the model demonstrated precision, recall, and F1-score values of 0.85, 0.81, and 0.83, respectively, with 405 instances supported.

```
Classification Report

              precision    recall  f1-score   support

           0       0.81      0.84      0.82       377
           1       0.85      0.81      0.83       405

    accuracy                           0.83       782
   macro avg       0.83      0.83      0.83       782
weighted avg       0.83      0.83      0.83       782
```

Figure 5.24: Classififcation report of MNB (BoW) on Scraped Dataset

```
Classification Report

              precision    recall  f1-score   support

           0       0.77      0.84      0.80       377
           1       0.83      0.77      0.80       405

    accuracy                           0.80       782
   macro avg       0.80      0.80      0.80       782
weighted avg       0.81      0.80      0.80       782
```

Figure 5.25: Classififcation report of LR (BoW) on Scraped Dataset

Model Performance with TF-IDF Feature Extraction Method on Scraped Dataset

Table 5.6: Best parameters of machine learning models using RandomizedSearch on Scraped Dataset

Model	Best-Parameter(RandomizedSearch)
LR (TF-IDF)	$C = 0.0.5666566321361542$,, Solver = liblinear,penalty= l2, max_iter= 400
MNB (TF-IDF)	alpha= 0.7011150117432088
SVM (TF-IDF)	$C = 1.6599452033620266$, ,gamma= scale, kernel= rbf, max_iter= 600
RF (TF-IDF)	min_samples_leaf=8, min_samples_split=16, n_estimators= 134
GB (TF-IDF)	subsample = 1.0, n_estimators= 200,min_samples_split=15, max_depth=7,learning_rate= 0.2

Table 5.7 shows Model Performance with Accuracy and F1 score on Scraped Dataset

Table 5.7: Model Performance

Model	Feature extractor	Accuracy%	F1%
LR	TF-IDF	82.23%	82.52%
MNB	TF-IDF	**83.50%**	**83.98%**
SVM	TF-IDF	76.85%	77.06%
RF	TF-IDF	71.48%	68.55%
GB	TF-IDF	74.81%	73.84

Making the switch to the TF-IDF feature extraction approach showcases notable improvements across machine learning models on the custom scraped dataset. Multinomial Naive Bayes (MNB) continues to demonstrate superior performance, achieving an impressive accuracy of 83.50% and an F1 score of 83.98%, highlighting the effectiveness of TF-IDF in capturing meaningful textual features. MNB's strong performance can be attributed to its ability to leverage TF-IDF weights to prioritize important words while down-weighting common ones, thus enhancing its discriminatory power. Logistic Regression (LR) also exhibits enhanced performance, with an accuracy of 82.23% and an F1 score of 82.52%. However, some models like Random Forest (RF) show decreased accuracy rates, indicating that TF-IDF may not always yield improved performance across all algorithms. The lower accuracy of RF in the TF-IDF setting may be due to its inability to fully leverage the informative nature of TF-IDF weights, leading to suboptimal feature representation and, consequently, reduced classification accuracy.

Figure 5.27 shows the confusion matrix of the Logistic Regression (LR) model trained on the TF-IDF feature extraction for the Scraped dataset and reveals its performance. It correctly identifies 315 fake news and 328 true news. However, it misclassifies 77 real news as fake and 62 fake news as true. These insights highlight LR's strengths and areas for improvement in distinguishing between fake and real news. Figure 5.28 shows the confusion matrix of the MNB model, trained on TF-IDF features for the Scraped dataset, revealing insights into its performance. It accurately identifies 315 fake news and 338 true news. However, it misclassifies 67 real news as fake and 62 fake news as true. These results highlight both the MNB model's strengths and areas for improvement in accurately classifying news articles. Figure **??** shows

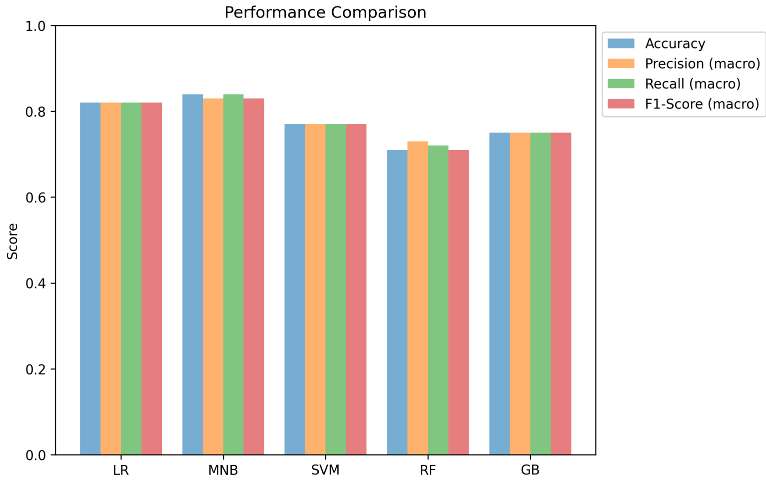

Figure 5.26: Comparison of Accuracy, Precision, Recall, F1-Score obtained applying Five classifiers LR, MNB, SVM, RF, GB for TF-IDF feature Extraction approach on Scraped Dataset

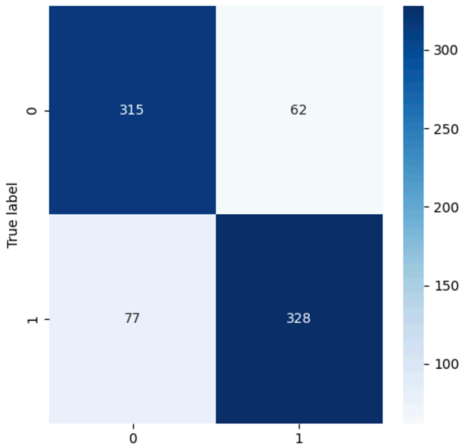

Figure 5.27: Confusion Matrix of LR (TF-IDF) on Scraped Dataset

the confusion matrix of the Support Vector Machine (SVM) model trained on TF-IDF features for the Scraped dataset and reveals its performance. Out of the total instances, 297 fake news articles were correctly identified, along with 304 authentic news articles. However, 101 real news articles were misclassified as fake, and 80 fake news articles were misclassified as true. The Figure 5.30 titled ”Classification report of MNB (TF-IDF) on Scraped Dataset” summarizes the model’s performance. It achieved a precision, recall, and F1-score of 0.82, 0.84, and 0.83, respectively, for the ’fake’ class, with 377 instances supported. For the ’real’ class, the model demonstrated precision, recall, and F1-score values of 0.84, 0.83, and 0.84, respectively, with 405 instances supported. These results highlight the model’s effectiveness in accurately classifying fake and real news articles within the Scraped Dataset.

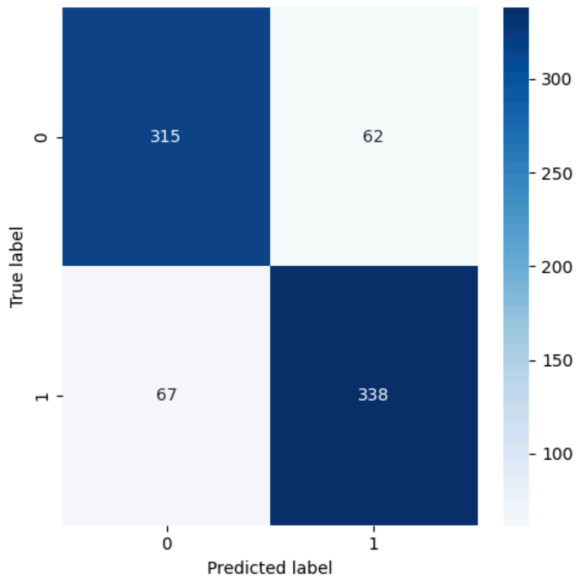

Figure 5.28: Confusion Matrix of MNB (TF-IDF) on Scraped Dataset

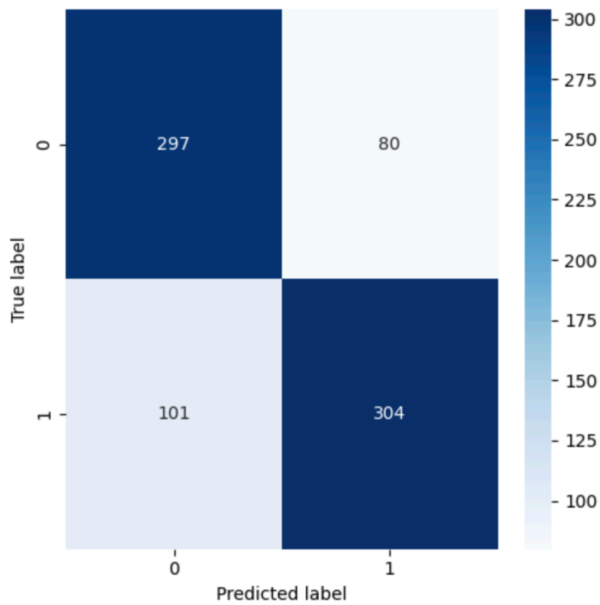

Figure 5.29: Confusion Matrix of SVM (TF-IDF) on Scraped Dataset

Classification Report

	precision	recall	f1-score	support
0	0.82	0.84	0.83	377
1	0.84	0.83	0.84	405
accuracy			0.84	782
macro avg	0.83	0.84	0.83	782
weighted avg	0.84	0.84	0.84	782

Figure 5.30: Classififcation report of MNB (TF-IDF) on Scraped Dataset

```
Classification Report
              precision    recall  f1-score   support

           0       0.80      0.84      0.82       377
           1       0.84      0.81      0.83       405

    accuracy                           0.82       782
   macro avg       0.82      0.82      0.82       782
weighted avg       0.82      0.82      0.82       782
```

Figure 5.31: Classififcation report of LR (TF-IDF) on Scraped Dataset

Bert Model

Table 5.8 shows Bangla-bert-base and mBert model Performance with Accuracy and F1 score on Scraped Dataset, among them Bangla-Bert-Base perform better than mBert. So That we just represent loss and Accuracy curve Figure 5.32, Confusion Matrix Figure 5.33, classification report Figure 5.34 of Bangla-Bert-Base below.

Table 5.8: Bert Models Performance on Scraped Dataset

Model	Accuracy%	F1%
Bangla-Bert-Base	**84.91%**	**85.03%**
mBert	77.88%	77.50%

The Bert models, known for their transformer-based architecture, demonstrate superior performance on the scraped dataset. Bangla-Bert-Base leads with an accuracy of 84.91% and an F1 score of 85.03%, surpassing traditional models. The effectiveness of Bert models lies in their ability to capture intricate relationships and semantic meanings within the text, enabling them to identify subtle patterns that may evade traditional machine learning algorithms. Despite slightly lower accuracy rates, mBert remains competitive. In contrast, traditional models like Logistic Regression (LR) and Random Forest (RF) may struggle due to their reliance on simpler feature representations, which may not capture the nuanced linguistic patterns present in the dataset as effectively.

The model was trained using hyper-parameters: 50 sequence length, 50 epochs, 32 batch size, 2e-5 learning rate, and AdamW as an optimizer. We used CrossEntropyLoss as our loss func-

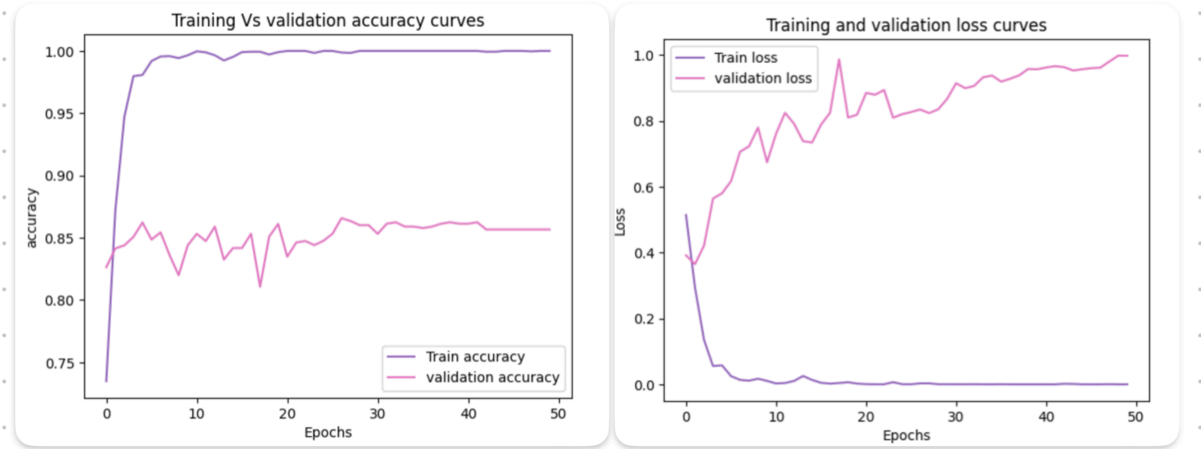

Figure 5.32: Accuracy and Loss Curve of Bangla-bert-base on Scraped Dataset

tion. Observing the Figure5.32, we find that the precision varies across a few epochs. When the accuracy curve flattens out over time, the model has reached the limit of its learning capacity. We reach an upper limit of 0.8657 for validation accuracy. We observed a concerning trend: despite training the Bangla-bert-base model for 50 rounds, the validation loss continued to increase rather than decrease. This implies that the model may be becoming overly dependent on the training set and struggling to learn new information. We propose a few straightforward fixes for this. Initially, dropout or weight decay might be used during training to prevent the model from becoming overly complex. Other options include simplifying the model or terminating training early. Adding more instances for the model to learn from or slightly altering the ones we currently have is another suggestion. Figure 5.33 shows the confusion matrix of the Bangla-bert-base model for the Scraped dataset reveals its performance. It correctly identified 329 fake news and 335 true news. However, it misclassified 70 real news as fake and 48 fake news as true. These insights underscore Bangla-bert-base effectiveness and areas for improvement in distinguishing between fake and authentic news articles.

Figure 5.34 Presents performance metrics for the model. For the 'fake' class, the model achieved a precision, recall, and F1-score of 0.82, 0.87, and 0.85, respectively, with a support of 377 instances. Similarly, for the 'real' class, the model exhibited precision, recall, and F1-score metrics of 0.87, 0.83, and 0.85, respectively, with a support of 405 instances. These results demonstrate the model's capability to accurately classify both fake and real news articles within the Scraped Dataset.

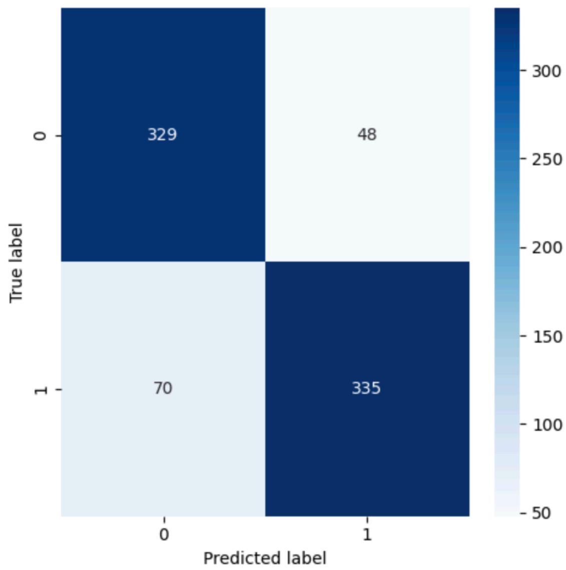

Figure 5.33: Confusion Matrix of Bangla-bert-base on Scraped Dataset

Classification Report

	precision	recall	f1-score	support
0	0.82	0.87	0.85	377
1	0.87	0.83	0.85	405
accuracy			0.85	782
macro avg	0.85	0.85	0.85	782
weighted avg	0.85	0.85	0.85	782

Figure 5.34: Classification report of Bangla-bert-base on Scraped Dataset

5.3.3 Comparison of the Proposed Model

The proposed model's performance is compared with other models, either similar models with the same dataset or different models with similar and different datasets. The goal is to identify the model that achieves the highest accuracy, precision, recall, f1-score, or any other relevant performance metric. Table 5.9 presents a comparison. We performed comparisons between models reported in studies that also used the BanFakenews dataset and those that were trained on it. In addition, we assessed models trained on our created dataset and compared them with research that used different classifiers and created their own dataset Table 5.10. We were able to determine the effectiveness of multiple models across different datasets and classifier selections because of this thorough comparative research, which provided insightful information about the models' performance and applicability.

Table 5.9: A Comparative analysis of different works with the proposed model. Acc: ACCURACY, F1:F1-score on BanFakeNews

Authors	Datasets	Models used	Major Findings
Benazir et al. [34]	Scraped about 2000 Health Tweets in Bangla, (Fake, Real, Ad, Info, Irrelevant, Query, Satire, Unsure Tweets)	CNN with Fasttext Embeddings	Acc: 91.00%
Imran et al.[35]	Scraped Bangla News, (Satire and Fake News)	DNN	Acc: 90.00%
MM Hossain et al. [36]	BanFakeNews Dataset	(SMOTE) LR (Model stacking) Random forest	F1: 93.10% F1: 79.10%
Keya et al.[27]	From BanFakeNews create Augmented Dataset,(Real:4000, Fake:4000)	AugFake-BERT(Bert)	Acc: 92.45% , F1: 91.85%
Hossain et al.[25]	BanFakeNews	Bert	For Fake class F1: 68.00%
proposed model machine learning model	BanFakeNews used(fake:1299, Real:1299) undersampling	feature Extraction:TF-IDF, LR	Acc: 95.19%, F1: 95.36%
proposed Bert model	BanFakeNews used (fake:1299, Real:1299) undersampling	Bangla-Bert-Base	Acc: 94.04%, F1: 93.98%

Table 5.10: A Comparative analysis of different works with the proposed model. Acc: ACCURACY, F1 :F1-score , on Scraped Dataset

Authors	Datasets	Models used	Major Findings
Hossain et al.[25]	BanFakeNews	Model:Bert	Fake class F1: 68.00%
Mugdha et al.[22]	private dataset(Real:296, Fake:296)	Feature Extraction:TF-IDF,Model:GNB	Acc: 87.42%,F1: 82.10%
Islam et al.[37]	private dataset(Ham:1319,spam:646)	Model:MNB	Acc: 82.44% F1: 80.80%
Proposed Model Bangla-Bert-Base	scraped dataset(Real News:1995, Fake News:1913)	Model:Bangla-Bert-base	Acc: 84.91% , F1: 85.03%

5.4 Conclusion

This chapter incorporates the utilization of the confusion matrix and various evaluation metrics to measure the model performance for all the models.

Chapter 6

Conclusion & Future Works

6.1 Introduction

The previous chapter included a discussion of experimental analysis. An overview of our research is given at the start of this chapter. The thesis's shortcomings are then immediately discussed. The next part provides an outline of the future work direction. The chapter concludes with a last thought.

6.2 Summary of Research Work

Two different datasets were used in our investigation. The first dataset is called the BanFakeNews dataset and it comes from Kaggle. Several YouTube channels and fact-checking websites were used to generate the second dataset. The Bert Model was then used in conjunction with a variety of machine learning classifiers to determine how effective they were. We also used two feature extraction techniques, Bag of Words (BoW) and TF-IDF. Among the techniques tested, Logistic Regression (LR) was the most successful, obtaining an outstanding accuracy of 95.19% with TF-IDF feature extraction on the BanFakeNews dataset. The Bangla-Bert-Base model performed better than other models on our Scraped Dataset, with an accuracy of 84.91%.

6.3 Research Contributions

The main contributions of the research are as follows:

- Explored the performance of various classifier models after undersampling the BanFakeNews dataset.
- constructed a dataset and a model utilized this dataset that, can distinguish between real and fraudulent news stories, using a small amount of textual input.
- Gathered labeled fake posts and news titles from fact-checking sites
- Collected legitimate news titles from Bangla news networks on YouTube.
- Tested various classifiers for Bengali false news classification using datasets acquired from various sources.

6.4 Limitations

Some limitations of our research are-

- Other feature extraction approaches such as Word2Vec, and Fasttext were not applied to the datasets.
- Hybrid designs like CNN-LSTM, CNN-BiLSTM, and solo CNN models were not included in the investigation.
- Undersampling was required due to the significant class imbalance present in the BanFakeNews dataset.

6.5 Future Works

There are many opportunities to broaden the area of inquiry of this research in the future. Here are some possible task examples:

- The primary goal will be to increase the accuracy of the model by thoroughly investigating different hyperparameter setups with GridSearch.
- we will make an effort to improve the generated dataset's quality and its amount to boost performance.

6.6 Conclusion

In conclusion, this research has explored machine learning classifiers and Transformer models for Bangla fake news detection. Logistic regression achieved remarkable accuracy on the BanFakeNews dataset, while the Bangla-Bert-Base model showed promise on the Scraped Dataset. Contributions include dataset construction and model development. Limitations include the omission of certain feature extraction approaches and the need for undersampling. Future work involves optimizing hyperparameters and enhancing dataset quality. Overall, this research lays a foundation for further advancements in Bangla fake news detection.

REFERENCES

[1] Wikipedia contributors, “Bengali language — Wikipedia, the free encyclopedia,” 2024. [Online; accessed 22-March-2024].

[2] Wikipedia contributors, “Supervised learning — Wikipedia, the free encyclopedia,” 2023. [Online; accessed 22-March-2024].

[3] Wikipedia contributors, “Transfer learning — Wikipedia, the free encyclopedia,” 2024. [Online; accessed 22-March-2024].

[4] Wikipedia contributors, “Natural language processing — Wikipedia, the free encyclopedia,” 2024. [Online; accessed 22-March-2024].

[5] [Online; Accessed 20-April-2024].

[6] W. Commons, “File:svm margin.png — wikimedia commons, the free media repository,” 2022. [Online; accessed 18-April-2024].

[7] Wikipedia contributors, “Rina dechter — Wikipedia, the free encyclopedia,” 2024. [Online; accessed 26-March-2024].

[8] Wikipedia contributors, “Deep learning — Wikipedia, the free encyclopedia,” 2024. [Online; accessed 26-March-2024].

[9] Wikipedia contributors, “Convolutional neural network — Wikipedia, the free encyclopedia,” 2024. [Online; accessed 26-March-2024].

[10] Wikipedia contributors, “Recurrent neural network — Wikipedia, the free encyclopedia,” 2024. [Online; accessed 26-March-2024].

[11] Wikipedia contributors, “Transformer (deep learning architecture) — Wikipedia, the free encyclopedia,” 2024. [Online; accessed 26-March-2024].

[12] Wikipedia contributors, "Neuron — Wikipedia, the free encyclopedia," 2024. [Online; accessed 26-March-2024].

[13] Wikipedia contributors, "Neural network (biology) — Wikipedia, the free encyclopedia," 2024. [Online; accessed 26-March-2024].

[14] Wikipedia contributors, "Frank rosenblatt — Wikipedia, the free encyclopedia," 2024. [Online; accessed 26-March-2024].

[15] javatpoint, "The perceptron." `https://www.javatpoint.com/single-layer-perceptron-in-tensorflow`.

[16] Wikipedia contributors, "Neural network (machine learning) — Wikipedia, the free encyclopedia," 2024. [Online; accessed 26-March-2024].

[17] W. Commons, "File:recurrent neural network unfold.svg — wikimedia commons, the free media repository," 2024. [Online; accessed 20-April-2024].

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] "Natural language processing with transformers: Building language applications with hugging face."

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

[21] Wikipedia contributors, "Natural language processing — Wikipedia, the free encyclopedia," 2024. [Online; accessed 26-March-2024].

[22] S. B. S. Mugdha, S. M. Ferdous, and A. Fahmin, "Evaluating machine learning algorithms for bengali fake news detection," in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, IEEE, 2020.

[23] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "Fndnet–a deep convolutional neural network for fake news detection," *Cognitive Systems Research*, vol. 61, pp. 32–44, 2020.

[24] E. Hossain, M. Nadim Kaysar, A. Z. M. Jalal Uddin Joy, M. Mizanur Rahman, and W. Rahman, "A study towards bangla fake news detection using machine learning and deep learning," in *Sentimental Analysis and Deep Learning: Proceedings of ICSADL 2021*, pp. 79–95, Springer, 2022.

[25] M. Z. Hossain, M. A. Rahman, M. S. Islam, and S. Kar, "Banfakenews: A dataset for detecting fake news in bangla," *arXiv preprint arXiv:2004.08789*, 2020.

[26] R. I. Rasel, A. H. Zihad, N. Sultana, and M. M. Hoque, "Bangla fake news detection using machine learning, deep learning and transformer models," in *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pp. 959–964, IEEE, 2022.

[27] A. J. Keya, M. A. H. Wadud, M. Mridha, M. Alatiyyah, and M. A. Hamid, "Augfake-bert: handling imbalance through augmentation of fake news using bert to enhance the performance of fake news classification," *Applied Sciences*, vol. 12, no. 17, p. 8398, 2022.

[28] M. A. T. Akbar, "fact-checker website." `https://www.boombd.com/`.

[29] jachai, "fact-checker website." `https://www.jachai.org/`.

[30] [Online; Accessed 20-April-2024].

[31] S. Sarker, "Banglabert: Bengali mask language model for bengali language understanding," 2020.

[32] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[33] A. Suresh, "What is a confusion matrix?." `https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5`.

[34] A. Benazir and S. Sharmin, "Credibility assessment of user generated health information of the bengali language in microblogging sites employing nlp techniques," in *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pp. 837–844, IEEE, 2020.

[35] A. Al Imran, Z. Wahid, and T. Ahmed, "Bnnet: A deep neural network for the identification of satire and fake bangla news," in *Computational Data and Social Networks:*

9th International Conference, CSoNet 2020, Dallas, TX, USA, December 11–13, 2020, Proceedings 9, pp. 464–475, Springer, 2020.

[36] M. M. Hossain, Z. Awosaf, M. S. H. Prottoy, A. S. M. Alvy, and M. K. Morol, "Approaches for improving the performance of fake news detection in bangla: Imbalance handling and model stacking," in *Proceedings of International Conference on Fourth Industrial Revolution and Beyond 2021*, pp. 723–734, Springer, 2022.

[37] T. Islam, S. Latif, and N. Ahmed, "Using social networks to detect malicious bangla text content," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1–4, IEEE, 2019.